SIAM J. IMAGING SCIENCES
Vol. 15, No. 3, pp. 1539–1575

WARPd: A Linearly Convergent First-Order Primal-Dual Algorithm for Inverse Problems with Approximate Sharpness Conditions*

Matthew J. Colbrook†

Abstract. Sharpness conditions directly control the recovery performance of restart schemes for first-order optimization methods without the need for restrictive assumptions such as strong convexity. However, they are challenging to apply in the presence of noise or approximate model classes (e.g., approximate sparsity). We provide a first-order method: weighted, accelerated, and restarted primal-dual (WARPd), based on primal-dual iterations and a novel restart-reweight scheme. Under a generic approximate sharpness condition, WARPd achieves stable linear convergence to the desired vector. Many problems of interest fit into this framework. For example, we analyze sparse recovery in compressed sensing, low-rank matrix recovery, matrix completion, TV regularization, minimization of $\|Bx\|_{l^1}$ under constraints (l^1-analysis problems for general B), and mixed regularization problems. We show how several quantities controlling recovery performance also provide explicit approximate sharpness constants. Numerical experiments show that WARPd compares favorably with specialized state-of-the-art methods and is ideally suited for solving large-scale problems. We also present a noise-blind variant based on a square-root LASSO decoder. Finally, we show how to unroll WARPd as neural networks. This approximation theory result provides lower bounds for stable and accurate neural networks for inverse problems and sheds light on architecture choices. Code and a gallery of examples are available online as a MATLAB package.

Key words. approximate sharpness, error bounds, accelerated methods, primal-dual algorithms, restart, compressed sensing, matrix completion, total variation minimization, image reconstruction, neural networks

MSC codes. 65K10, 68U10, 65Y20, 68Q25, 90C25, 94A08, 15A83

DOI. 10.1137/21M1455000

1. Introduction.

Reconstruction from sampled measurements is a key problem in signal and image processing, machine learning, statistics, computer vision, and a variety of other fields. In this paper, we consider the following canonical linear inverse problem:[1]

$$\text{Given measurements } b = A\mathbb{x} + e \in \mathcal{Y}_1^*, \text{ recover } \mathbb{x} \in \mathcal{X}. \tag{1.1}$$

Here $\mathcal{X}$ and $\mathcal{Y}_1$ are (real or complex) Banach spaces, $A : \mathcal{X} \to \mathcal{Y}_1^*$ is a bounded linear operator that represents a sampling model, and $e \in \mathcal{Y}_1^*$ models noise or perturbations. The dual space (the space of all continuous linear functionals) of a Banach space $\mathcal{Z}$ is denoted by $\mathcal{Z}^*$.

Over the last few decades there has been an explosion in nonlinear reconstruction techniques for (1.1) (see [6, 12, 16, 21, 32, 40, 43, 47, 58, 67, 71, 91, 105, 111, 116] for a very small

*Received by the editors October 25, 2021; accepted for publication (in revised form) June 10, 2022; published electronically September 15, 2022.
https://doi.org/10.1137/21M1455000
Funding: The work of the author was supported by a Research Fellowship at Trinity College, University of Cambridge, and a Fondation Sciences Mathematiques de Paris postdoctoral fellowship at École Normale Supérieure.

†Centre Sciences des Données, École Normale Supérieure, Paris, France (m.colbrook@damtp.cam.ac.uk).

[1]We have used the notation $\mathbb{x}$ to avoid confusion with x used to denote a dummy variable.

sample). For example, the field of compressed sensing shows that, under certain conditions, accurate reconstruction is possible if $\mathbb{x}$ is (approximately) sparse [33, 34, 50]. More generally, a popular approach for recovering $\mathbb{x}$ is to solve an optimization problem of the form[2]

$$\min_{x\in\mathcal{X}} \mathcal{J}(x) + \mathcal{F}(Bx) \quad \text{such that} \quad \|Ax-b\|_{\mathcal{Y}_1^*} \le \epsilon. \tag{1.2}$$

Here, $\mathcal{J} : \mathcal{X} \to \mathbb{R}\cup\{+\infty\}$ and $\mathcal{F} : \mathcal{Y}_2^* \to \mathbb{R}\cup\{+\infty\}$ are proper, lower semicontinuous, convex functions, $\mathcal{Y}_2$ is a Banach space, and $B : \mathcal{X} \to \mathcal{Y}_2^*$ is a bounded linear operator. We assume that the sum $\mathcal{J}(x) + \mathcal{F}(Bx)$ is bounded below and, without loss of generality, nonnegative. We use $\|\cdot\|_{\mathcal{Z}}$ to denote the norm of a Banach space $\mathcal{Z}$, $\langle\cdot,\cdot\rangle$ for the bilinear form on $\mathcal{Z}^*\times\mathcal{Z}$, and $\langle\cdot,\cdot\rangle_{\mathbb{R}}$ for the real part of $\langle\cdot,\cdot\rangle$. For a space $\mathcal{Z}$, we use a Bregman distance [54] $D_{\mathcal{Z}}$ associated with a 1-strongly convex (with respect to $\|\cdot\|_{\mathcal{Z}}$) continuously differentiable function. In particular, $D_{\mathcal{Z}}(z,\hat{z}) \ge \frac{1}{2}\|z-\hat{z}\|_{\mathcal{Z}}^2$. If $\mathcal{Z}$ is a Hilbert space, the most common choice is $D_{\mathcal{Z}}(z,\hat{z}) = \frac{1}{2}\|z-\hat{z}\|_{\mathcal{Z}}^2$. We let $g_1, g_2 : \mathbb{R}_{>0} \to \mathbb{R}_{>0}\cup\{\infty\}$ be functions such that for all $\eta > 0$ and all $x \in \mathcal{X}$,

$$\sup_{\{y_1\in\mathcal{Y}_1 : \|y_1\|_{\mathcal{Y}_1}\le\eta\}} D_{\mathcal{Y}_1}(y_1,0) \le g_1(\eta), \tag{1.3}$$

$$\eta\mathcal{F}(Bx) = \sup_{\{y_2\in\mathcal{Y}_2 : D_{\mathcal{Y}_2}(y_2,0)\le g_2(\eta)\}} \langle Bx, y_2\rangle_{\mathbb{R}} - \eta\mathcal{F}^*(y_2/\eta), \tag{1.4}$$

where f^* denotes the convex conjugate of a function f. We make assumptions on the finiteness of g_1 and g_2 for fixed specific inputs. These are explicitly given in (1.6) and (2.20). In particular, we do not assume that the domain of the Bregman function $D_{\mathcal{Z}}$ is all of $\mathcal{Z}$.[3]

Example 1.1 (finite-dimensional Hilbert spaces). A concrete example that the reader should keep in mind is when $\mathcal{X} = \mathbb{C}^N$, $\mathcal{Y}_1 = \mathbb{C}^m$, and $\mathcal{Y}_2 = \mathbb{C}^q$, all with the usual inner product. Here, $\mathbb{x} \in \mathbb{C}^N$ could also correspond to a vectorized pixelated image or matrix. In the undersampled regime, $A \in \mathbb{C}^{m\times N}$ with $m < N$. A common choice of $\mathcal{J}$ is a regularizer that depends on prior assumptions about the signal $\mathbb{x}$, such as $\mathcal{J}(x) = \|x\|_{l^1}$, which promotes sparsity (see section 3), or the nuclear norm of matrices, which promotes matrices of low-rank (see section 4). If $\mathcal{F}(x) = \|x\|_{l^1}$, then $\mathcal{F}(Bx) = \|Bx\|_{l^1}$ could correspond to the popular TV-seminorm $\|x\|_{\mathrm{TV}}$ [38] (see subsection 6.1.2) or a sum $\|Wx\|_{l^1} + \lambda\|x\|_{\mathrm{TV}}$ for general W (see subsection 6.2). Note that if $D_{\mathcal{Z}}(z,\hat{z}) = \frac{1}{2}\|z-\hat{z}\|_{l^2}^2$, then we can take $g_1(\eta) = \eta^2/2$, and if $\mathcal{F}(x) = \|x\|_{l^1}$, then we can take $g_2(\eta) = q\eta^2/2$.

Due to the great interest in solving (1.2) and similar problems, there is a long list of algorithms (see section SM1), with a particular emphasis on first-order methods[4] for large-

[2]The formulation in (1.2) with the constraint $\|Ax-b\|_{\mathcal{Y}_1^*} \le \epsilon$ is often theoretically preferred over other variations (e.g., $\|Ax-b\|_{\mathcal{Y}_1^*}^2$ in the objective function) because a reasonable estimate of ϵ may be known [17]. The case of unknown ϵ and replacing the constraint $\|Ax-b\|_{\mathcal{Y}_1^*} \le \epsilon$ by an additional term $\lambda^{-1}\|Ax-b\|_{\mathcal{Y}_1^*}$ in the objective function, where λ scales *independently* of the noise, is treated in subsection 2.4.

[3]Note that if $\{y_2 \in \mathcal{Y}_2 : D_{\mathcal{Y}_2}(y_2,0) \le g_2(\eta)\}$ is bounded and the supremum is realized, then (1.4) implies at most linear growth of $\mathcal{F}$ on the image of B. It may be possible to get around this assumption by a careful continuation argument of the dual variables in our restart scheme, but we do not pursue this further.

[4]Due to their dimensionality, many large-scale optimization models have rendered second-order methods computationally impractical (typically large systems of linear equations are solved to compute Newton steps). Thus, efficient and accelerated first-order algorithms have become essential for tackling numerous problems.

scale problems. The goal is to design simple schemes (e.g., matrix/vector multiplications in the case of Example 1.1) that produce approximate solutions efficiently. Solving (1.2) is a notoriously difficult challenge, with common issues being nonsmoothness of $\mathcal{J}$, the term $\mathcal{F}(Bx)$ (e.g., the analysis term in Example 1.1), the constraint $\|Ax-b\|_{\mathcal{Y}_1^*} \leq \epsilon$, etc. First-order methods typically need a very large number of iterations when high accuracy is required (see [98] for optimal convergence rates for different classes of objective functions). There are also many works on the limits of recovering $\mathbb{x}$ via solutions of (1.2), and this is intimately linked to numerical performance. It has been observed that recovery problems (1.1) that are easier to solve theoretically (e.g., larger m in Example 1.1) often lead to optimization problems (1.2) that are easier/more efficient to solve numerically [9, 42, 51]. In some cases, this has led to algorithms with accelerated convergence guarantees [109].

This paper provides a general framework for the accelerated (linear) convergence and stable solution of (1.1). Our only assumption is an inequality of the form

$$\sqrt{2D_{\mathcal{X}}(\mathbb{x},\hat{x})} \leq C_1\Big[\underbrace{\mathcal{J}(\hat{x})+\mathcal{F}(B\hat{x})-\mathcal{J}(\mathbb{x})-\mathcal{F}(B\mathbb{x})}_{\text{objective function difference}}+C_2\underbrace{\big(\|A\hat{x}-b\|_{\mathcal{Y}_1^*}-\epsilon\big)}_{\text{feasibility gap}}+\underbrace{c(\mathbb{x},b)}_{\text{approx. term}}\Big] \quad \forall \hat{x}\in\mathcal{X}. \tag{1.5}$$

Though we have written (1.5) in a global form over $\hat{x}$, we only need it to hold for $\hat{x}$ sufficiently close to $\mathbb{x}$ for a suitable initial vector of our iteration scheme. For brevity, we omit the details. Here C_1 and C_2 are constants and $c(\mathbb{x},b)$ should be understood as a small approximation term. For example, for sparse recovery considered in section 3, $c(\mathbb{x},b)$ measures the distance of $\mathbb{x}$ to sparse vectors and contains a term proportional to the noise level ϵ (see Theorem 3.3). We further assume that

$$g_1(C_2) < \infty, \quad g_2(1) < \infty, \tag{1.6}$$

where g_1 and g_2 are the functions in (1.3) and (1.4). The assumption (1.5) is much weaker than typical assumptions for acceleration such as strong convexity and can be considered an *approximate* Łojasiewicz-type inequality or "approximate sharpness" [110]. We discuss its links to other error bounds in section SM1. A key difference between (1.5) and sharpness, and hence also between the restart scheme of this paper and others, is that we only assume *approximate* control of the distance via the objective function difference—this is reflected by the parameter δ in Theorem 1.2 and the term $c(\mathbb{x},b)$ in (1.5). For the type of problems we consider, this gives us greater generality and allows us to tackle the case of *noisy measurements*, as well as prove *robustness* of our results (e.g., when considering sparse recovery, we cover approximately sparse vectors). However, it also means that the vector $\mathbb{x}$ can only be recovered approximately to order δ. Curiously, numerical experiments below demonstrate that we continue to achieve linear convergence in the objective function values. In sections 3 to 6, we provide an analysis that verifies (1.5) for a range of important examples.

Given (1.5), we provide an iterative algorithm, weighted, accelerated, and restarted primal-dual (WARPd), based on primal-dual iterations and a novel restart-reweight scheme. Our main convergence result is summarized in the following theorem.

Theorem 1.2 (stable recovery with linear convergence). *Let L_A and L_B be upper bounds for $\|A\|$ and $\|B\|$, respectively, and $\delta > 0$. Then for any $n \in \mathbb{N}$ and any pair $(\mathbb{x},b) \in \mathcal{X} \times \mathcal{Y}_1^*$ such*

that $\|A\mathbb{x} - b\|_{\mathcal{Y}_1^*} \leq \epsilon$, (1.5) *holds, and* $c(\mathbb{x}, b) \leq \delta$,

$$\|\phi_n(b) - \mathbb{x}\|_{\mathcal{X}} \leq \sqrt{2D_{\mathcal{X}}(\mathbb{x}, \phi_n(b))} \leq C_1 \left(\frac{\delta}{1 - e^{-1}} + e^{-n} \left[\mathcal{J}(0) + \mathcal{F}(0) + C_2 \|b\|_{\mathcal{Y}_1^*} \right] \right), \tag{1.7}$$

where $\phi_n(b)$ *denotes the output of WARPd (Algorithm* 2.3*).*

The total number of inner iterations scales linearly with n. Only

$$\sim C_1 \left(L_A \sqrt{g_1(C_2)} + L_B \sqrt{g_2(1)} \right) \log \left(\left[\mathcal{J}(0) + \mathcal{F}(0) + C_2 \|b\|_{\mathcal{Y}_1^*} \right] / \delta \right) \tag{1.8}$$

total inner iterations are required to balance the two terms on the right-hand side of (1.7). In other words, Theorem 1.2 demonstrates *linear (or exponential) convergence down to the error bound* $\sim C_1\delta$. Note that the barrier $C_1\delta$ between solutions of (1.2) and $\mathbb{x}$ in (1.1) is to be expected from the $c(\mathbb{x}, b)$ term in (1.5) when the objective function difference and feasibility gap vanish. The convergence result is stable with respect to perturbations of $\mathbb{x}$ or b, with stability governed by $c(\mathbb{x}, b) \leq \delta$. Finally, each iteration of WARPd only requires applying appropriate proximal maps. In the case of Example 1.1, we only need the proximal map of $\mathcal{J}$ and a few matrix-vector multiplications. In particular, we do not assume anything on the matrices A and B (e.g., we do not assume that A^*A is an orthogonal projector or that B is diagonal). Together with the acceleration, this makes WARPd very computationally efficient.

Many problems of interest satisfy a version of (1.5) and there is great flexibility in our framework. To be concrete, we explicitly analyze the following examples:

- Section 3: Sparse recovery, using the robust null space property (in levels) to obtain (1.5).
- Section 4: Low-rank matrix recovery, using the Frobenius-robust rank null space property to obtain (1.5).
- Section 5: Matrix completion, using approximate dual certificates to obtain (1.5).
- Section 6: Examples with nontrivial matrix B including l^1-analysis with frames (using a generalization of the restricted isometry property to obtain (1.5)) and total variation (TV) minimization (using the restricted isometry property to obtain (1.5)). Similar results can be proven using the robust null space property for frames.

Comprehensive numerical experiments demonstrate that WARPd compares favorably with state-of-the-art methods. We also consider a variant WARPd-SR in subsection 2.4 that covers the case of unknown ϵ and replaces the constraint $\|Ax - b\|_{\mathcal{Y}_1^*} \leq \epsilon$ by a term $\lambda^{-1}\|Ax - b\|_{\mathcal{Y}_1^*}$ in the objective function, where λ scales *independently* of the noise.

1.1. Accurate and stable neural networks. Given the current interest in deep learning (DL), it is not surprising that numerous DL-based methods are now being proposed for the above and similar problems (see [12, 26, 66, 71, 75, 92, 122] for a small sample). There is ample evidence that DL has the potential to achieve state-of-the-art results in numerous applications. However, a current challenge is that many DL-based methods lack theoretical foundations regarding reconstruction guarantees, convergence rates, stability analysis, and other basic numerical analysis questions. The stability question is particularly alarming, with empirical evidence that current DL techniques can lead to unstable methods for inverse problems (e.g., "adversarial attacks") [11, 59, 69]. For example, this is a problem in real-world clinical practice. Facebook and NYU's 2019 FastMRI challenge reported that networks that performed well in

standard image quality metrics were prone to false negatives, failing to reconstruct small but physically relevant image abnormalities [74]. Subsequently, the 2020 FastMRI challenge [94] focused on pathologies and "AI-generated hallucinations." AI-generated hallucinations pose a serious danger in applications such as medical imaging. The big problem, therefore, is to compute/train neural networks (NNs) that are both accurate and stable [4, 46, 48].

In light of this, we consider unrolling WARPd as an NN. Unrolling iterative algorithms as NNs is an increasingly popular method [68, 92, 93] and is particularly well suited to scenarios where it is difficult to collect large training samples. Without stronger assumptions on the objective function, naive unrolling of first-order iterative methods typically provides slow $\mathcal{O}(\delta + n^{-1})$ convergence guarantees in the number of hidden layers n.[5] Instead, we gain $\mathcal{O}(\delta + e^{-n})$ convergence, providing lower bounds on what is achievable in terms of stability and accuracy of an NN. The following theorem provides the approximation theory result.

Theorem 1.3. *Suppose we are in the setting of Example 1.1 and that* $\mathcal{F}(Bx) = \|Bx\|_{l^1}$. *Let* L_A *and* L_B *be upper bounds for* $\|A\|$ *and* $\|B\|$, *respectively, and* $\delta > 0$. *Suppose that* (1.5) *holds and that the proximal map of* $\mathcal{J}$ *can be approximated to the required accuracy described by* (2.26) *via an NN of width bounded by a constant times* $N+m+q$ *and depth* M. *We provide a NN* ϕ *of width bounded by a constant times* $N + m + q$ *and depth bounded by a constant times* $MC_1(L_AC_2 + L_B\sqrt{q}) \cdot \log\left(\left[\mathcal{J}(0) + C_2\|b\|_{l^2}\right]/\delta\right)$ *such that the following stable recovery guarantee holds. For any pair* $(\mathbb{x}, b) \in \mathbb{C}^N \times \mathbb{C}^m$ *such that* $\|A\mathbb{x} - b\|_{l^2} \leq \epsilon$ *and* $c(\mathbb{x}, b) \leq \delta$,

$$\|\phi(b) - \mathbb{x}\|_{l^2} \lesssim C_1\delta. \tag{1.9}$$

The key points are (a) the total number of parameters and depth of the NN only depend logarithmically on the error tolerance δ (*accuracy* and *efficiency*), and (b) the recovery guarantee is stable in the l^2-norm (in terms of the data b and the bound $c(\mathbb{x}, b) \leq \delta$) for the model class described by $c(\mathbb{x}, b) \leq \delta$ (*stability*). This result provides lower bounds for what is achievable in terms of stable and accurate NNs.

Regarding the approximation of the proximal map of $\mathcal{J}$, for the examples in sections 3 and 6 this can be achieved exactly using a fixed depth ($M = \mathcal{O}(1)$ in Theorem 1.3). For the examples of low-rank matrix recovery and matrix completion in sections 4 and 5, the proximal map is computed via a partial singular value decomposition (SVD). This is typically achieved via iterative methods, which can be unrolled as recurrent NNs. The precise number of iterations needed depends on the matrix and singular values/vectors that are sought. Finally, one can obtain similar results when the matrices A and B are only known approximately, and the layers of the NN are only applied approximately (see subsection 2.5).

1.2. Notation. In addition to the notation introduced in the first subsection of the paper, we use $\|\cdot\|_{l^p}$ to denote the standard l^p-norm of vectors in $\mathbb{C}^n$. Given a bounded linear operator A between Banach spaces, we denote the operator norm of A by $\|A\|$. Given a lower semi-continuous convex function f from a Hilbert space $\mathcal{H}$ to $[-\infty, \infty]$, we use the proximal operator

[5] There are some exceptions. For example, [62] ensures that the iterations are contractive. Another example is [45, 85] for LISTA (a learned version of ISTA) that ensures the existence of NN with linear convergence toward the minimizer. However, neither [45] nor [85] uses the theoretically correct weights, as these can only be computed as solutions of intractably large optimization problems. It is also unclear whether the needed assumptions on the measurement operator A hold in practice.

$\text{prox}_f(v) = \text{argmin}_{x\in\mathcal{H}} f(x) + \frac{1}{2}\|v-x\|_{\mathcal{H}}^2$. Throughout, $a \lesssim b$ will mean there is a constant C (independent of all relevant parameters) such that $a \le Cb$. Given a matrix M with singular values $\sigma_1(M) \ge \sigma_2(M) \ge \cdots \ge \sigma_r(M)$, we denote by $\|M\|_p$ the Schatten p-norm of M, which is the l^p-norm of the sequence of singular values $\{\sigma_j(M)\}$. We let χ_S denote the indicator function of a set S, taking the value 0 on S and $+\infty$ otherwise. Finally, $\mathcal{B}_{\mathcal{Z}}$ denotes the unit ball of a Banach space $\mathcal{Z}$ and $\mathcal{P}_{\mathcal{Z}}$ denotes the standard projection onto $\mathcal{B}_{\mathcal{Z}}$.

1.3. Outline of paper. In section 2, we introduce WARPd, prove its convergence properties (e.g., Theorem 1.2), discuss its computational complexity and practice, provide a variation (WARPd-SR) suitable for noise-blind recovery problems (unknown ϵ), and prove Theorem 1.3. Section 3 analyzes the example of sparse recovery, section 4 analyzes the example of (approximately) low-rank matrix recovery, section 5 analyzes the example of matrix completion, and section 6 analyzes the examples of l^1-analysis and TV minimization. Numerical examples are given throughout the paper, and code is available at https://github.com/MColbrook/WARPd. For brevity, proofs of the theoretical results we derive in sections 3 to 6 as well as of Theorem 2.3 are given in the supplementary materials (M145500_01.pdf [local/web 460KB]), which also contains an expanded section on connections with previous work. We conclude the paper with a discussion, including future work, in section 7.

2. The accelerated algorithm. We begin with the primal-dual iterations in subsection 2.1 (general case) and subsection 2.2 (Hilbert space case), and then describe the restart scheme in subsection 2.3. Theorem 1.2 provides the error bounds for WARPd described in Algorithm 2.3. In subsection 2.4, we provide a variation, WARPd-SR, based on replacing the constraint $\|Ax-b\|_{\mathcal{Y}_1^*} \le \epsilon$ in (1.2) with an additional data fitting term $\|Ax-b\|_{\mathcal{Y}_1^*}$ in the objective function. This is well suited to *noise-blind* recovery problems (unknown ϵ) and provides an elegant means to bound dual variables for warm restarts. Computational considerations are given in subsection 2.5 and we prove Theorem 1.3 in subsection 2.6.

2.1. Primal-dual iterations: The general case. The following provide useful characterizations of the constraint $\|Ax-b\|_{\mathcal{Y}_1^*} \le \epsilon$ and the $\mathcal{F}(Bx)$ term appearing in (1.2):

$$\chi_{\{\hat{x}:\|A\hat{x}-b\|_{\mathcal{Y}_1^*}\le\epsilon\}}(x) = \sup_{y_1\in\mathcal{Y}_1} \langle Ax-b, y_1\rangle_{\mathbb{R}} - \epsilon\|y_1\|_{\mathcal{Y}_1}, \quad \mathcal{F}(Bx) = \sup_{y_2\in\mathcal{Y}_2} \langle Bx, y_2\rangle_{\mathbb{R}} - \mathcal{F}^*(y_2). \tag{2.1}$$

It follows that the problem (1.2) is equivalent to the saddle point problem

$$\inf_{x\in\mathcal{X}} \sup_{y_1\in\mathcal{Y}_1, y_2\in\mathcal{Y}_2} \Big[\mathcal{L}(x,y_1,y_2) := \langle Ax-b, y_1\rangle_{\mathbb{R}} + \langle Bx, y_2\rangle_{\mathbb{R}} + \mathcal{J}(x) - \epsilon\|y_1\|_{\mathcal{Y}_1} - \mathcal{F}^*(y_2)\Big]. \tag{2.2}$$

To solve (2.2), we use a primal-dual algorithm [39, 41], with iterates summarized in Algorithm 2.1. The quantities $\tau_1, \tau_2, \tau_3 > 0$ denote proximal step sizes. Letting $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3)$, we use $\text{PD}_{\boldsymbol{\tau}}$ to denote the updates in (2.5) so that

$$(x^{(j+1)}, y_1^{(j+1)}, y_2^{(j+1)}) = \text{PD}_{\boldsymbol{\tau}}(x^{(j)}, y_1^{(j)}, y_2^{(j)}). \tag{2.3}$$

We assume that the proximal maps appearing in the updates can be efficiently executed to sufficient accuracy. Further discussion of this point is delayed until subsection 2.5.

Algorithm 2.1 Inner iterations (primal-dual updates) that are used in WARPd. At any one time, only the current ergodic average, two primal and two dual variables need to be stored.

Input: Initial vector $x_0 \in \mathcal{X}$, proximal step sizes $\tau_1 > 0$, $\tau_2 > 0$, and $\tau_3 > 0$, number of iterations $k \in \mathbb{N}$, data $b \in \mathcal{Y}_1^*$, $\epsilon > 0$, $\mathcal{J}$, $\mathcal{F}^*$, A, and B.

1: Initiate with $x^{(0)} = x_0$, $y_1^{(0)} = 0 \in \mathcal{Y}_1$, $y_2^{(0)} = 0 \in \mathcal{Y}_2$, and $X_0 = 0 \in \mathcal{X}$.

2: For $j = 0, \dots, k-1$ compute

$$
\begin{aligned}
x^{(j+1)} &= \operatorname{argmin}_{x \in \mathcal{X}} \mathcal{J}(x) + \langle Ax, y_1^{(j)} \rangle_{\mathbb{R}} + \langle Bx, y_2^{(j)} \rangle_{\mathbb{R}} + \frac{1}{\tau_1} D_{\mathcal{X}}(x, x^{(j)}), \\
y_1^{(j+1)} &= \operatorname{argmin}_{y_1 \in \mathcal{Y}_1} \epsilon \|y_1\|_{\mathcal{Y}_1} - \langle A(2x^{(j+1)} - x^{(j)}) - b, y_1 \rangle_{\mathbb{R}} + \frac{1}{\tau_2} D_{\mathcal{Y}_1}(y_1, y_1^{(j)}), \\
y_2^{(j+1)} &= \operatorname{argmin}_{y_2 \in \mathcal{Y}_2} \mathcal{F}^*(y_2) - \langle B(2x^{(j+1)} - x^{(j)}), y_2 \rangle_{\mathbb{R}} + \frac{1}{\tau_3} D_{\mathcal{Y}_2}(y_2, y_2^{(j)}),
\end{aligned} \tag{2.5}
$$

and update the ergodic average $X_{j+1} = \frac{1}{j+1}\left(jX_j + x^{(j+1)}\right)$.

Output: $\texttt{InnerIt}\,(x_0, \tau_1, \tau_2, \tau_3, k; b, \epsilon, \mathcal{J}, \mathcal{F}^*, A, B) = X_k$.

For notational convenience, we define

$$
G_\eta(\hat{x}, x, b) := \underbrace{\mathcal{J}(\hat{x}) + \mathcal{F}(B\hat{x}) - \mathcal{J}(x) - \mathcal{F}(Bx)}_{\text{objective function difference}} + \eta \underbrace{\left(\|A\hat{x} - b\|_{\mathcal{Y}_1^*} - \epsilon\right)}_{\text{feasibility gap}} \tag{2.4}
$$

for a multiplier $\eta \geq 0$. Note that the inequality (1.5) allows us to bound $\sqrt{2D_{\mathcal{X}}(\mathbb{x}, \hat{x})}$ in terms of $G_{C_2}(\hat{x}, \mathbb{x}, b)$ and $c(\mathbb{x}, b)$. Hence we would like to control the size of $G_{C_2}(\hat{x}, \mathbb{x}, b)$. As a first step, Proposition 2.1 provides an explicit bound for G_η (see (2.6)) for exact primal-dual updates without restarts.

Proposition 2.1 (bounds on G_η for primal-dual updates). *Suppose that the step sizes τ_1, τ_2, and τ_3 satisfy $\tau_1(\tau_2\|A\|^2 + \tau_3\|B\|^2) < 1$. Let $x_0 \in \mathcal{X}$, $y_1^{(0)} = 0 \in \mathcal{Y}_1$, $y_2^{(0)} = 0 \in \mathcal{Y}_2$, and*

$$
(x^{(j+1)}, y_1^{(j+1)}, y_2^{(j+1)}) = \mathrm{PD}_{\boldsymbol{\tau}}(x^{(j)}, y_1^{(j)}, y_2^{(j)}), \quad j = 0, \dots, k-1.
$$

Define the ergodic averages

$$
X_k = \frac{1}{k}\sum_{j=1}^{k} x^{(j)}, \quad [Y_k]_1 = \frac{1}{k}\sum_{j=1}^{k} y_1^{(j)}, \quad [Y_k]_2 = \frac{1}{k}\sum_{j=1}^{k} y_2^{(j)}.
$$

Then for any $\eta \geq 0$ and any feasible $x \in \mathcal{X}$ (i.e., such that $\|Ax - b\|_{\mathcal{Y}_1^*} \leq \epsilon$),

$$
G_\eta(X_k, x, b) \leq \frac{2}{k}\left(\frac{D_{\mathcal{X}}(x, x^{(0)})}{\tau_1} + \frac{g_1(\eta)}{\tau_2} + \frac{g_2(1)}{\tau_3}\right), \tag{2.6}
$$

where g_1 and g_2 satisfy (1.3) and (1.4), respectively.

Proof. Recall the definition of $\mathcal{L}$ in (2.2). Since $\tau_1(\tau_2\|A\|^2 + \tau_3\|B\|^2) < 1$, a simple adaptation of [41, Theorem 1, Remark 2] shows that for any $x \in \mathcal{X}$, $y_1 \in \mathcal{Y}_1$, and $y_2 \in \mathcal{Y}_2$,

$$\mathcal{L}(X_k, y_1, y_2) - \mathcal{L}(x, [Y_k]_1, [Y_k]_2) \le \frac{2}{k}\left(\frac{D_{\mathcal{X}}(x, x^{(0)})}{\tau_1} + \frac{D_{\mathcal{Y}_1}(y_1, 0)}{\tau_2} + \frac{D_{\mathcal{Y}_2}(y_2, 0)}{\tau_3}\right). \tag{2.7}$$

Let x be feasible and $y_1 = \eta y_1'$, where y_1' is any unit-norm vector such that $\|AX_k - b\|_{\mathcal{Y}_1^*} = \langle AX_k - b, y_1'\rangle_{\mathbb{R}}$. Writing out the difference on the left-hand side of (2.7), and recalling (1.3),

$$\begin{aligned}&\eta\left(\|AX_k - b\|_{\mathcal{Y}_1^*} - \epsilon\right) + \mathcal{J}(X_k) - \mathcal{J}(x)\\ &- \langle Ax - b, [Y_k]_1\rangle_{\mathbb{R}} + \epsilon\|[Y_k]_1\|_{\mathcal{Y}_1} - \mathcal{F}^*(y_2)\\ &+ \langle BX_k, y_2\rangle_{\mathbb{R}} - \langle Bx, [Y_k]_2\rangle_{\mathbb{R}} + \mathcal{F}^*([Y_k]_2) \le \frac{2}{k}\left(\frac{D_{\mathcal{X}}(x, x^{(0)})}{\tau_1} + \frac{g_1(\eta)}{\tau_2} + \frac{D_{\mathcal{Y}_2}(y_2, 0)}{\tau_3}\right).\end{aligned} \tag{2.8}$$

We now take the supremum of the left-hand side of (2.8) over y_2 with $D_{\mathcal{Y}_2}(y_2, 0) \le g_2(1)$ and recall that $\mathcal{F}$ satisfies (1.4). Moreover, x is feasible so that $\|Ax - b\|_{\mathcal{Y}_1^*} \le \epsilon$ and hence $0 \le -\langle Ax - b, [Y_k]_1\rangle_{\mathbb{R}} + \epsilon\|[Y_k]_1\|_{\mathcal{Y}_1}$. It follows from (2.8) that

$$\begin{aligned}&\eta\left(\|AX_k - b\|_{\mathcal{Y}_1^*} - \epsilon\right) + \mathcal{J}(X_k) + \mathcal{F}(BX_k)\\ &- \mathcal{J}(x) - \langle Bx, [Y_k]_2\rangle_{\mathbb{R}} + \mathcal{F}^*([Y_k]_2) \le \frac{2}{k}\left(\frac{D_{\mathcal{X}}(x, x^{(0)})}{\tau_1} + \frac{g_1(\eta)}{\tau_2} + \frac{g_2(1)}{\tau_3}\right).\end{aligned} \tag{2.9}$$

Since $-\mathcal{F}(Bx) \le -\langle Bx, [Y_k]_2\rangle_{\mathbb{R}} + \mathcal{F}^*([Y_k]_2)$, (2.9) yields (2.6). ∎

2.2. Primal-dual iterations: The Hilbert space case. In the case that $\mathcal{X}$, $\mathcal{Y}_1$, and $\mathcal{Y}_2$ are Hilbert spaces and that $D_{\mathcal{Z}}(z, \hat{z}) = \|z - \hat{z}\|_{\mathcal{Z}}^2/2$ for $\mathcal{Z} = \mathcal{X}$, $\mathcal{Y}_1$, and $\mathcal{Y}_2$, the primal-dual iterations are simplified. In particular, we can absorb the bilinear forms into the Bregman distances in (2.5). Algorithm 2.2 summarizes the iterations, where

$$\gamma_\rho(y_1) := y_1 - \rho\mathcal{P}_{\mathcal{Y}_1}(y_1/\rho) = \operatorname{prox}_{\rho\|\cdot\|_{\mathcal{Y}_1}}(y_1), \tag{2.10}$$

and we recall that $\mathcal{P}_{\mathcal{Z}}$ denotes the projection onto the unit ball of $\mathcal{Z}$. Moreover, we can bound the error (see (2.12)) due to inexact primal-dual updates without restarts. We treat inexact updates to provide stability results, cover the case of inexact information regarding A and B, and cover cases where the proximal map of $\mathcal{J}$ is applied approximately (see subsection 5.2).

Proposition 2.2 (perturbation bound for inexact primal-dual updates). *Suppose that $\mathcal{X}$, $\mathcal{Y}_1$, and $\mathcal{Y}_2$ are Hilbert spaces and that the step sizes τ_1, τ_2, and τ_3 satisfy $\tau_1(\tau_2\|A\|^2 + \tau_3\|B\|^2) < 1$. Let $x_0 \in \mathcal{X}$ and set $\tilde{x}^{(0)} = x^{(0)}$ and $\tilde{y}_i^{(0)} = y_i^{(0)} = 0 \in \mathcal{Y}_i$ for $i = 1, 2$. Suppose that $\tilde{x}^{(j)} \in \mathcal{X}$, $\tilde{y}_1^{(j)} \in \mathcal{Y}_1$, and $\tilde{y}_2^{(j)} \in \mathcal{Y}_2$ are such that*

$$\left\|(\tilde{x}^{(j)}, \tilde{y}_1^{(j)}, \tilde{y}_2^{(j)}) - \mathrm{PD}_{\boldsymbol{\tau}}(\tilde{x}^{(j-1)}, \tilde{y}_1^{(j-1)}, \tilde{y}_2^{(j-1)})\right\|_{\mathcal{X}\oplus\mathcal{Y}_1\oplus\mathcal{Y}_2} \le \delta_j, \quad j = 0, \dots, k-1. \tag{2.11}$$

In other words, the jth primal-dual iterate is approximately computed to accuracy δ_j. Let

$$(x^{(j+1)}, y_1^{(j+1)}, y_2^{(j+1)}) = \mathrm{PD}_{\boldsymbol{\tau}}(x^{(j)}, y_1^{(j)}, y_2^{(j)}), \quad j = 0, \dots, k-1,$$

Algorithm 2.2 Inner iterations (primal-dual updates) that are used in WARPd for the Hilbert space case. At any one time, only the current ergodic average, two primal and two dual variables need to be stored. The function γ_ρ is defined in (2.10).

Input: Initial vector $x_0 \in \mathcal{X}$, proximal step sizes $\tau_1 > 0$, $\tau_2 > 0$ and $\tau_3 > 0$, number of iterations $k \in \mathbb{N}$, data $b \in \mathcal{Y}_1^*$, $\epsilon > 0$, $\mathcal{J}$, $\mathcal{F}^*$, A, and B.

1: Initiate with $x^{(0)} = x_0$, $y_1^{(0)} = 0 \in \mathcal{Y}_1$, $y_2^{(0)} = 0 \in \mathcal{Y}_2$ and $X_0 = 0$.
2: For $j = 0, \dots, k-1$ compute

$$\begin{aligned} x^{(j+1)} &= \operatorname{prox}_{\tau_1 \mathcal{J}}(x^{(j)} - \tau_1 A^* y_1^{(j)} - \tau_1 B^* y_2^{(j)}), \\ y_1^{(j+1)} &= \gamma_{\tau_2 \epsilon}\left(y_1^{(j)} + \tau_2 A(2x^{(j+1)} - x^{(j)}) - \tau_2 b\right), \\ y_2^{(j+1)} &= \operatorname{prox}_{\tau_3 \mathcal{F}^*}(y_2^{(j)} + \tau_3 B(2x^{(j+1)} - x^{(j)})), \end{aligned}$$

and update the ergodic average $X_{j+1} = \frac{1}{j+1}\left(jX_j + x^{(j+1)}\right)$.

Output: $\texttt{InnerIt}\,(x_0, \tau_1, \tau_2, \tau_3, k; b, \epsilon, \mathcal{J}, \mathcal{F}^*, A, B) = X_k$.

denote the exact primal-dual iterates and define the ergodic averages

$$X_k = \frac{1}{k}\sum_{j=1}^{k} x^{(j)}, \quad \tilde{X}_k = \frac{1}{k}\sum_{j=1}^{k} \tilde{x}^{(j)}, \quad Y_k = \frac{1}{k}\sum_{j=1}^{k} (y_1^{(j)}, y_2^{(j)}), \quad \tilde{Y}_k = \frac{1}{k}\sum_{j=1}^{k} (\tilde{y}_1^{(j)}, \tilde{y}_2^{(j)}).$$

Then

$$\left\|(X_k, Y_k) - (\tilde{X}_k, \tilde{Y}_k)\right\|_{\mathcal{X}\oplus\mathcal{Y}_1\oplus\mathcal{Y}_2} \le \sqrt{\frac{\max\{\tau_1,\tau_2,\tau_3\}\,[1+\tau_1(\tau_2\|A\|^2+\tau_3\|B\|^2)]}{\min\{\tau_1,\tau_2,\tau_3\}\,[1-\tau_1(\tau_2\|A\|^2+\tau_3\|B\|^2)]}} \sum_{j=1}^{k}\sum_{i=1}^{j} \frac{\delta_i}{k}. \tag{2.12}$$

Proof. Let $v = (x, y_1, y_2)$, and define the following operators acting on $\mathcal{X} \oplus \mathcal{Y}_1 \oplus \mathcal{Y}_2$:

$$M_{\boldsymbol{\tau}} = \begin{pmatrix} \frac{1}{\tau_1} I_{\mathcal{X}} & -A^* & -B^* \\ -A & \frac{1}{\tau_2} I_{\mathcal{Y}_1} & 0 \\ -B & 0 & \frac{1}{\tau_3} I_{\mathcal{Y}_2} \end{pmatrix}, \quad P_{\boldsymbol{\tau}} = \begin{pmatrix} I_{\mathcal{X}} & -A^*\sqrt{\tau_1\tau_2} & -B^*\sqrt{\tau_1\tau_3} \\ -A\sqrt{\tau_1\tau_2} & I_{\mathcal{Y}_1} & 0 \\ -B\sqrt{\tau_1\tau_3} & 0 & I_{\mathcal{Y}_2} \end{pmatrix},$$

where $I_{\mathcal{Z}}$ denotes the identity operator on $\mathcal{Z}$. The operators $M_{\boldsymbol{\tau}}$ and $P_{\boldsymbol{\tau}}$ are related via $P_{\boldsymbol{\tau}} = D_{\boldsymbol{\tau}} M_{\boldsymbol{\tau}} D_{\boldsymbol{\tau}}$, where $D_{\boldsymbol{\tau}} = \sqrt{\tau_1} I_{\mathcal{X}} \oplus \sqrt{\tau_2} I_{\mathcal{Y}_1} \oplus \sqrt{\tau_3} I_{\mathcal{Y}_2}$. An application of the AM–GM inequality shows that $\|P_{\boldsymbol{\tau}} - I\| \le \tau_1(\tau_2\|A\|^2 + \tau_3\|B\|^2)$. In particular, a Neumann series argument shows that $\|P_{\boldsymbol{\tau}}^{\pm 1}\| \le (1 \pm \tau_1(\tau_2\|A\|^2 + \tau_3\|B\|^2))^{\pm 1}$. Since $\tau_1(\tau_2\|A\|^2 + \tau_3\|B\|^2) < 1$, it follows that $M_{\boldsymbol{\tau}}$ is positive definite (as well as being bounded) and thus induces a norm $\|v\|_{\boldsymbol{\tau}} := \|M_{\boldsymbol{\tau}}^{1/2} v\|_{\mathcal{X}\oplus\mathcal{Y}_1\oplus\mathcal{Y}_2}$. Moreover, since $\|D_{\boldsymbol{\tau}}\|^2 = \max\{\tau_1,\tau_2,\tau_3\}$ and $\|D_{\boldsymbol{\tau}}^{-1}\|^2 = \max\{\tau_1^{-1},\tau_2^{-1},\tau_3^{-1}\}$,

$$\begin{aligned} \|v\|_{\boldsymbol{\tau}} &\le \sqrt{\max\{\tau_1^{-1},\tau_2^{-1},\tau_3^{-1}\}\,[1+\tau_1(\tau_2\|A\|^2+\tau_3\|B\|^2)]} \cdot \|v\|_{\mathcal{X}\oplus\mathcal{Y}_1\oplus\mathcal{Y}_2} \quad \text{and} \\ \|v\|_{\mathcal{X}\oplus\mathcal{Y}_1\oplus\mathcal{Y}_2} &\le \sqrt{\max\{\tau_1,\tau_2,\tau_3\}\,[1-\tau_1(\tau_2\|A\|^2+\tau_3\|B\|^2)]^{-1}} \cdot \|v\|_{\boldsymbol{\tau}}. \end{aligned} \tag{2.13}$$

We can write the iterations defined by $\mathrm{PD}_{\boldsymbol{\tau}}$ as

$$0 \in \mathcal{T} v^{(j+1)} + (v^{(j+1)} - v^{(j)}) \quad \text{with} \quad \mathcal{T} := M_{\boldsymbol{\tau}}^{-1} \begin{pmatrix} \partial \mathcal{J} & A^* & B^* \\ -A & \partial h^* & 0 \\ -B & 0 & \partial \mathcal{F}^* \end{pmatrix},$$

where $h^*(y_1) = \epsilon \|y_1\|_{\mathcal{Y}_1} + \langle b, y_1 \rangle_{\mathbb{R}}$. It follows that $v^{(j+1)} = [I + \mathcal{T}]^{-1} v^{(j)}$. The operator $\mathcal{T}$ is maximal monotone with respect to the inner product induced by $M_{\boldsymbol{\tau}}$ [108], and hence the iterations are nonexpansive in the norm $\|\cdot\|_{\boldsymbol{\tau}}$. It follows that

$$\begin{aligned} \|v^{(j)} - \tilde{v}^{(j)}\|_{\boldsymbol{\tau}} &\leq \|\tilde{v}^{(j)} - \mathrm{PD}_{\boldsymbol{\tau}}(\tilde{v}^{(j-1)})\|_{\boldsymbol{\tau}} + \|\mathrm{PD}_{\boldsymbol{\tau}}(v^{(j-1)}) - \mathrm{PD}_{\boldsymbol{\tau}}(\tilde{v}^{(j-1)})\|_{\boldsymbol{\tau}} \\ &\leq \|\tilde{v}^{(j)} - \mathrm{PD}_{\boldsymbol{\tau}}(\tilde{v}^{(j-1)})\|_{\boldsymbol{\tau}} + \|v^{(j-1)} - \tilde{v}^{(j-1)}\|_{\boldsymbol{\tau}} \\ &\leq \delta_j \sqrt{\max\{\tau_1^{-1}, \tau_2^{-1}, \tau_3^{-1}\} \left[1 + \tau_1(\tau_2 \|A\|^2 + \tau_3 \|B\|^2)\right]} + \|v^{(j-1)} - \tilde{v}^{(j-1)}\|_{\boldsymbol{\tau}}, \end{aligned}$$

where we have used the triangle inequality in the first inequality, the fact that the iterates are nonexpansive in $\|\cdot\|_{\boldsymbol{\tau}}$ in the second inequality, and (2.13) in the final inequality. Iterating and using (2.13) once more, we have

$$\begin{aligned} \|v^{(j)} - \tilde{v}^{(j)}\|_{\mathcal{X} \oplus \mathcal{Y}_1 \oplus \mathcal{Y}_2} &\leq \sqrt{\frac{\max\{\tau_1, \tau_2, \tau_3\}}{[1 - \tau_1(\tau_2 \|A\|^2 + \tau_3 \|B\|^2)]}} \|v^{(j)} - \tilde{v}^{(j)}\|_{\boldsymbol{\tau}} \\ &\leq \sqrt{\frac{\max\{\tau_1, \tau_2, \tau_3\} \left[1 + \tau_1(\tau_2 \|A\|^2 + \tau_3 \|B\|^2)\right]}{\min\{\tau_1, \tau_2, \tau_3\} \left[1 - \tau_1(\tau_2 \|A\|^2 + \tau_3 \|B\|^2)\right]}} \sum_{i=1}^{j} \delta_i. \end{aligned}$$

Summing this bound for the ergodic averages reduces to (2.12). ∎

For different notions of inexact primal-dual iterations and their effect on convergence rates in the Hilbert space case, see [103]. Our scenario is different since we only use a fixed number of iterations per restart and provide convergence to order $\delta > 0$ in (1.7), thus allowing the δ_j to be bounded below. In particular, we do not require a decay of errors (see (2.26)).

2.3. The restart scheme. We now describe the accelerated scheme. The idea is to take advantage of the different proximal step sizes on the right-hand side of (2.6). Together with (1.5), this allows a decrease in the relevant gap G_η (defined in (2.4)) by a constant factor for a fixed number of iterations. An interpretation is that we *reweight* the sizes of the proximal steps at each restart as $D_{\mathcal{X}}(\mathbb{x}, x)$ decreases. The full algorithm is described in Algorithm 2.3, where, for simplicity, we have written the algorithm for exact primal-dual iterations. Theorem 1.2 summarizes the convergence result.

Proof of Theorem 1.2. We prove the theorem under the more general conditions of inexact primal-dual iterations. Consider the setup in the statement of Theorem 1.2. Let $\psi_k = \texttt{InnerIt}(x_0, \tau_1, \tau_2, \tau_3, k)$ denote the exact updates described in Algorithm 2.1 (or Algorithm 2.2). Suppose that the initial starting vector x_0 satisfies $\sqrt{2D_{\mathcal{X}}(\mathbb{x}, x_0)} \leq C_1(\delta + \omega)$ for some $\omega > 0$. Combining this with (2.6) for the choice $\eta = C_2$, we have

$$G_{C_2}(\psi_k, \mathbb{x}, b) \leq \frac{1}{k} \left(\frac{C_1^2(\delta + \omega)^2}{\tau_1} + \frac{2g_1(C_2)}{\tau_2} + \frac{2g_2(1)}{\tau_3} \right). \tag{2.14}$$

Algorithm 2.3 (WARPd) Accelerated algorithm for the solution of (1.2) and recovery of desired vector. The updates in (2.15) correspond to restarted primal-dual iterations performed by the routine `InnerIt` in Algorithm 2.1 (general case) or Algorithm 2.2 (Hilbert space case).

Input: C_1 and C_2 such that (1.5) holds, g_1 such that (1.3) holds, g_2 such that (1.4) holds, L_A and L_B (upper bounds for $\|A\|$ and $\|B\|$, respectively), $\tau \in (0,1)$, $\epsilon > 0$, $\delta > 0$, $\mathcal{J}$, $\mathcal{F}^*$, A, B, $b \in \mathcal{Y}_1^*$, and $n \in \mathbb{N}$.

1: Set $\omega_0 = \mathcal{J}(0) + \mathcal{F}(0) + C_2\|b\|_{\mathcal{Y}_1^*}$ (or an upper bound) and compute $\omega_j = e^{-1}(\delta + \omega_{j-1})$ for $j = 1, \dots, n-1$.

2: Set $k = \lceil 2C_1 e(L_A\sqrt{2g_1(C_2)} + L_B\sqrt{2g_2(1)})/\tau \rceil$.

3: For $j = 0, \dots, n-1$ compute

$$\tau_1^{(j)} = \frac{\tau C_1(\delta + \omega_j)}{L_A\sqrt{2g_1(C_2)} + L_B\sqrt{2g_2(1)}}, \quad \tau_2^{(j)} = \frac{\tau\sqrt{2g_1(C_2)}}{L_A C_1(\delta + \omega_j)}, \quad \tau_3^{(j)} = \frac{\tau\sqrt{2g_2(1)}}{L_B C_1(\delta + \omega_j)}.$$

4: Set $\phi_0(b) = 0$ (or any other initial approximation) and for $j = 1, \dots, n$, compute

$$\phi_j(b) = \texttt{InnerIt}\left(\phi_{j-1}(b), \tau_1^{(j-1)}, \tau_2^{(j-1)}, \tau_3^{(j-1)}, k; b, \epsilon, \mathcal{J}, \mathcal{F}^*, A, B\right). \tag{2.15}$$

Output: $\phi_n(b) \in \mathcal{X}$.

Let $\tau \in (0,1)$, and suppose that $\tau_1(\tau_2 L_A^2 + \tau_3 L_B^2) = \tau^2$. In this case, the optimal choices of τ_1, τ_2, and τ_3 that minimize the right-hand side of (2.14) are

$$\tau_1(\omega) = \frac{\tau C_1(\delta + \omega)}{L_A\sqrt{2g_1(C_2)} + L_B\sqrt{2g_2(1)}}, \quad \tau_2(\omega) = \frac{\tau\sqrt{2g_1(C_2)}}{L_A C_1(\delta + \omega)}, \quad \tau_3(\omega) = \frac{\tau\sqrt{2g_2(1)}}{L_B C_1(\delta + \omega)}.$$

With this choice, we have that

$$G_{C_2}(\psi_k, \mathbb{x}, b) \le \frac{2C_1}{k}\left(\frac{L_A\sqrt{2g_1(C_2)} + L_B\sqrt{2g_2(1)}}{\tau}\right)(\delta + \omega). \tag{2.16}$$

For $\nu \in (0,1)$ that we optimize later, set $k = \lceil 2C_1(L_A\sqrt{2g_1(C_2)} + L_B\sqrt{2g_2(1)})/(\nu\tau) \rceil$ so that (2.16) implies that $G_{C_2}(\psi_k, \mathbb{x}, b) \le \nu(\delta + \omega)$.

Suppose now that instead of ψ_k, we compute $\Psi_k = \Psi_k(x_0, \tau_1, \tau_2, \tau_3) \in \mathcal{X}$ with

$$\sqrt{2D_{\mathcal{X}}(\mathbb{x}, \Psi_k)} \le \sqrt{2D_{\mathcal{X}}(\mathbb{x}, \psi_k)} + \alpha C_1 \delta\nu \tag{2.17}$$

for some accuracy parameter $\alpha > 0$. Since $c(\mathbb{x}, b) \le \delta$, it follows from (1.5) that

$$\sqrt{2D_{\mathcal{X}}(\mathbb{x}, \Psi_k)} \le C_1(\delta + \nu(\delta + \alpha\delta + \omega)).$$

We now describe the restart scheme. From (1.5) and the assumption that $\mathcal{J}(\cdot) + \mathcal{F}(B\cdot) \ge 0$,

$$G_{C_2}(0, \mathbb{x}, b) = \mathcal{J}(0) + \mathcal{F}(0) - \mathcal{J}(\mathbb{x}) - \mathcal{F}(B\mathbb{x}) + C_2(\|b\|_{\mathcal{Y}_1^*} - \epsilon) \le \mathcal{J}(0) + \mathcal{F}(0) + C_2\|b\|_{\mathcal{Y}_1^*}.$$

It follows from (1.5) that $\sqrt{2D_{\mathcal{X}}(\mathbb{x},0)} \leq C_1(\delta+\omega_0)$ with $\omega_0 = \mathcal{J}(0)+\mathcal{F}(0)+C_2\|b\|_{\mathcal{Y}_1^*}$. Given $n \in \mathbb{N}$, set $\omega_j = \nu\left(\delta+\alpha\delta+\omega_{j-1}\right)$ for $j = 1,\dots,n-1$. By summing a geometric series, this implies that $\omega_n \leq \frac{\nu\delta(1+\alpha)}{1-\nu} + \nu^n[\mathcal{J}(0)+\mathcal{F}(0)+C_2\|b\|_{\mathcal{Y}_1^*}]$. We define $\phi_n(b)$ iteratively via

$$\phi_1(b) = \Psi_k\left(0,\tau_1(\omega_0),\tau_2(\omega_0),\tau_3(\omega_0)\right), \quad \phi_j(b) = \Psi_k\left(\phi_{j-1}(b),\tau_1(\omega_{j-1}),\tau_2(\omega_{j-1}),\tau_3(\omega_{j-1})\right),$$

for $j = 1,\dots,n$. The choice of ω_j and the above argument inductively show that

$$\sqrt{2D_{\mathcal{X}}(\mathbb{x},\phi_n(b))} \leq C_1(\delta+\omega_n) \leq C_1\left(\delta + \frac{\nu\delta(1+\alpha)}{1-\nu} + \nu^n\left[\mathcal{J}(0)+\mathcal{F}(0)+C_2\|b\|_{\mathcal{Y}_1^*}\right]\right).$$

For $T = kn$ inner iterations, the error term ν^n is equal to $\exp(Tk^{-1}\log(\nu))$. If we ignore the ceiling function in the choice of k, the optimal choice of $\nu = e^{-1}$ is found via differentiation. This choice yields (1.7) in the limit $\alpha \downarrow 0$ (i.e., for exact primal-dual iterations). ∎

2.4. Noise-blind recovery: Replacing the constraint with a data fitting term. We now discuss a variation of WARPd based on the following *unconstrained* optimization problem:

$$\min_{x\in\mathcal{X}} \lambda\left[\mathcal{J}(x)+\mathcal{F}(Bx)\right] + \|Ax-b\|_{\mathcal{Y}_1^*} \quad (\text{with } \lambda > 0). \tag{2.18}$$

The optimization problem in (2.18) differs from its LASSO-type cousin by replacing the conventional $\|Ax-b\|_{\mathcal{Y}_1^*}^2$ term with $\|Ax-b\|_{\mathcal{Y}_1^*}$. In the case of sparse recovery (for the finite-dimensional Hilbert spaces $\mathcal{X} = \mathbb{C}^N$ and $\mathcal{Y}_1 = \mathbb{C}^m$, $\mathcal{J}(x) = \|x\|_{l^1_w}$, $\mathcal{F} = 0$) in section 3, this is known as the square-root LASSO (SR-LASSO) decoder. SR-LASSO was introduced in [19]; see also [1, 20]. In particular, SR-LASSO allows an optimal parameter choice for λ that is *independent of the noise level* [6, Table 6.1] and is therefore well suited to noise-blind recovery problems. This property also holds for the algorithm we describe, WARPd-SR. Moreover, there is an additional benefit. Using (2.18) allows an elegant bound on the size of dual variables, and hence allows an easier analysis with additional dual variable restarts (see the discussion at the end of subsection 2.5 and in section SM2).

Throughout this section, we replace the assumption (1.5) by

$$\sqrt{2D_{\mathcal{X}}(\mathbb{x},\hat{x})} \leq \hat{C}_1\Big[\mathcal{J}(\hat{x})+\mathcal{F}(B\hat{x})-\mathcal{J}(\mathbb{x})-\mathcal{F}(B\mathbb{x})+\hat{C}_2\left(\|A\hat{x}-b\|_{\mathcal{Y}_1^*}-\|A\mathbb{x}-b\|_{\mathcal{Y}_1^*}\right)+\hat{c}(\mathbb{x},b)\Big]. \tag{2.19}$$

In practice, the assumptions (1.5) and (2.19) are equivalent up to a change in $c(\mathbb{x},b)$ and $\hat{c}(\mathbb{x},b)$. For example, if (1.5) holds, then (2.19) holds with $C_j = \hat{C}_j$ and $\hat{c}(\mathbb{x},b) = c(\mathbb{x},b) + C_2(\|A\mathbb{x}-b\|_{\mathcal{Y}_1^*}-\epsilon)$. The similarity is also reflected in the proofs for the examples we give in later sections—we typically prove (1.5) via (2.19). We further assume that

$$g_1(1) < \infty, \quad g_2(1/\hat{C}_2) < \infty, \tag{2.20}$$

where g_1 and g_2 are the functions in (1.3) and (1.4). To analyze the problem, we proceed as in subsection 2.1. The problem (2.18) is equivalent to the saddle point problem

$$\inf_{x\in\mathcal{X}} \sup_{y_1\in\mathcal{Y}_1, y_2\in\mathcal{Y}_2} \Big[\widehat{\mathcal{L}}(x,y_1,y_2) := \langle Ax-b,y_1\rangle_{\mathbb{R}} + \langle Bx,y_2\rangle_{\mathbb{R}} + \lambda\mathcal{J}(x) - \chi_{\mathcal{B}_{\mathcal{Y}_1}}(y_1) - \lambda\mathcal{F}^*(y_2/\lambda)\Big], \tag{2.21}$$

Algorithm 2.4 Inner iterations (primal-dual updates) that are used in WARPd-SR. At any one time, only the current ergodic average, two primal and two dual variables need to be stored.

Input: Initial vector $x_0 \in \mathcal{X}$, proximal step sizes $\tau_1 > 0$, $\tau_2 > 0$, and $\tau_3 > 0$, number of iterations $k \in \mathbb{N}$, data $b \in \mathcal{Y}_1^*$, $\lambda > 0$, $\mathcal{J}$, $\mathcal{F}^*$, A, and B.

1: Initiate with $x^{(0)} = x_0$, $y_1^{(0)} = 0 \in \mathcal{Y}_1$, $y_2^{(0)} = 0 \in \mathcal{Y}_2$, and $X_0 = 0 \in \mathcal{X}$.
2: For $j = 0, \dots, k-1$ compute

$$
\begin{aligned}
x^{(j+1)} &= \operatorname{argmin}_{x \in \mathcal{X}} \lambda \mathcal{J}(x) + \langle Ax, y_1^{(j)} \rangle_{\mathbb{R}} + \langle Bx, y_2^{(j)} \rangle_{\mathbb{R}} + \frac{1}{\tau_1} D_{\mathcal{X}}(x, x^{(j)}), \\
y_1^{(j+1)} &= \operatorname{argmin}_{y_1 \in \mathcal{Y}_1} \chi_{\mathcal{B}_{\mathcal{Y}_1}}(y_1) - \langle A(2x^{(j+1)} - x^{(j)}) - b, y_1 \rangle_{\mathbb{R}} + \frac{1}{\tau_2} D_{\mathcal{Y}_1}(y_1, y_1^{(j)}), \\
y_2^{(j+1)} &= \operatorname{argmin}_{y_2 \in \mathcal{Y}_2} \lambda \mathcal{F}^*(y_2/\lambda) - \langle B(2x^{(j+1)} - x^{(j)}), y_2 \rangle_{\mathbb{R}} + \frac{1}{\tau_3} D_{\mathcal{Y}_2}(y_2, y_2^{(j)}),
\end{aligned} \tag{2.22}
$$

and update the ergodic average $X_{j+1} = \frac{1}{j+1}\left(jX_j + x^{(j+1)}\right)$.

Output: $\texttt{InnerIt-SR}\,(x_0, \tau_1, \tau_2, \tau_3, k; b, \lambda, \mathcal{J}, \mathcal{F}^*, A, B) = X_k$.

Algorithm 2.5 Inner iterations (primal-dual updates) that are used in WARPd-SR for the Hilbert space case. At any one time, only the current ergodic average, two primal and two dual variables need to be stored. $\mathcal{P}_{\mathcal{Y}_1}$ denotes the projection onto the unit ball of $\mathcal{Y}_1$.

Input: Initial vector $x_0 \in \mathcal{X}$, proximal step sizes $\tau_1 > 0$, $\tau_2 > 0$, and $\tau_3 > 0$, number of iterations $k \in \mathbb{N}$, data $b \in \mathcal{Y}_1^*$, $\lambda > 0$, $\mathcal{J}$, $\mathcal{F}^*$, A, and B.

1: Initiate with $x^{(0)} = x_0$, $y_1^{(0)} = 0 \in \mathcal{Y}_1$, $y_2^{(0)} = 0 \in \mathcal{Y}_2$, and $X_0 = 0 \in \mathcal{X}$.
2: For $j = 0, \dots, k-1$ compute

$$
\begin{aligned}
x^{(j+1)} &= \operatorname{prox}_{\lambda\tau_1 \mathcal{J}}\left(x^{(j)} - \tau_1 A^* y_1^{(j)} - \tau_1 B^* y_2^{(j)}\right), \\
y_1^{(j+1)} &= \mathcal{P}_{\mathcal{Y}_1}\left(y_1^{(j)} + \tau_2 A(2x^{(j+1)} - x^{(j)}) - \tau_2 b\right), \\
y_2^{(j+1)} &= \operatorname{prox}_{\tau_3[\lambda\mathcal{F}]^*}\left(y_2^{(j)} + \tau_3 B(2x^{(j+1)} - x^{(j)})\right),
\end{aligned}
$$

and update the ergodic average $X_{j+1} = \frac{1}{j+1}\left(jX_j + x^{(j+1)}\right)$.

Output: $\texttt{InnerIt-SR}\,(x_0, \tau_1, \tau_2, \tau_3, k; b, \lambda, \mathcal{J}, \mathcal{F}^*, A, B) = X_k$.

where we recall that $\mathcal{B}_{\mathcal{Z}}$ denotes the unit ball of a Banach space $\mathcal{Z}$. The primal-dual iterations for this saddle point problem are described in Algorithm 2.4 (general case) and Algorithm 2.5 (Hilbert space case). The accelerated restart scheme is summarized in Algorithm 2.6 with $\lambda = 1/\hat{C}_2$. The following theorem describes the convergence. Note that WARPd-SR does not need any ϵ (noise level) parameter as an input.

Theorem 2.3. *Let L_A and L_B be upper bounds for $\|A\|$ and $\|B\|$, respectively, and $\delta > 0$. Then for any $n \in \mathbb{N}$ and any pair $(\mathtt{x}, b) \in \mathcal{X} \times \mathcal{Y}_1^*$ such that (2.19) holds with $\hat{c}(\mathtt{x}, b) \le \delta$, the*

Algorithm 2.6 (WARPd-SR): Accelerated algorithm for the solution of (2.18) and recovery of desired vector. The updates in (2.23) correspond to restarted primal-dual iterations performed by the routine `InnerIt-SR` in Algorithm 2.4 (general case) or Algorithm 2.5 (Hilbert space case).

Input: $\hat{C}_1$ and $\hat{C}_2$ such that (2.19) holds, g_1 such that (1.3) holds, g_2 such that (1.4) holds, L_A and L_B (upper bounds for $\|A\|$ and $\|B\|$, respectively), $\tau \in (0,1)$, $\delta > 0$, $\mathcal{J}$, $\mathcal{F}^*$, A, B, $b \in \mathcal{Y}_1^*$, and $n \in \mathbb{N}$.

1: Set $\omega_0 = \mathcal{J}(0) + \mathcal{F}(0) + \hat{C}_2\|b\|_{\mathcal{Y}_1^*}$ (or an upper bound) and compute $\omega_j = e^{-1}(\delta + \omega_{j-1})$ for $j = 1, \dots, n-1$.

2: Set $k = \lceil 2e\hat{C}_1\hat{C}_2(L_A\sqrt{2g_1(1)} + L_B\sqrt{2g_2(1/\hat{C}_2)})/\tau \rceil$.

3: For $j = 0, \dots, n-1$ compute

$$\tau_1^{(j)} = \frac{\tau\hat{C}_1(\delta+\omega_j)}{L_A\sqrt{2g_1(1)} + L_B\sqrt{2g_2(1/\hat{C}_2)}}, \quad \tau_2^{(j)} = \frac{\tau\sqrt{2g_1(1)}}{L_A\hat{C}_1(\delta+\omega_j)}, \quad \tau_3^{(j)} = \frac{\tau\sqrt{2g_2(1/\hat{C}_2)}}{L_B\hat{C}_1(\delta+\omega_j)}.$$

4: Set $\phi_0(b) = 0$ (or any other initial approximation) and for $j = 1, \dots, n$, compute

$$\phi_j(b) = \texttt{InnerIt-SR}\left(\phi_{j-1}(b), \tau_1^{(j-1)}, \tau_2^{(j-1)}, \tau_3^{(j-1)}, k; b, 1/\hat{C}_2, \mathcal{J}, \mathcal{F}^*, A, B\right). \tag{2.23}$$

Output: $\phi_n(b) \in \mathcal{X}$.

following recovery bound holds:

$$\|\phi_n(b) - \mathbb{x}\|_{\mathcal{X}} \leq \sqrt{2D_{\mathcal{X}}(\mathbb{x}, \phi_n(b))} \leq \hat{C}_1\left(\frac{\delta}{1-e^{-1}} + e^{-n}\left[\mathcal{J}(0) + \mathcal{F}(0) + \hat{C}_2\|b\|_{\mathcal{Y}_1^*}\right]\right), \tag{2.24}$$

where $\phi_n(b)$ denotes the output of WARPd-SR in Algorithm 2.6.

Proof. See section SM2. ∎

2.5. Computational complexity and remarks. We analyze the computational cost of WARPd in the setting of Example 1.1, where Algorithm 2.2 is used. Let C_A, C_{A^*}, C_B, and C_{B^*} denote the computational cost of applying A, A^*, B, and B^*, respectively, and let $C_{\mathcal{J}}$ and $C_{\mathcal{F}^*}$ denote the cost of applying the proximal maps of $\mathcal{J}$ and $\mathcal{F}^*$, respectively. The cost per inner iteration is

$$C_A + C_{A^*} + C_B + C_{B^*} + C_{\mathcal{J}} + C_{\mathcal{F}^*} + \mathcal{O}(N + m + q).$$

If $\mathcal{F}(x) = \|x\|_{l^1}$, then

$$\text{prox}_{\tau_3\mathcal{F}^*}(y_3) = \varsigma_1(y_3), \quad \text{where} \quad [\varsigma_\rho(z)]_j = \min\{1, \rho/|z_j|\}\, z_j. \tag{2.25}$$

It follows that $C_{\mathcal{F}^*} = \mathcal{O}(q)$. For simple $\mathcal{J}$, such as those considered in section 3, $C_{\mathcal{J}} = \mathcal{O}(N)$. In compressed sensing applications, it is common for A to be a submatrix of a (rescaled)

unitary operator that admits a fast transform for matrix-vector products. Similarly, in l^1-analysis problems, B and B^* often admit fast transforms. In this case, the cost per iteration is bounded by a small multiple of N (and possibly logarithmic factors). Hence, each iteration is extremely fast. In the more general case, such as the nuclear norm in sections 4 and 5, where an SVD needs to be computed to apply $\text{prox}_{\tau_1 \mathcal{J}}$, $C_{\mathcal{J}}$ can be larger than $\mathcal{O}(N)$. However, the algorithm is still scalable to large problems and competitive with state-of-the-art methods (see subsections 4.2 and 5.4).

In the Hilbert space case, with $2D_{\mathcal{X}}(\mathbb{x}, \Psi_k) = \|\mathbb{x} - \Psi_k\|_{\mathcal{X}}$, Proposition 2.2 shows that (2.17) is achieved if each of the errors δ_j in (2.11) is bounded with $\delta_j \leq \mu$ for some

$$\mu \sim \frac{\delta \alpha \tau \sqrt{1-\tau^2}}{L_A \sqrt{2g_1(C_2)} + L_B \sqrt{2g_2(1)}} \sqrt{\frac{\min\{\tau_1(\omega), \tau_2(\omega), \tau_3(\omega)\}}{\max\{\tau_1(\omega), \tau_2(\omega), \tau_3(\omega)\}}}. \tag{2.26}$$

The values of ω in the proof of Theorem 1.2 are bounded above and below for given inputs. It follows that μ can be chosen as a constant that scales as $\mu \sim \delta^2$. This is useful in scenarios where the proximal map of $\mathcal{J}$ can only be applied approximately. Moreover, in certain cases, we may not know the matrices A or B exactly, or they have been stored to a finite precision. We can absorb this additional error into the error bounds for inexact computation in Proposition 2.2. Similarly, all of the algorithms in this paper can be executed on a Turing machine with almost identical error bounds. This is important for the computability of solutions of (1.2) to a given accuracy (e.g., see [15] and its numerical experiments). However, we have taken the usual convention throughout of proving results in exact arithmetic and providing stability bounds such as (2.12).

The following sections discuss how to select the constants C_1 and C_2 in different scenarios. For cases where $\|A\|$ and $\|B\|$ are unknown, we use the power method (applied to A^*A and B^*B) to find suitable L_A and L_B. This computation incurs a one-off upfront cost which is usually only as expensive as a few iterations of `InnerIt`. Practically, we found that Algorithm 2.3 performed better if the initial dual variables in `InnerIt` were selected as the final dual variables of the previous operation of `InnerIt` (as opposed to zero). Proposition 2.1 can be adapted accordingly by bounding the dual variables (the only change is to the final term on the right-hand side of (2.6)). We omit the details and instead discuss this point for WARPd-SR in section SM2, where the dual variables are bounded using the dual of the data fitting term $\|Ax - b\|_{\mathcal{Y}_1^*}$.

2.6. Unrolling Algorithm 2.3 as a stable and accurate NN. We now consider the setting of Example 1.1 with $\mathcal{F}(Bx) = \|Bx\|_{l^1}$ and prove Theorem 1.3. To capture standard architectures used in practice, such as skip connections, we consider the following definition of an NN. Without loss of generality and for ease of exposition, we also work with complex-valued NNs. Real-valued NNs can realize such networks by splitting into real and imaginary parts. An NN is a mapping $\phi\colon \mathbb{C}^m \to \mathbb{C}^N$ that can be written as a composition

$$\phi(y) = [V_T \circ \rho_{T-1} \circ V_{T-1} \circ \cdots \circ V_2 \circ \rho_1 \circ V_1](y), \quad \text{where}$$

- each V_j is an affine map $\mathbb{C}^{N_{j-1}} \to \mathbb{C}^{N_j}$ given by $V_j(x) = W_j x + b_j(y)$ where $W_j \in \mathbb{C}^{N_j \times N_{j-1}}$ and $b_j(y) = R_j y + c_j$ are affine functions of the input y;

- each $\rho_j \colon \mathbb{C}^{N_j} \to \mathbb{C}^{N_j}$ is one of two forms;

(i) There exists an index set $I_j \subset \{1, \dots, N_j\}$ such that ρ_j applies a nonlinear function $f_j : \mathbb{C} \to \mathbb{C}$ elementwise on the input vector's components with indices in I_j:

$$\rho_j(x)_k = \begin{cases} f_j(x_k) & \text{if } k \in I_j, \\ x_k & \text{otherwise.} \end{cases}$$

(ii) There exists a function $f_j : \mathbb{C} \to \mathbb{C}$ such that, after decomposing the input vector x as $(x_0, X^\top, Y^\top)^\top$ for scalar x_0 and $X \in \mathbb{C}^{m_j}, Y \in \mathbb{C}^{N_j - 1 - m_j}$, we have

$$\rho_j : \begin{pmatrix} x_0 \\ X \\ Y \end{pmatrix} \to \begin{pmatrix} 0 \\ f_j(x_0)X \\ Y \end{pmatrix}.$$

The affine dependence of $b_j(y)$ on y allows skip connections from the input to the current level, as in definitions of feed-forward NNs [114, p. 269], and the above architecture has become standard [48, 66, 71]. The use of nonlinear functions of the form (ii) may be reexpressed using nonlinear functions of the form (i) and the following standard elementwise squaring trick:

$$f_j(x_0)X = \frac{1}{2}\left[[f_j(x_0)\mathbf{1} + X]^2 - f_j(x_0)^2\mathbf{1} - X^2\right],$$

where $\mathbf{1}$ denotes a vector of ones of the same size as X. The key observation is that the basic operations of Algorithm 2.2 can be unrolled as NNs. For example, γ_ρ can be executed via

$$x \xrightarrow{\text{L}} \begin{pmatrix} x \\ x \end{pmatrix} \xrightarrow{\text{NL}} \begin{pmatrix} |x_1|^2 \\ \vdots \\ |x_m|^2 \\ x \end{pmatrix} \xrightarrow{\text{L}} \begin{pmatrix} \sum_{j=1}^M |x_j|^2 \\ x \end{pmatrix} \xrightarrow{\text{NL}} \begin{pmatrix} 0 \\ \max\left\{0, 1 - \frac{\rho}{\|x\|_{l^2}}\right\} x \end{pmatrix} \xrightarrow{\text{L}} \max\left\{0, 1 - \frac{\rho}{\|x\|_{l^2}}\right\} x,$$

where "L" denotes affine maps and "NL" nonlinear maps. The second arrow applies pointwise modulus squaring (type (i) above), and the penultimate arrow applies a nonlinear map (type (ii) above). Similarly, ς_ρ defined in (2.25) can be unrolled as an NN of fixed depth and width of order $\mathcal{O}(N + m + q)$.

Proof of Theorem 1.3. Under the assumptions, we see that each iteration in Algorithm 2.2 (now with the appropriate change of parameters to encompass inexact primal-dual iterates as in the proof of Theorem 1.2) can be executed by an NN of width $\mathcal{O}(N + m + q)$ and depth $\mathcal{O}(M)$. This follows via the unrolling of γ_ρ and ς_ρ, the approximation of the proximal map of $\mathcal{J}$, and concatenation of NNs. Similarly, the operations and restarting in Algorithm 2.3 can be combined into an NN. The result now follows from Theorem 1.2 and the bound (1.8) on the number of inner iterations required to achieve (1.9). ∎

3. Sparse recovery. We consider the setup of Example 1.1 with $D_{\mathcal{Z}}(z, \hat{z}) = \frac{1}{2}\|z - \hat{z}\|_{l^2}^2$, and sparse recovery via the (weighted) l^1-norm

$$\mathcal{J}(x) = \|x\|_{l^1_w} := \sum_{j=1}^N w_j |x_j|, \quad w_j \geq 0 \quad \text{(and take } \mathcal{F} = 0\text{)}, \tag{3.1}$$

for which (1.2) becomes the famous basis pursuit denoising problem. This is a ubiquitous problem in many fields, including machine learning, compressed sensing, and image processing [30, 33, 34, 50]. The assumption (1.5) holds for matrices A that have a (weighted) robust null space property (in levels) defined in Definition 3.2, allowing the recovery of vectors $\mathtt{x}$ that are approximately sparse (in levels).[6] Our result is presented explicitly in Theorem 3.3. This setting is very general, for example, encompassing both classical and structured compressed sensing. Examples in imaging for Fourier and Walsh measurements are given in subsection 3.2.

3.1. A general result. We consider sparsity in levels [7], which has been shown to play a key role in the quality of image recovery in compressed sensing via the so-called flip test [7, 14]. For many imaging modalities, sparsity in levels is crucial in demonstrating that sparse regularization is near-optimal for image recovery [3, 14, 72]. It is needed to account for the good recovery often found in practice for problems such as the Fourier-wavelet problem.[7] For example, [87] observed both poor recovery from uniform random sampling and the improvement offered by variable density sampling for magnetic resonance imaging (MRI). For further works on structured compressed sensing, see [23, 24, 56, 79, 81, 83, 119]. The following definitions also encompass classical compressed sensing.

Definition 3.1 (sparsity in levels). *Let $\mathbf{M} = (M_1, \dots, M_r) \in \mathbb{N}^r$, $1 \leq M_1 < \cdots < M_r = N$, and $\mathbf{s} = (s_1, \dots, s_r) \in \mathbb{N}^r$, where $s_k \leq M_k - M_{k-1}$ for $k = 1, \dots, r$ ($M_0 = 0$). A vector $x \in \mathbb{C}^N$ is $(\mathbf{s}, \mathbf{M})$-sparse in levels if*

$$|\mathrm{supp}(x) \cap \{M_{k-1} + 1, \dots, M_k\}| \leq s_k, \quad k = 1, \dots, r.$$

The total sparsity is $s = s_1 + \cdots + s_r$. We denote the set of $(\mathbf{s}, \mathbf{M})$-sparse vectors by $\Sigma_{\mathbf{s},\mathbf{M}}$. We also define the measure of distance of a vector x to $\Sigma_{\mathbf{s},\mathbf{M}}$ by

$$\sigma_{\mathbf{s},\mathbf{M}}(x)_{l^1_w} = \inf \left\{ \|x - z\|_{l^1_w} : z \in \Sigma_{\mathbf{s},\mathbf{M}} \right\}.$$

In section 6, we will drop the $\mathbf{M}$ subscript when considering a single level. For simplicity, we assume that $w_i = w_{(j)} > 0$ if $M_{j-1} + 1 \leq i \leq M_j$. If an image c is compressible in a wavelet basis with coefficients x, then $\sigma_{\mathbf{s},\mathbf{M}}(x)_{l^1_w}$ is expected to be small whenever the levels correspond to wavelet levels [91, Chapter 9]. In general, the weights are a prior on the anticipated approximate support of the vector [61]. We also define the following quantities:

$$\xi = \xi(\mathbf{s}, \mathbf{M}, w) := \sum_{k=1}^{r} w_{(k)}^2 s_k, \quad \zeta = \zeta(\mathbf{s}, \mathbf{M}, w) := \min_{k=1,\dots,r} w_{(k)}^2 s_k, \quad \kappa = \kappa(\mathbf{s}, \mathbf{M}, w) := \xi/\zeta.$$

Definition 3.2 (weighted rNSP in levels [14]). *Let $(\mathbf{s}, \mathbf{M})$ be local sparsities and sparsity levels, respectively. For weights $\{w_i\}_{i=1}^N$ ($w_i > 0$), we say that $A \in \mathbb{C}^{m \times N}$ satisfies the weighted robust null space property in levels (weighted rNSPL) of order $(\mathbf{s}, \mathbf{M})$ with constants $0 < \rho < 1$ and $\gamma > 0$ if for any $(\mathbf{s}, \mathbf{M})$ support set Δ,*

$$\|x_\Delta\|_{l^2} \leq \rho \|x_{\Delta^c}\|_{l^1_w} / \sqrt{\xi} + \gamma \|Ax\|_{l^2} \qquad \forall\, x \in \mathbb{C}^N.$$

Here, x_S denotes the vector with $[x_S]_j = x_j$ if $j \in S$ and $[x_S]_j = 0$ otherwise.

[6] This is a weaker assumption than the restricted isometry property [60, Theorem 6.13].

[7] In this example, the main problem for sparsity in one level is that the Fourier-wavelet matrix is coherent [7].

With these definitions in hand, the following provides the reconstruction guarantee.

Theorem 3.3. *Suppose that A has the weighted rNSPL of order $(\mathbf{s}, \mathbf{M})$ with constants $0 < \rho < 1$ and $\gamma > 0$. Then the approximate sharpness condition (1.5) holds for any $\mathbb{x} \in \mathbb{C}^N$ with*

$$C_1 = \left(\rho + \frac{(1+\rho)\kappa^{1/4}}{2}\right)\frac{1+\rho}{\sqrt{\xi}(1-\rho)}, \quad C_2 = \frac{\gamma}{C_1}\cdot\frac{2+2\rho+(3+\rho)\kappa^{1/4}}{2(1-\rho)}, \quad \text{and}$$
$$c(\mathbb{x}, b) = 2\sigma_{\mathbf{s},\mathbf{M}}(\mathbb{x})_{l_w^1} + C_2\left(\|A\mathbb{x} - b\|_{l^2} + \epsilon\right).$$

Let $\epsilon > 0$, L_A be an upper bound for $\|A\|$, $\tau \in (0,1)$, $\delta > 0$. Then for any $n \in \mathbb{N}$ and any pair $(\mathbb{x}, b) \in \mathbb{C}^N \times \mathbb{C}^m$ such that $\|A\mathbb{x} - b\| \le \epsilon$ and $c(\mathbb{x}, b) \le \delta$,

$$\|\phi_n(b) - \mathbb{x}\|_{l^2} \le C_1\left[\frac{\delta}{1-\exp(-1)} + C_2\|b\|_{l^2}\cdot\exp\left(-T(n)\left\lfloor 2eL_A\gamma\frac{2+2\rho+(3+\rho)\kappa^{1/4}}{2\tau(1-\rho)}\right\rfloor^{-1}\right)\right],$$
$$\|\phi_n(b) - \mathbb{x}\|_{l_w^1} \le \frac{1+\rho}{1-\rho}\left[\frac{\delta}{1-\exp(-1)} + C_2\|b\|_{l^2}\cdot\exp\left(-T(n)\left\lfloor 2eL_A\gamma\frac{2+2\rho+(3+\rho)\kappa^{1/4}}{2\tau(1-\rho)}\right\rfloor^{-1}\right)\right],$$

where $\phi_n(b)$ denotes the output of WARPd (Algorithm 2.3) and $T(n) = nk$ denotes the total number of inner iterations.

Proof. See section SM3. ∎

In summary, if A satisfies the robust null space property (in levels), then WARPd provides accelerated recovery. The condition $c(\mathbb{x}, b) \le \delta$ means that both the measurement error $\|A\mathbb{x} - b\|_{l^2} + \epsilon$ and the distance of $\mathbb{x}$ to $\Sigma_{\mathbf{s},\mathbf{M}}$ (measured by $\sigma_{\mathbf{s},\mathbf{M}}(\mathbb{x})_{l_w^1}$) are small. Moreover, the rate of convergence is directly related to ρ, γ, and κ.

3.2. Example in compressive imaging. We consider the case that A is a multilevel subsampled unitary matrix [7] with respect to $U = V\Psi^*$, where Ψ denotes the db2 wavelet transform and V is the discrete Fourier (Fourier sampling) or Walsh–Hadamard transform (binary sampling). A and A^* are implemented rapidly using the fast Fourier transform or fast Walsh–Hadamard transform, and the discrete wavelet transform. Note that $[\text{prox}_{\tau_1\mathcal{J}}(x)]_i = \max\{0, 1 - \tau_1 w_i/|x_i|\}x_i$. Hence, the cost per inner iteration is $\mathcal{O}(N\log(N))$. Fourier sampling arises in numerous applications such as MRI, nuclear magnetic resonance, and radio interferometry, while binary sampling arises in optical imaging modalities such as lensless imaging, infrared imaging holography, and fluorescence microscopy. Further details on the bases used, sampling structure, and results that A has the weighted rNSPL are given in section SM3. Figure 1 (left) shows the test image used in this section.

As a benchmark, we compare to the algorithm NESTA [17] (available at https://statweb.stanford.edu/~candes/software/nesta/), which applies a smoothing technique and an accelerated first-order algorithm [99]. NESTA is widely regarded as a state-of-the-art method for basis pursuit, is widely used for solving large-scale compressed sensing reconstruction problems, and compares favorably with other state-of-the-art methods (see, for example, the extensive numerical tests in [17, section 5]). We run two versions of NESTA to solve (1.2), both with default parameters and acceleration through continuation. For the first version,

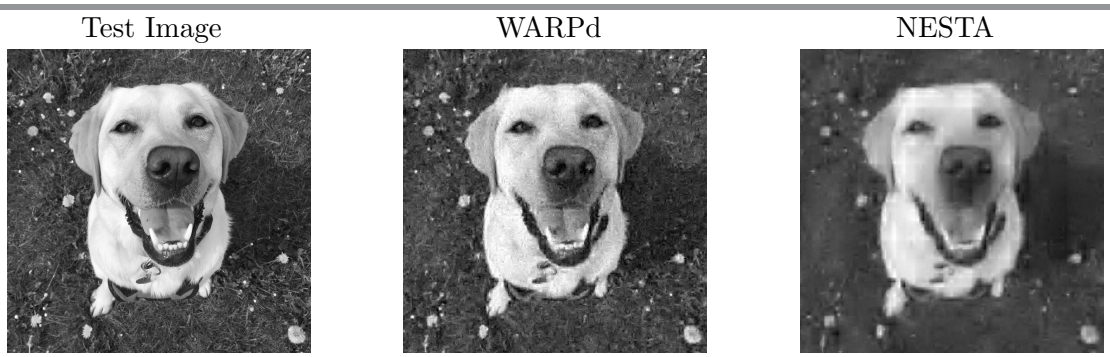

Figure 1. *Left:* 1024×1024 *test image with pixel values scaled to* $[0, 1]$. *Middle: Recovered image from* 5% *binary measurements using WARPd and* 30 *matrix-vector products. Right: Recovered image from* 5% *binary measurements using NESTA and* 150 *matrix-vector products.*

we take a smoothing parameter $\mu = 0.001$. For the second version, we perform a grid-based search for optimal smoothing parameters, and for each number of iterations, we report the error for an optimal smoothing parameter. As an error metric for an iterate x, we take

$$\text{Error}(x) = \left(\left| \|x\|_{l_w^1} - \|x^*\|_{l_w^1} \right| + C_2 \left| \|Ax - b\|_{l^2} - \epsilon \right| \right) / \|x^*\|_{l_w^1}, \tag{3.2}$$

where x^* (approximately) minimizes (1.2) and is computed using several hundred thousand iterations to be sure of convergence. This error directly measures the objective function optimality gap and the feasibility gap (also note that $\text{Error}(x^*) = 0$). It also controls the recovery of the sought-for image $\mathbb{x}$ (see the proof of Theorem 3.3). In what follows, we present this error metric as a function of the number of matrix-vector products (A or A^*) used.

We first consider 15% subsampling and corrupt the measurements with 5% Gaussian noise. The constants C_1 and C_2 are taken from the discussion in section SM3. The sparsities and weights are estimated by thresholding the wavelet coefficients of a Shepp–Logan phantom. In particular, we do not choose or tune any parameters based on the image we use to test the algorithm. We take $\epsilon = 0.06\|b\|_{l^2}$, $\delta = C_2\epsilon$, and $\tau = 1$. Figure 2 (left and middle) shows the convergence for our algorithm using ergodic iterates and nonergodic iterates in the inner iterations. We have also shown results for nonrestarted primal-dual iterations (labeled PD). The benefit of acceleration is clear, and our algorithm converges at a much faster rate than NESTA. The nonergodic version of our algorithm performs better than Algorithm 2.3. We do not have a theoretical explanation for this, but this kind of behavior (and its reverse, i.e., ergodic iterates performing better) has been observed for nonrestarted primal-dual iterations [41]. The case of binary sampling also converges slightly faster (this is to be expected from the sampling bounds mentioned in section SM3). We found similar behavior for different images, subsampling rates, higher-order wavelets, etc.

We now consider the difference between the reconstruction and the image itself. From Theorem 3.3, we expect that this error will decrease linearly down to the intrinsic bound $\sim C_1\delta$, which corresponds to the distance from the image to the set of solutions of (1.2). Figure 2 (right) shows the relative mean square error (MSE) between the reconstruction and the image for Fourier sampling at different sampling rates. In all cases, the level of noise was chosen so that it contributes an error comparable to solving (1.2). We see the expected behavior, where

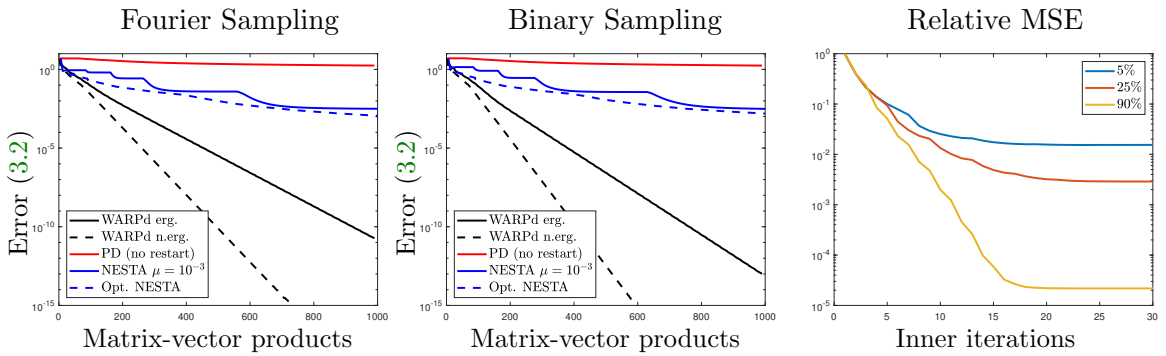

Figure 2. *Left: Convergence for Fourier* 15% *sampling. Middle: Convergence for binary* 15% *sampling. Right: Convergence for Fourier sampling and different sampling rates (relative MSE).*

the final error is due to the fact that the image's wavelet coefficients are only approximately sparse (for example, the error for a standard phantom image was of the order 10^{-12}), and, as expected, is smaller for the larger sampling rate, with a faster rate of convergence. Similar behavior occurs for binary sampling. For example, Figure 1 (middle, right) shows the reconstruction using 5% sampling and 30 matrix-vector products for WARPd, as well as 150 matrix-vector products for NESTA. Again, this demonstrates the faster convergence of WARPd.

4. Low-rank matrix recovery. We consider the setup of Example 1.1 with $D_{\mathcal{Z}}(z, \hat{z}) = \frac{1}{2}\|z - \hat{z}\|_{l^2}^2$, where $\mathcal{X} = \mathbb{C}^{n_1 \times n_2}$ (so that the vectorized l^2-norm is the Frobenius norm). We consider the problem of recovering an approximately low-rank matrix $\mathbb{x} \in \mathbb{C}^{n_1 \times n_2}$ via (1.2) with the nuclear norm regularizer

$$\mathcal{J}(M) = \|M\|_1 := \sum_{j=1}^{\min\{n_1,n_2\}} \sigma_j(M) \quad \text{(and take } \mathcal{F} = 0\text{)}, \tag{4.1}$$

where $\sigma_j(M)$ denotes the singular values of $M \in \mathbb{C}^{n_1 \times n_2}$. Low-rank matrix recovery is a non-commutative version of recovery of (approximately) sparse vectors. The low-rank assumption means that the matrix $\mathbb{x}^*\mathbb{x}$ is sparse in its eigenbasis. There are numerous instances where nuclear norm minimization and related problems provably recover the desired low-rank matrix from considerably fewer than $n_1 n_2$ measurements [32, 63, 77, 80, 86, 105].[8]

We consider measurement maps of the form (tr denotes trace)

$$A(M) = \sum_{j=1}^{m} \operatorname{tr}(MA_j^*)e_j \in \mathbb{C}^m, \quad A^*(y) = \sum_{j=1}^{m} y_j A_j \in \mathbb{C}^{n_1 \times n_2}, \tag{4.2}$$

where $A_j \in \mathbb{C}^{n_1 \times n_2}$ are measurement matrices and the $\{e_j\}_{j=1}^m$ are the canonical basis vectors of $\mathbb{C}^m$. The assumption (1.5) holds for measurement maps A that satisfy the Frobenius-robust rank null space property in Definition 4.1, which is analogous to Definition 3.2. This is a weaker assumption than the rank restricted isometry property [60, Theorem 6.13][9] (another

[8] Similar to the relationship between l^1 and l^0 minimization, the nuclear norm is a convex relaxation of the rank operator and the rank minimization problem is NP-hard in general.

[9] The cited theorem is for the analogous properties of sparse recovery of vectors. The adaptation of the proof for the case of recovery of low-rank matrices is straightforward using the relevant Schatten p-norms.

common property used to prove recovery results [31, 105]) and allows the recovery of matrices M that are approximately low-rank. Theorem 4.2 gives our result and, as an example, we consider Pauli measurements in quantum state tomography.

4.1. A general result. The following definition is analogous to Definition 3.2 for a single level and unweighted l^1-norm,[10] but now the relevant norms are replaced by their Schatten p-norm counterparts. We use $\|M_c\|_1$ to denote $\sum_{j>r}\sigma_j(M)$ for a given r.

Definition 4.1 (Frobenius-robust rank null space property [73]). *We say that $A:\mathbb{C}^{n_1\times n_2}\to\mathbb{C}^m$ satisfies the Frobenius-robust rank null space property of order r with constants $\rho\in(0,1)$ and $\gamma>0$ if for all $M\in\mathbb{C}^{n_1\times n_2}$, the singular values of M satisfy*

$$\sqrt{\sigma_1(M)^2+\cdots+\sigma_r(M)^2}\le\rho\|M_c\|_1/\sqrt{r}+\gamma\|A(M)\|_{l^2}.$$

The following provides the reconstruction guarantee.

Theorem 4.2. *Suppose that $A:\mathbb{C}^{n_1\times n_2}\to\mathbb{C}^m$ satisfies the Frobenius-robust rank null space property of order r with constants $\rho\in(0,1)$ and $\gamma>0$. Then the approximate sharpness condition (1.5) holds for any $\mathbb{x}\in\mathbb{C}^{n_1\times n_2}$ with*

$$C_1=\frac{(1+\rho)^2}{(1-\rho)r^{\frac12}},\quad C_2=\frac{\gamma(3+\rho)r^{\frac12}}{(1+\rho)^2},\quad c(\mathbb{x},b)=2\|\mathbb{x}_c\|_1+\frac{\gamma(3+\rho)r^{\frac12}}{(1+\rho)^2}\left(\|A(\mathbb{x})-b\|_{l^2}+\epsilon\right). \tag{4.3}$$

Let $\epsilon>0$, L_A be an upper bound for $\|A\|$, $\tau\in(0,1)$, $\delta>0$, and let C_1,C_2 and $c(\cdot,\cdot)$ be given by (4.3). Then for any $n\in\mathbb{N}$, and $p\in[1,2]$, and any pair $(\mathbb{x},b)\in\mathbb{C}^{n_1\times n_2}\times\mathbb{C}^m$ such that $\|A(\mathbb{x})-b\|\le\epsilon$, and $c(\mathbb{x},b)\le\delta$, the following uniform recovery bounds hold:

$$\|\phi_n(b)-\mathbb{x}\|_p\le\frac{(1+\rho)^2}{(1-\rho)}\left[\frac{\delta r^{\frac{1-p}{p}}}{1-\exp(-1)}+\frac{\gamma(3+\rho)r^{\frac1p-\frac12}}{(1+\rho)^2}\|b\|_{l^2}\cdot\exp\left(-T(n)\left[\frac{2eL_A\gamma}{\tau}\frac{(3+\rho)}{(1-\rho)}\right]^{-1}\right)\right],$$

where $\phi_n(b)$ denotes the output of WARPd (Algorithm 2.3) and $T(n)=nk$ denotes the total number of inner iterations.

Proof. See section SM4. ∎

In summary, if A satisfies the Frobenius-robust rank null space property, then WARPd provides accelerated recovery. The condition $c(\mathbb{x},b)\le\delta$ means that both the measurement error $\|A(\mathbb{x})-b\|_{l^2}+\epsilon$ and the distance of $\mathbb{x}$ to low-rank matrices (i.e., $\|\mathbb{x}_c\|_1$) are small. Moreover, the convergence rate is directly related to ρ and γ.

4.2. Example: Pauli measurements and quantum state tomography. An important application of matrix recovery in physics, known as quantum state tomography (QST), is reconstructing a finite n-dimensional quantum mechanical system. Such a system is fully characterized by its density operator ρ, an $n\times n$ positive semidefinite matrix with trace 1. For example, QST is now a routine task for designing, testing, and tuning qubits in the quest of building quantum information processing devices [88]. A key structural property, for which

[10] It is possible to consider a weighted version of the nuclear norm. However, the associated optimization problem is very difficult and nonconvex [65].

the quantum system is called "almost pure," is that ρ be well approximated by a low-rank matrix. Under this assumption, QST becomes a low-rank matrix recovery problem [63, 64, 86]. QST requires a measurement process that is experimentally realizable and efficient.

In this example, we consider Pauli measurements, where the A_j are constructed from randomly sampling tensor products of the usual Pauli matrices. Pauli measurements lead to efficient recovery of low-rank density operators [63, 64] and are especially easy to carry out experimentally [107, 112]. It was shown in [86] that sets of $\mathcal{O}(rn \cdot \text{poly}(\log(n)))$ Pauli measurements satisfy the rank restricted isometry property, and hence satisfy the Frobenius-robust rank null space property in Definition 4.1. We can thus apply Theorem 4.2.

In the general case of (4.1), the proximal map of $\mathcal{J}$ is computed using the SVD. Namely, if $M = U\text{diag}(\sigma(M))V^* \in \mathbb{C}^{n_1 \times n_2}$, then [27, Theorem 2.1]

$$\text{prox}_{\tau_1 \mathcal{J}}(M) = U\phi_{\tau_1}(\text{diag}(\sigma(M)))V^*, \quad \phi_\alpha(z) = \max\{0, 1 - \alpha/|z|\}\, z, \tag{4.4}$$

where ϕ_{τ_1} is applied elementwise to the diagonal matrix $\text{diag}(\sigma(M))$. Naively, the cost of applying $\text{prox}_{\tau_1 \mathcal{J}}$ is dominated by the $\mathcal{O}(n_1 n_2 \min\{n_1, n_2\})$ cost of computing the SVD [120, Chapter 31]. In this example, since the measurement matrix is sparse and, due to the thresholding, we only need the dominant eigenvalues (the matrices are Hermitian so the SVD reduces to an eigenvalue decomposition), we found it beneficial to use methods for computing eigenvalue decompositions based on matrix-vector products (see subsection 5.2). In general, reducing the number of iterations through accelerated methods such as WARPd is particularly important in low-rank matrix recovery since the cost of applying A may be large for large n_1 and n_2 (e.g., for Gaussian measurements used in phase retrieval [115]).[11]

As a benchmark, we compare to TFOCS [18] (available at http://cvxr.com/tfocs/), which has become a de facto method for matrix retrieval problems such as PhaseLift [29, 35, 58] and other related techniques. TFOCS applies an optimal first-order method [13] to a smoothed version of the dual problem. We use the default parameters (apart from the tolerance, which we decrease to achieve higher accuracy), accelerated continuation, and a smoothing parameter $\mu = 1$ (relative to $\|A\|$). In this case, the smoothing term is $\frac{\mu}{2}\|\cdot - M_0\|_2^2$, with M_0 updated at each restart. As an error metric for an iterate $\tilde{M}$, we take the relative error

$$\text{Error}(M) = \left(\left|\|M\|_1 - \|M^*\|_1\right| + C_2\left|\|A(M) - b\| - \epsilon\right|\right)/\|M^*\|_1, \tag{4.5}$$

where M^* (approximately) minimizes (1.2), and is computed using a much larger number of iterations.

For our example, we set $r = 10$ and $n = 2^{10}$ (corresponding to 10 qubits). We generate two independent complex standard Gaussian matrices $M_L, M_R \in \mathbb{C}^{n \times r}$ and set $M = M_L M_R^* M_R M_L^*$, $\mathbb{x} = M/\text{tr}(M)$. We then use 10% subsampling and corrupt the measurements with 2% Gaussian noise. We take $\epsilon = 0.03\|b\|_{l^2}$, $\delta = C_2\epsilon$ (C_1 and C_2 are selected based on the theorem in [86]), and $\tau = 1$. Figure 3 shows the results. We see the clear benefit of acceleration and that WARPd converges at a much faster rate than TFOCS.

[11] For Gaussian measurements and general measurement matrices A_j in (4.2), $C_A = \mathcal{O}(n_1 n_2 m)$ with $m \gtrsim n_1 + n_2$ so there is little benefit gained by using an approximate SVD.

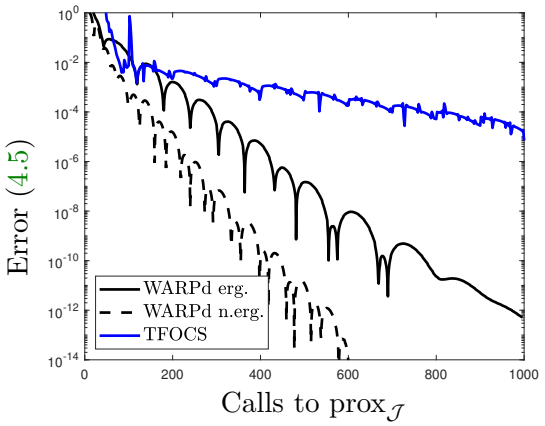

Figure 3. *Errors for Pauli measurements example.*

5. Matrix completion and nonuniform recovery guarantees. In this section, we consider the problem of matrix completion. Given an approximately low-rank matrix $\mathbb{x} \in \mathbb{C}^{n_1 \times n_2}$ and an index set $\Omega \subset \{1, \dots, n_1\} \times \{1, \dots, n_2\}$ with $|\Omega| = m < n_1 n_2$, we recover $\mathbb{x}$ from measurements b where

$$[P_\Omega(M)]_{i,j} = \begin{cases} M_{i,j} & \text{if } (i,j) \in \Omega, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad b - e = A(\mathbb{x}) = \operatorname{vect}(P_\Omega(\mathbb{x})) \in \mathbb{C}^{|\Omega|}.$$

This can be viewed as a special case of the problem considered in section 4 and we consider

$$\min_{M \in \mathbb{C}^{n_1 \times n_2}} \|M\|_1 \quad \text{such that} \quad \|\operatorname{vect}(P_\Omega(M)) - b\|_{l^2} \leq \epsilon. \tag{5.1}$$

However, we treat the problem separately for at least three reasons. First, there are obvious rank-one matrices in the kernel of the measurement operator, and hence the Frobenius-robust rank null space property we made use of in subsection 4.1 cannot hold. The lack of such a global property renders matrix completion a more challenging problem. However, if certain conditions on the left and right singular vectors of the underlying low-rank matrix are imposed, essentially requiring that such vectors are uncorrelated with the canonical basis, then the matrix can be recovered with sufficiently many measurements [32, 36, 63, 104]. Such conditions lead to *nonuniform recovery guarantees*. We show how such results fall within our framework of (1.5). Similar arguments also hold for the nonuniform recovery of sparse vectors. Second, this problem has distinct algorithmic challenges when dealing with large-scale problems, discussed in subsection 5.2. Third, (approximately) low-rank matrices pervade data science [121] and matrix completion has received much attention with applications ranging from recommender systems [76, 106], inferring camera motion [44, 118], multiclass learning [10, 57], and many more in statistics, machine learning, and signal processing.

5.1. A general result. We show that (1.5) holds for the problem of matrix completion under the existence of an approximate dual certificate. The existence of an approximate dual certificate is a predominant method of proving that solutions of optimization problems such

as (5.1) approximate $\mathbb{x}$. We take $D_{\mathcal{Z}}(z,\hat{z}) = \|z-\hat{z}\|_{\mathcal{Z}}^2/2$, where $\mathcal{Z}$ is the appropriate Hilbert space (the l^2-norm for vectors and Frobenius norm for matrices).

Let $\mathbb{x} \in \mathbb{C}^{n_1\times n_2}$ and let $\mathbb{x} = U\Sigma V^*$ denote its SVD. Here $\Sigma \in \mathbb{R}^{r\times r}$ is diagonal (with $r \leq \min\{n_1,n_2\}$), and $U \in \mathbb{C}^{n_1\times r}, V \in \mathbb{C}^{n_2\times r}$ are partial isometries (so that $U^*U = V^*V = I_r$). To state the conditions for accelerated recovery, we first introduce a few objects associated with $\mathbb{x}$. The tangent space of the variety of rank r matrices at the point $\mathbb{x}$ is given by

$$T_{\mathbb{x}} = \{UB_1^* + B_2V^* : B_1 \in \mathbb{C}^{n_2\times r}, B_2 \in \mathbb{C}^{n_1\times r}\}.$$

We denote by $P_{T_{\mathbb{x}}}$, the (Hilbert–Schmidt) orthogonal projection onto the tangent space and set $P_{T_{\mathbb{x}}^\perp} = I - P_{T_{\mathbb{x}}}$. Let $P = UU^*$ and $Q = VV^*$. One can easily check that

$$P_{T_{\mathbb{x}}^\perp} : M \to P^\perp M Q^\perp, \quad P_{T_{\mathbb{x}}} : M \to PM + MQ - PMQ.$$

With these in hand, we provide the following two definitions. These are slightly weaker than those usually used in the literature (for example, it is common to assume a restricted isometry property instead of Definition 5.2), but they suffice to prove Theorem 5.3.

Definition 5.1. *Given a measurement operator $A : \mathbb{C}^{n_1\times n_2} \to \mathbb{C}^m$, a vector $z \in \mathbb{C}^m$, with matrix $Y = A^*(z) \in \mathbb{C}^{n_1\times n_2}$, is an approximate dual certificate at $\mathbb{x}$ if upon defining*

$$\alpha_1 = \|UV^* - P_{T_{\mathbb{x}}}Y\|_2 \text{ and } \alpha_2 = \|P_{T_{\mathbb{x}}^\perp}Y\|, \quad \text{it holds that} \quad \alpha_2 < 1. \tag{5.2}$$

Definition 5.2. *We say that A is bounded below on $T_{\mathbb{x}}$ with constant $\gamma > 0$ if*

$$\gamma\|Z\|_2 \leq \|A(Z)\|_{l^2} \quad \forall Z \in T_{\mathbb{x}}. \tag{5.3}$$

Theorem 5.3. *Let $\mathbb{x} \in \mathbb{C}^{n_1\times n_2}$ and suppose that A is bounded below on $T_{\mathbb{x}}$ with constant $\gamma > 0$ and $z \in \mathbb{C}^m$ is an approximate dual certificate at $\mathbb{x}$ (so that (5.2) holds). If $\alpha_1\|A\| < (1-\alpha_2)\gamma$, then for any $\widehat{M} \in \mathbb{C}^{n_1\times n_2}$,*

$$\|\widehat{M} - \mathbb{x}\|_2 \leq \frac{\gamma + \|A\|}{(1-\alpha_2)\gamma - \alpha_1\|A\|}\left[\|\widehat{M}\|_1 - \|\mathbb{x}\|_1 + \left(\frac{\alpha_1 + 1 - \alpha_2}{\gamma + \|A\|} + \|z\|_{l^2}\right)\left\|A\left(\widehat{M} - \mathbb{x}\right)\right\|_{l^2}\right]. \tag{5.4}$$

It follows that (1.5) is satisfied for the problem (4.1) with

$$C_1 = \frac{\gamma + \|A\|}{(1-\alpha_2)\gamma - \alpha_1\|A\|}, C_2 = \left(\frac{\alpha_1 + 1 - \alpha_2}{\gamma + \|A\|} + \|z\|_{l^2}\right), c(\mathbb{x}, b) = C_2(\epsilon + \|A(\mathbb{x}) - b\|_{l^2}).$$

Proof. See section SM4. ∎

The existence of (approximate) dual certificates for matrix completion has been studied extensively [32, 36, 63, 104]. We follow [49], which gives the current state-of-the-art sample complexity. The observation indices Ω are chosen randomly such that $\mathbb{P}((i,j) \in \Omega) = p \in [0,1)$ for all (i,j) independently. Using the standard basis $\{e_je_k^*\}_{j=1,k=1}^{n_1,n_2}$, the coherence of $\mathbb{x}$ is

$$\mu(\mathbb{x}) = \max\left\{\frac{n_1}{r}\max_{i\in\{1,\dots,n_1\}}\|U^*e_i\|_{l^2}^2, \frac{n_2}{r}\max_{i\in\{1,\dots,n_2\}}\|V^*e_i\|_{l^2}^2\right\} \in \left[1, \frac{\max\{n_1,n_2\}}{r}\right].$$

It was shown in [49][12] that if

$$p \gtrsim \mu(\mathbb{x}) r \log(\mu(\mathbb{x}) r) \log(\max\{n_1, n_2\}) / \min\{n_1, n_2\},$$

then with high probability,[13] there is an approximate dual certificate at $\mathbb{x}$ with $\alpha_1 \leq p/4$ and $\alpha_2 \leq 1/2$, and $\|P_{T_{\mathbb{x}}} p^{-1} P_\Omega P_{T_{\mathbb{x}}} - P_{T_{\mathbb{x}}}\| \leq 1/2$. Let $Z \in T_{\mathbb{x}}$; then

$$\|P_\Omega Z\|_2^2 = p\langle Z, P_{T_{\mathbb{x}}} p^{-1} P_\Omega P_{T_{\mathbb{x}}} Z\rangle \geq p\|Z\|_2^2 (1 - \|P_{T_{\mathbb{x}}} p^{-1} P_\Omega P_{T_{\mathbb{x}}} - P_{T_{\mathbb{x}}}\|) \geq \frac{p}{2}\|Z\|_2^2. \tag{5.5}$$

Hence, we take $\gamma = \sqrt{p/2}$ in (5.3) with $A = P_\Omega$ (treating outputs as vectors $P_\Omega(M) \in \mathbb{C}^{|\Omega|}$).

Corollary 5.4. *If $p \gtrsim \mu(\mathbb{x}) r \log(\mu(\mathbb{x}) r) \frac{\log(\max\{n_1,n_2\})}{\min\{n_1,n_2\}}$, then with high probability the conditions of Theorem 5.3 hold with $C_1 \lesssim p^{-1/2}$ and C_2 bounded independently of all parameters. It follows that the conclusions of Theorems 1.2 and 1.3 hold.*

Ignoring logarithmic factors, the above has a dimension scaling $C_1 C_2 \sim \sqrt{\min\{n_1, n_2\}}$. In general, it is impossible to eliminate this dimensional scaling [77, Theorem 3.5].

5.2. Algorithmic considerations. For matrix completion, the main computational burden of our algorithm is the step

$$M^{(j+1)} = \mathrm{prox}_{\tau_1 \|\cdot\|_1}\big(\underbrace{M^{(j)}}_{\text{low-rank}} - \underbrace{\tau_1 A^* z_1^{(j)}}_{\text{sparse}} \big),$$

which requires the application of the singular value thresholding (SVT) operator in (4.4). To reduce memory consumption, we store the iterates in low-rank factored SVD form $M^{(j)} = U^{(j)} \Sigma^{(j)} V^{(j)}$. The chosen rank of this factored form will be close to the approximate rank of $\mathbb{x}$ when using our update rule below. The matrix $A^* z_1^{(j)}$ is sparse and its nonzero entries correspond to the indices in Ω. It follows that $M^{(j)} - \tau_1 A^* z_1^{(j)}$ is a sum of a low-rank factorized matrix and a sparse matrix. Hence both it and its adjoint can be applied rapidly to vectors. We, therefore, make use of the PROPACK package [82], which uses iterative methods based on Lanczos bidiagonalization with partial reorthogonalization for computing the first r' singular vectors/values.[14] PROPACK only uses matrix-vector products and has been found to be an efficient and stable package for computing the dominant singular values and singular vectors of large matrices. To use PROPACK in this scenario, we must supply a prediction of the dimension of the principal singular space whose singular values are larger than the given threshold. We provide an initial starting guess r' (5 in our experiments), and at each iteration, we increase r' by one for the following iteration if the dimension of the principal singular space is too small, or decrease by one if it is too large.

[12] [49] considers real matrices but the result can be easily extended to complex matrices.

[13] Meaning with probability at least $1 - c_1(n_1 + n_2)^{-c_2}$ for constants $c_1, c_2 > 0$.

[14] There are very efficient direct matrix factorization methods for calculating the SVD of matrices of moderate size (at most a few thousand). When the matrix is sparse, larger problems can be solved; however, the computational cost depends heavily upon the sparsity structure of the matrix. In general, for large matrices one has to resort to indirect iterative methods for calculating the leading singular vectors/values.

Following the arguments in subsection 5.1, we use the parameters $C_1 = \sqrt{n_1 n_2/|\Omega|}$ and $C_2 = 1$ (as well as $\tau = 1$). The choice $C_2 = 1$ is based on empirical testing and has not been tuned. Smaller values of these constants will undoubtedly yield faster convergence for certain problems. We could use $L = 1$ as a bound for $\|A\|$, but following (5.5) under an incoherence assumption, we expect a local bound to scale as $\sqrt{|\Omega|/(n_1 n_2)}$. We therefore take $L = \min\{1.4\sqrt{|\Omega|/(n_1 n_2)}, 1\}$. Finally, the nonergodic version of WARPd converged faster than the ergodic version for the following, and so we report computational results for the nonergodic version.

5.3. Current state-of-the-art methods. Below we provide a brief summary of four state-of-the-art methods for matrix completion based on nuclear norm minimization, to which we compare our algorithm in subsection 5.4. We do not claim that this is a complete list. Rather, we selected these methods for comparison based on their effectiveness, the variation of approaches, their popularity, and the availability of well-documented code.[15]

5.3.1. Nuclear norm regularized linear least squares (NNLS). NNLS [117] (see also [70] for a similar approach) can be considered a generalization of the fast iterative shrinkage-thresholding algorithm (FISTA; see [16]) to matrix problems. The algorithm considers the problem $\min \mu\|M\|_1 + \|A(M) - b\|_2^2/2$ with linesearch, continuation for different μ, and SVD truncations for efficiency. The partial SVD is computed using PROPACK, with an update rule for the approximate rank. The code can be found at https://blog.nus.edu.sg/mattohkc/softwares/nnls/ and we use the default parameters throughout.

5.3.2. Singular value thresholding (SVT). SVT [27] performs shrinkage iterations to solve a smoothed problem (addition of an $\|M\|_2^2$ term), taking advantage of the sparsity and low rankness of the matrix iterates for approximation of the SVT operator. The algorithm uses low-rank SVD factorizations to reduce memory consumption and PROPACK. The code can be found at https://statweb.stanford.edu/~candes/software/svt/code.html and we use the default parameters suggested by [27] throughout. These parameters are based on empirical testing in [27]—we found the alternative parameters with guaranteed convergence (related to a smaller step size) to perform worse than the results we report.

5.3.3. Fixed point continuation with approximate SVD (FPCA). FPCA [90] has some similarities with SVT in that it makes use of shrinkage operations. However, the Lagrangian form of the problem, $\min \mu\|M\|_1 + \|A(M) - b\|_2^2/2$, is solved with continuation for a sequence of parameters μ. For the shrinkage operator, an approximate SVD is computed using a fast Monte Carlo algorithm [52]. The code can be found at https://www.math.ucdavis.edu/~sqma/FPCA.html, and we use the given routine that selects parameters throughout.

5.3.4. Augmented Lagrange multiplier method (ALM). ALM [84] is based on the augmented Lagrangian function $\|M\|_1 + \langle Y, P_\Omega(M) - M - E\rangle + \mu\|P_\Omega(M) - M - E\|_2^2/2$ (E is the difference between M and $P_\Omega(M)$ and Y is a dual variable). The general method of augmented Lagrange multipliers [22] applies simple updates rules for M, Y, and E for a

[15] The listed methods are all first-order methods. While nuclear norm minimization can be reformulated as a semidefinite program and solved by off-the-shelf interior point solvers, typically such methods have difficulty treating matrices larger than $n \sim 100$ because the complexity of computing each step grows quickly with n (due to reliance on second-order information of the objective function). To overcome this scalability issue, the literature has focused on first-order methods.

Table 1

Computational times for a wide variety of parameter values for the low-rank random matrix recovery problem. All times are averaged over five runs and the best average for each experiment is shown in green. We report an "NC" if convergence was not obtained after 5,000 *iterations or after* 100,000*s.*

n	r	p	Time (s), $\texttt{tol} = 10^{-4}$					Time (s), $\texttt{tol} = 10^{-6}$				
			WARPd	NNLS	SVT	FPCA	ALM	WARPd	NNLS	SVT	FPCA	ALM
1000	10	0.14	1.2	3.5	2.6	1.9	4.5	2.2	6.3	3.8	2.8	9.0
	30	0.40	3.8	5.2	7.2	4.6	7.0	6.3	8.8	10.9	6.4	8.1
	60	0.57	6.3	10.9	14.3	8.4	8.5	11.3	25.4	24.2	12.1	11.5
5000	10	0.02	4.3	8.9	335.0	1093.5	203.7	9.5	18.0	NC	NC	465.7
	30	0.08	17.9	20.7	50.4	69.2	165.7	39.6	47.7	83.7	129.5	345.9
	60	0.19	58.2	67.1	156.0	81.3	194.5	100.5	116.1	257.3	189.0	443.0
10000	10	0.01	12.6	15.1	356.7	NC	1335.7	14.3	28.0	NC	NC	1787.1
	30	0.04	57.8	45.0	1132.9	7312.0	1237.9	108.0	87.0	1810.2	NC	1639.4
	60	0.10	159.8	134.9	496.5	432.8	1160.7	298.2	281.7	836.2	507.9	1614.1
20000	10	0.005	27.6	32.7	9114.3	NC	4085.8	40.7	50.5	NC	NC	NC
	30	0.020	158.6	122.7	384.0	NC	3732.3	236.6	174.8	1283.7	NC	9349.2
	60	0.049	430.3	279.3	1296.8	NC	6704.4	753.6	614.2	4461.5	NC	9597.1

sequence of increasing μ's. In the case of matrix completion, a numerical difficulty is that for large μ, the thresholding procedure (computed via an SVD) becomes numerically expensive. An inexact version of ALM was developed in [84] to overcome this issue and was shown to converge (the inexactness precludes a convergence rate analysis). The code can be found at https://zhouchenlin.github.io/, and we use the default parameters throughout. The code uses PROPACK and a simple update rule for the number of desired singular values.

5.4. Numerical examples. As our first experiment, we perform the following benchmark test often used in the literature [27, 84, 90]. We generate two independent standard Gaussian matrices $M_L \in \mathbb{R}^{n\times r}$, $M_R \in \mathbb{R}^{(n+20)\times r}$ and set $\mathbb{x} = M_L M_R^* \in \mathbb{R}^{n\times(n+20)}$. Given $p \in (0,1)$, we then sample as described in subsection 5.1. We measure the time taken by each algorithm to achieve a relative error below $\texttt{tol}$, measured in the Frobenius norm. Table 1 shows the results, where we have taken the average time over five runs for each parameter selection and we report NC (highlighted in red) if convergence was not obtained after 5,000 iterations or 100,000s. For each parameter selection, we have highlighted the best average in green. Experiments were run on a modest desktop computer with a 3.4 GHz CPU. We have chosen a high accuracy tolerance $\texttt{tol} = 10^{-6}$, as well as a moderate accuracy tolerance $\texttt{tol} = 10^{-4}$.

For moderate $n < 10{,}000$, WARPd is the fastest method. For large r when n is large, NNLS is faster than WARPd, but the two methods are roughly comparable. A possible reason for the NCs is the chosen value of p—larger p generally gives an easier problem with better convergence properties, though sometimes larger computational times due to the larger number of nonzero entries in the sparse matrices. We have deliberately shown results for varied p to probe the robustness of algorithms for more challenging problems. In summary, Table 1 shows clear benefits of the acceleration and demonstrates the speed and robustness of WARPd across a broad range of matrix sizes, ranks, and sampling ratios.

We now consider a real data example and an *approximately* low-rank matrix. We took the data set https://dataportal.orr.gov.uk/statistics/usage/estimates-of-station-usage/ of the

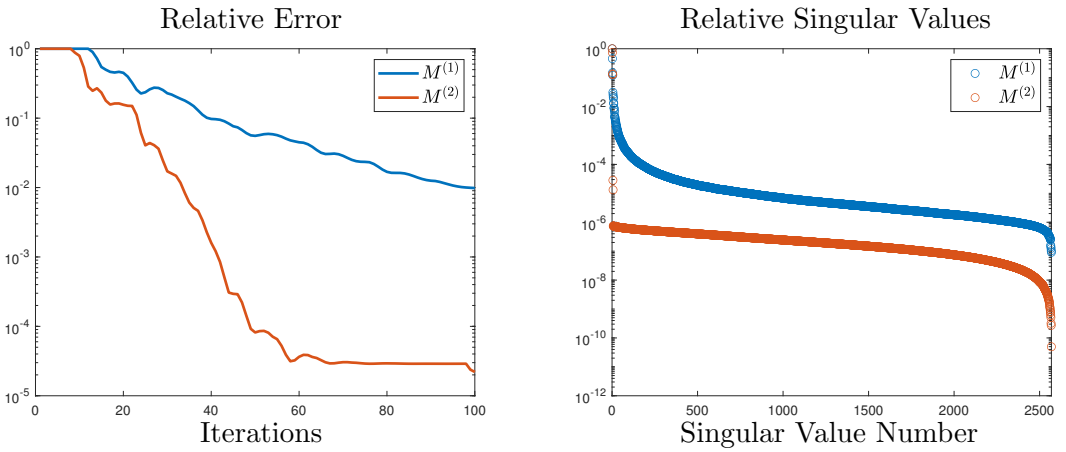

Figure 4. *Results for the experiment with real data. Left: Relative error in the Frobenius norm. Right: Relative singular values (singular values normalized by the largest singular value) for each matrix.*

locations of all 2,569 railway stations in Great Britain. We considered two matrices, $M^{(1)} \in \mathbb{R}^{2569\times 2569}$ corresponding to the geodesic distance between all pairs of stations (rounded to the nearest 10m) and $M^{(2)}$ corresponding to the distance squared, both with $p = 0.07$ (so that only approximately 7% of the entries are sampled). Figure 4 (left) shows the convergence for WARPd with $\epsilon = 10^{-10}$ and $\delta = C_2\epsilon$. The accuracy of solutions of (1.2) is achieved in around 100 and 60 iterations, respectively, with linear convergence down to this bound. Figure 4 (right) shows the singular values of both matrices and explains why recovering $M^{(2)}$ is easier than $M^{(1)}$. For example, the best rank six approximations of each matrix satisfy

$$\|M_6^{(1)} - M^{(1)}\|_2/\|M^{(1)}\|_2 \approx 0.0359 \quad \text{and} \quad \|M_6^{(2)} - M^{(2)}\|_2/\|M^{(2)}\|_2 \approx 1.16 \times 10^{-5}.$$

6. Examples with nontrivial matrix B.

Our final section consider examples of Example 1.1 with $D_{\mathcal{Z}}(z,\hat{z}) = \frac{1}{2}\|z-\hat{z}\|_{l^2}^2$, $\mathcal{F}(\cdot) = \|\cdot\|_{l^1}$, and nontrivial matrix B. We provide theorems for two common scenarios: l^1-analysis and TV regularization. We end with a numerical example involving shearlets and total generalized variation (TGV), combined with an iterative reweighting of the l^1-norm.

6.1. Two example theorems.

6.1.1. l^1-analysis with tight frames.

We consider the problem (with $\mathcal{J} = 0$)

$$\min_{x\in\mathbb{C}^N} \|D^*x\|_{l^1} \quad \text{such that} \quad \|Ax-b\|_{l^2} \leq \epsilon, \tag{6.1}$$

where the columns of D provide a tight frame.[16] Common examples of D include oversampled DFT, Gabor frames, curvelets, shearlets, concatenations of orthonormal bases, etc. Without loss of generality, we assume that DD^* is the identity. See [53, 55, 95, 113] for examples where an analysis approach (6.1) has advantages over a synthesis approach such as (3.1).

The following definition (which imposes no incoherence restriction on the dictionary) is a natural generalization of the well-known restricted isometry property.

[16] Our results can be extended to frames that are not tight, but the analysis is more complicated.

Definition 6.1 ([28]). *Let $s \in \mathbb{N}$ and let Σ_s denote the union of all subspaces spanned by all subsets of s columns of D. We say that the measurement matrix A obeys the restricted isometry property adapted to D (D-RIP) with constant $\delta_s = \delta_s(A, D)$ if*

$$(1 - \delta_s)\|v\|_{l^2}^2 \leq \|Av\|_{l^2}^2 \leq (1 + \delta_s)\|v\|_{l^2}^2 \quad \forall v \in \Sigma_s. \tag{6.2}$$

For explicit examples where Definition 6.1 holds, see [28]. This definition yields the following theorem, whose proof is partly based on the arguments of [28].

Theorem 6.2. *Let $t > s$ and set $\rho = s/t < 1$. Suppose that*

$$\omega(A, D) := 1 - \rho - \frac{\sqrt{\rho(1 + \delta_t(A, D))}}{\sqrt{1 - \delta_{s+t}(A, D)}} > 0.$$

Then the approximate sharpness condition (1.5) holds for (6.1) for any $\mathbb{x} \in \mathbb{C}^N$ with

$$C_1 = \frac{1 + \sqrt{\rho^2 + \rho} - \omega(A, D)}{\omega(A, D)\sqrt{s}}, \quad C_2 = \frac{\sqrt{s}\left(1 + \sqrt{\rho^2 + \rho} - \omega(A, D)\right)^{-1}}{\sqrt{1 - \delta_{s+t}(A, D)}}, \tag{6.3}$$
$$c(\mathbb{x}, b) = 2\sigma_{\mathbf{s}}(D^*\mathbb{x})_{l^1} + C_2(\|A\mathbb{x} - b\|_{l^2} + \epsilon).$$

It follows that the conclusions of Theorems 1.2 and 1.3 hold.

Proof. See section SM5. ∎

In summary, if A satisfies the D-RIP, then WARPd provides accelerated recovery via (6.1). Using $\delta_t < \delta_{s+t}$, the condition $\omega(A, D) > 0$ is satisfied if $\delta_{s+t}(A, D) < \frac{1+\rho^2-3\rho}{1+\rho^2-\rho}$.

6.1.2. Total variation minimization. TV minimization [111] is widely used for image restoration tasks such as denoising, deblurring, and inpainting [25, 37, 38, 101], as well as compressed sensing [33, 87]. We consider a two-dimensional signal $X \in \mathbb{C}^{\hat{N} \times \hat{N}}$. For vectorized $x = \mathrm{vect}(X) \in \mathbb{C}^N$, $N = \hat{N}^2$, $\nabla \in \mathbb{C}^{2N \times N}$ is given by $\nabla = (\nabla_1 \ \nabla_2)^\top$ with

$$[\nabla_1 X]_{i_1,i_2} = X_{i_1+1,i_2} - X_{i_1,i_2}, \quad [\nabla_2 X]_{i_1,i_2} = X_{i_1,i_2+1} - X_{i_1,i_2},$$

where $X_{\hat{N}+1,i_2} = X_{1,i_2}, X_{i_1,\hat{N}+1} = X_{i_1,1}$. The periodic anisotropic TV-seminorm is given by

$$\|X\|_{\mathrm{TV}} = \|x\|_{\mathrm{TV}} = \|\nabla x\|_{l^1} = \sum_{i_1,i_2=1}^{\hat{N}} |X_{i_1+1,i_2} - X_{i_1,i_2}| + |X_{i_1,i_2+1} - X_{i_1,i_2}|.$$

We consider the problem (with $\mathcal{J} = 0$ and $B = \nabla$)

$$\min_{x \in \mathbb{C}^N} \|x\|_{\mathrm{TV}} \quad \text{such that} \quad \|Ax - b\|_{l^2} \leq \epsilon. \tag{6.4}$$

For accurate and stable recovery guarantees for this problem, see [96, 97], which exploit the connection between the TV-seminorm and Haar wavelet coefficients. For sampling strategies for Fourier and binary measurements, see [5, 78, 102]. It is beyond the scope of this paper to

discuss how all of these results fit into our framework, and so we consider the following general setting. Recall that a matrix $A \in \mathbb{C}^{m\times N}$ satisfies the restricted isometry property (RIP) of order s if there exists $\delta_s(A) \in (0,1)$ such that for any s-sparse vector $z \in \mathbb{C}^N$,

$$(1-\delta_s(A))\|z\|_{l^2}^2 \leq \|Az\|_{l^2}^2 \leq (1+\delta_s(A))\|z\|_{l^2}^2.$$

The following theorem [6, Theorem 17.17][17] provides a version of (1.5) (it is possible to chase down the explicit constants by studying the proof), and, to facilitate Corollary 6.4, we have stated the conclusion slightly differently to [6].

Theorem 6.3 ([6]). *Let $\hat{N} \geq s \geq 2$, $\Phi \in \mathbb{R}^{\hat{N}^2\times\hat{N}^2}$ be the matrix of the two-dimensional discrete Haar wavelet sparsifying transform and $A \in \mathbb{C}^{m\times\hat{N}^2}$. Suppose that $A\Phi$ has the RIP of order $t \gtrsim s\log(\hat{N})\log^2(2\hat{N}^2/s)$ with constant $\delta_t(A\Phi) \leq 1/2$. Then for any $x, \hat{x} \in \mathbb{C}^{\hat{N}^2}$,*

$$\|\hat{x}-x\|_{l^2} \lesssim \left(\|\hat{x}\|_{\mathrm{TV}} - \|x\|_{\mathrm{TV}} + \sigma_{\mathbf{s}}(\nabla x)_{l^1}\right)/\sqrt{s\log(\hat{N})} + (\|A\hat{x}-b\|_{l^2}-\epsilon) + (\|Ax-b\|_{l^2}+\epsilon).$$

The following shows WARPd allows accelerated recovery via (6.4) if $A\Phi$ satisfies the RIP.

Corollary 6.4. *Suppose that the conditions of Theorem 6.3 hold. Then the approximate sharpness condition (1.5) holds for any $\mathbb{x} \in \mathbb{C}^{\hat{N}^2}$, with*

$$C_1 \lesssim 1/\sqrt{s\log(\hat{N})}, \quad C_2 \lesssim \sqrt{s\log(\hat{N})}, \quad c(\mathbb{x},b) = \sigma_{\mathbf{s}}(\nabla\mathbb{x})_{l^1} + C_2(\|A\mathbb{x}-b\|_{l^2}+\epsilon),$$

for the problem (6.4). It follows that the conclusions of Theorems 1.2 and 1.3 hold.

6.2. A numerical example involving shearlets and TGV. The goal of this final numerical example is to demonstrate the flexibility of our algorithm, rather than promote a particular transform or regularizer. Figure 5 (left) shows the used test image. We let A be a DFT, 15% subsampled according to an inverse square law density [79]. This sampling pattern has recently been shown to be optimal for TV reconstruction [5]. The measurements are corrupted with 5% Gaussian noise. We first use WARPd to reconstruct the image via (6.4); the results are shown in Figure 5 (middle). While convergence to a solution of (6.4) was rapid, the reconstruction shows the typical artifacts of TV regularization such as staircasing. Next, we replace the TV regularizer with the (discrete) TGV regularizer [25]

$$\mathrm{TGV}_\alpha^2(x) = \min_{v\in\mathbb{C}^{2N}} \alpha_1\|\nabla x - v\| + \alpha_0 \left\| \begin{pmatrix} \nabla_1 v_x & \frac{1}{2}(\nabla_2 v_x + \nabla_1 v_y) \\ \frac{1}{2}(\nabla_2 v_x + \nabla_1 v_y) & \nabla_2 v_y \end{pmatrix} \right\|_1, \tag{6.5}$$

which has been proposed to improve on these issues by involving higher-order derivatives. The improved results are shown in Figure 5 (right). Again, convergence to a solution of the optimization problem was rapid.

To improve the reconstruction further, we consider

$$\min_{x\in\mathbb{C}^N} \|WD^*x\|_{l^1} + \mathrm{TGV}_\alpha^2(x) \quad \text{such that} \quad \|Ax-b\|_{l^2} \leq \epsilon, \tag{6.6}$$

[17] The result of [6] considered the isotropic version of the TV-seminorm. Both versions are equivalent up to a factor of $\sqrt{2}$ and hence the theoretical result is the same. We have considered the anisotropic version to fit into (1.2). It is straightforward to adapt WARPd to the isotropic case by adapting the proximal maps.

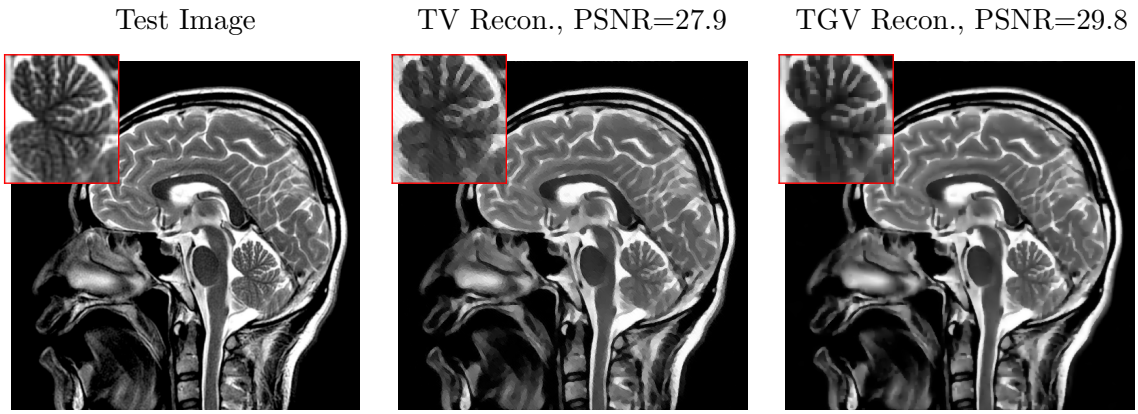

Figure 5. *Left:* 512×512 *test image with pixel values scaled to* $[0, 1]$*; the red box shows a zoomed in section. Middle: Converged reconstruction using TV. Right: Converged reconstruction using TGV (using* $\alpha_0 = 0.4$ *and* $\alpha_1 = 0.2$*; see* (6.5) *for meaning of parameters). Both reconstructions were computed using WARPd.*

where W denotes a diagonal scaling matrix and D corresponds to a shearlet frame. We used the MATLAB `ShearLab` package in this example, which can be found at https://shearlab.math.lmu.de/. Throughout this paper, we have so far only discussed numerical examples for WARPd, since the results of WARPd-SR are similar (if not better). For completeness, in this example we also consider WARPd-SR to demonstrate that it sometimes leads to better reconstructions. The weight matrix W is updated after each call to `InnerIt` in Algorithm 2.3 (or `InnerIt-SR` in Algorithm 2.6) according to

$$W_{jj} = \frac{1}{\max\{[D^*x]_{jj}, 10^{-5}\}} \times \frac{\sum_{k \in I(j)} \max\{[D^*x]_{kk}, 10^{-5}\}}{|I(j)|}, \tag{6.7}$$

where $I(j)$ denotes the set of indices corresponding to the shearlet scale containing the index j, and x is the current reconstruction. We initialized the weights according to (6.7) with $x = A^*b$. The update rule takes into account the difference in magnitudes of the shearlet coefficients of an image at different scales—see [8, 89] and [6, section 4.6] for the motivation of similar update rules. Figure 6 shows the reconstruction using WARPd (left) and WARPd-SR (middle), which show a marked improvement on the results of Figure 5. Moreover, WARPd-SR shows a better reconstruction of the fine details of the image. Figure 6 (right) plots the relative MSE error between the reconstruction and the image against the number of inner iterations. We also show the MSE for nonrestarted primal-dual iterations (dashed lines). The benefit of acceleration is clear with WARPd and WARPd-SR converging in under 40 iterations. This example demonstrates that WARPd and WARPd-SR can easily handle more complicated mixed regularization problems such as (6.6).

7. Concluding remarks.

We have provided an accelerated algorithm for the recovery problem (1.1) via the optimization problem (1.2), under the assumption (1.5) of *approximate sharpness*. Linear convergence is achieved, down to the approximation term in (1.5). We also translated this result into a statement about the complexity of stable and accurate NNs. Our

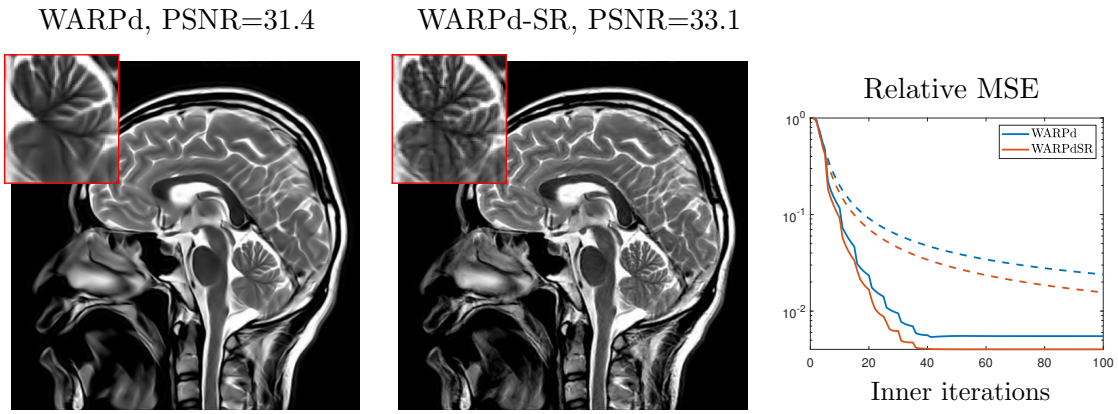

Figure 6. *Left: Reconstruction using WARPd and* (6.6)*. Middle: Reconstruction using WARPd-SR. Right: The relative MSE as a function of the number of inner iterations. The dashed lines correspond to nonrestarted primal-dual iterations.*

framework was demonstrated on several important problems, and it was shown that WARPd compares favorably with specialized state-of-the-art methods.

There are many further lines of work that could build on these results. Straightforward extensions of the algorithm include optimization problems with additional terms similar to those in (1.2), and also restrictions to convex sets (e.g., if x represents a matrix, we may want to enforce that it is Hermitian). The rapid solution of (1.2) could also lend itself to bilevel optimization problems, where small computational cost is essential. Such problems are increasingly relevant to learning-based methods. It is likely that other methods based on similar restart techniques and assumptions similar to (1.5) could be developed using different first-order methods [100]. For example, Table 1 suggests that an acceleration scheme based on FISTA may be faster for some problems. Additionally, it may be possible to treat more general discrepancy terms, such as the Kullback–Leibler divergence or Wasserstein metric. In the case of primal-dual iterations for a saddle point representation of the optimization problem, such generalizations may require specific structures to enable error bounds similar to Proposition 2.1. Finally, we assumed some prior knowledge of suitable constants appearing in (1.5). It may be possible to develop methods that learn suitable constants for (1.5) and when to restart. Another option is the use of logarithmic grids to search for parameters [109].

Added note. While this paper was being finalized, WARPd was used for recovering high-dimensional, Hilbert-valued functions from limited samples [2]. Here a weighted robust null space property allows the proof of an approximate sharpness inequality and accelerated convergence.

Acknowledgments. I would like to thank Vegard Antun and Luca Gazdag for providing feedback on a draft of this paper. I would also like to thank the referees for their suggestions that led to the improvement of this paper. I am also grateful to Dominik Stoeger for discussions regarding the current state of the art in matrix completion. Special thanks are also due to Elizabeth Sawchuk for allowing me to use the image of her labrador in Figure 1.

REFERENCES

[1] B. Adcock, A. Bao, and S. Brugiapaglia, *Correcting for unknown errors in sparse high-dimensional function approximation*, Numer. Math., 142 (2019), pp. 667–711.

[2] B. Adcock, S. Brugiapaglia, N. Dexter, and S. Moraga, *On Efficient Algorithms for Computing Near-Best Polynomial Approximations to High-Dimensional, Hilbert-Valued Functions from Limited Samples*, preprint, arXiv:2203.13908, 2022.

[3] B. Adcock, S. Brugiapaglia, and M. King-Roskamp, *Do log factors matter? On optimal wavelet approximation and the foundations of compressed sensing*, Found. Comput. Math. (2021), pp. 1–61.

[4] B. Adcock and N. Dexter, *The gap between theory and practice in function approximation with deep neural networks*, SIAM J. Math. Data Sci., 3 (2021), pp. 624–655.

[5] B. Adcock, N. Dexter, and Q. Xu, *Improved recovery guarantees and sampling strategies for TV minimization in compressive imaging*, SIAM J. Imaging Sci., 14 (2021), pp. 1149–1183.

[6] B. Adcock and A. Hansen, *Compressive Imaging: Structure, Sampling, Learning*, Cambridge University Press, Cambridge, UK, 2021.

[7] B. Adcock, A. C. Hansen, C. Poon, and B. Roman, *Breaking the coherence barrier: A new theory for compressed sensing*, in Forum Math. Sigma, 5 (2017).

[8] R. Ahmad and P. Schniter, *Iteratively reweighted ℓ_1 approaches to sparse composite regularization*, IEEE Transa. Comput. Imaging, 1 (2015), pp. 220–235.

[9] A. A. Amini and M. J. Wainwright, *High-dimensional analysis of semidefinite relaxations for sparse principal components*, in Proceedings of the International Symposium on Information Theory, IEEE, 2008, pp. 2454–2458.

[10] Y. Amit, M. Fink, N. Srebro, and S. Ullman, *Uncovering shared structures in multiclass classification*, in Proceedings of ICML, 2007, pp. 17–24.

[11] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, *On instabilities of deep learning in image reconstruction and the potential costs of AI*, Proc. Natl. Acad. Sci. USA, 117 (2020), pp. 30088–30095.

[12] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, *Solving inverse problems using data-driven models*, Acta Numer., 28 (2019), pp. 1–174.

[13] A. Auslender and M. Teboulle, *Interior gradient and proximal methods for convex and conic optimization*, SIAM J. Optim., 16 (2006), pp. 697–725.

[14] A. Bastounis and A. C. Hansen, *On the absence of uniform recovery in many real-world applications of compressed sensing and the restricted isometry property and nullspace property in levels*, SIAM J. Imaging Sci., 10 (2017), pp. 335–371.

[15] A. Bastounis, A. C. Hansen, and V. Vlačić, *The Extended Smale's 9th Problem – On Computational Barriers and Paradoxes in Estimation, Regularisation, Computer-Assisted Proofs and Learning*, 2021, https://arxiv.org/abs/2110.15734.

[16] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[17] S. Becker, J. Bobin, and E. J. Candès, *NESTA: A fast and accurate first-order method for sparse recovery*, SIAM J. Imaging Sci., 4 (2011), pp. 1–39.

[18] S. R. Becker, E. J. Candès, and M. C. Grant, *Templates for convex cone problems with applications to sparse signal recovery*, Math. Program. Comput., 3 (2011), 165.

[19] A. Belloni, V. Chernozhukov, and L. Wang, *Square-root LASSO: pivotal recovery of sparse signals via conic programming*, Biometrika, 98 (2011), pp. 791–806.

[20] A. Belloni, V. Chernozhukov, and L. Wang, *Pivotal estimation via square-root LASSO in nonparametric regression*, Ann. Statist., 42 (2014), pp. 757–788.

[21] M. Benning and M. Burger, *Modern regularization methods for inverse problems*, Acta Numer., 27 (2018), pp. 1–111.

[22] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 2014.

[23] J. Bigot, C. Boyer, and P. Weiss, *An analysis of block sampling strategies in compressed sensing*, IEEE Trans. Inform. Theory, 62 (2016), pp. 2125–2139.

[24] C. Boyer, J. Bigot, and P. Weiss, *Compressed sensing with structured sparsity and structured acquisition*, Appl. Comput. Harmon. Anal., 46 (2019), pp. 312 – 350.

[25] K. Bredies, K. Kunisch, and T. Pock, *Total generalized variation*, SIAM J. Imaging Sci., 3 (2010), pp. 492–526.
[26] T. A. Bubba, G. Kutyniok, M. Lassas, M. März, W. Samek, S. Siltanen, and V. Srinivasan, *Learning the invisible: A hybrid deep learning-shearlet framework for limited angle computed tomography*, Inverse Probl, 35 (2019), 064002.
[27] J.-F. Cai, E. J. Candès, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim., 20 (2010), pp. 1956–1982.
[28] E. J. Candes, Y. C. Eldar, D. Needell, and P. Randall, *Compressed sensing with coherent and redundant dictionaries*, Appl. Comput. Harmon. Anal., 31 (2011), pp. 59–73.
[29] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski, *Phase retrieval via matrix completion*, SIAM Rev., 57 (2015), pp. 225–251.
[30] E. J. Candès et al., *Compressive sampling*, in Proceedings of ICM, vol. 3, 2006, pp. 1433–1452.
[31] E. J. Candes and Y. Plan, *Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements*, IEEE Trans. Inform. Theory, 57 (2011), pp. 2342–2359.
[32] E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717–772.
[33] E. J. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, 52 (2006), pp. 489–509.
[34] E. J. Candes, J. K. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Commun. Pure Appl. Math., 59 (2006), pp. 1207–1223.
[35] E. J. Candes, T. Strohmer, and V. Voroninski, *Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming*, Comm. Pure Appl. Math., 66 (2013).
[36] E. J. Candès and T. Tao, *The power of convex relaxation: Near-optimal matrix completion*, IEEE Trans. Inform. Theory, 56 (2010), pp. 2053–2080.
[37] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97.
[38] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, *An introduction to total variation for image analysis*, in Theoretical Foundations and Numerical Methods for Sparse Recovery, de Gruyter, Berlin, 2010, pp. 263–340.
[39] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.
[40] A. Chambolle and T. Pock, *An introduction to continuous optimization for imaging*, Acta Numer., 25 (2016), pp. 161–319.
[41] A. Chambolle and T. Pock, *On the ergodic convergence rates of a first-order primal–dual algorithm*, Math. Program., 159 (2016), pp. 253–287.
[42] V. Chandrasekaran and M. I. Jordan, *Computational and statistical tradeoffs via convex relaxation*, Proc. Natl. Acad. Sci. USA, 110 (2013), pp. E1181–E1190.
[43] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, *The convex geometry of linear inverse problems*, Found. Comput. Math., 12 (2012), pp. 805–849.
[44] P. Chen and D. Suter, *Recovering the missing components in a large noisy low-rank matrix: Application to SFM*, IEEE Trans. Pattern Anal. Mach. Intell., 26 (2004), pp. 1051–1063.
[45] X. Chen, J. Liu, Z. Wang, and W. Yin, *Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds*, in Advances in Neural Information Processing Systems, 2018, pp. 9061–9071.
[46] M. J. Colbrook, V. Antun, and A. C. Hansen, *The difficulty of computing stable and accurate neural networks: On the barriers of deep learning and Smale's 18th problem*, Proc. Natl. Acad. Sci. USA, 119 (2022).
[47] I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Comm. Pure Appl. Math., 57 (2004), pp. 1413–1457.
[48] R. DeVore, B. Hanin, and G. Petrova, *Neural network approximation*, Acta Numer., 30 (2021), pp. 327–444.
[49] L. Ding and Y. Chen, *Leave-one-out approach for matrix completion: Primal and dual analysis*, IEEE Trans. Inform. Theory, 66 (2020), pp. 7274–7301.
[50] D. L. Donoho, *Compressed sensing*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1289–1306.

[51] D. L. Donoho and Y. Tsaig, *Fast solution of ℓ_1-norm minimization problems when the solution may be sparse*, IEEE Trans. Inform. Theory, 54 (2008), pp. 4789–4812.
[52] P. Drineas, R. Kannan, and M. W. Mahoney, *Fast Monte Carlo algorithms for matrices* II*: Computing a low-rank approximation to a matrix*, SIAM J. Comput., 36 (2006), pp. 158–183.
[53] M. F. Duarte and R. G. Baraniuk, *Spectral compressive sensing*, Appl. Comput. Harmon. Anal., 35 (2013), pp. 111–129.
[54] J. Eckstein, *Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming*, Math. Oper. Res., 18 (1993), pp. 202–226.
[55] M. Elad, P. Milanfar, and R. Rubinstein, *Analysis versus synthesis in signal priors*, Inverse Problems, 23 (2007), 947.
[56] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, *Block-sparse signals: Uncertainty relations and efficient recovery*, IEEE Trans. Signal Process., 58 (2010), pp. 3042–3054.
[57] A. Evgeniou and M. Pontil, *Multi-task feature learning*, Adv. Neural Inf. Process. Syst., 19 (2007).
[58] A. Fannjiang and T. Strohmer, *The numerics of phase retrieval*, Acta Numer., 29 (2020), pp. 125–228.
[59] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, *Adversarial attacks on medical machine learning*, Science, 363 (2019), pp. 1287–1289.
[60] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, Springer, New York, 2013.
[61] M. P. Friedlander, H. Mansour, R. Saab, and Ö. Yilmaz, *Recovering compressively sampled signals using partial support information*, IEEE Trans. Inform. Theory, 58 (2012), pp. 1122–1134.
[62] D. Gilton, G. Ongie, and R. Willett, *Deep equilibrium architectures for inverse problems in imaging*, IEEE Trans. Comput. Imaging, 7 (2021), pp. 1123–1133.
[63] D. Gross, *Recovering low-rank matrices from few coefficients in any basis*, IEEE Trans. Inform. Theory, 57 (2011), pp. 1548–1566.
[64] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, *Quantum state tomography via compressed sensing*, Phys. Rev. Lett., 105 (2010), 150401.
[65] S. Gu, L. Zhang, W. Zuo, and X. Feng, *Weighted nuclear norm minimization with application to image denoising*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2862–2869.
[66] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, *Learning a variational network for reconstruction of accelerated MRI data*, Magn. Reson. Med., 79 (2018), pp. 3055–3071.
[67] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The LASSO and Generalizations*, CRC Press, Boca Raton, FE, 2015.
[68] J. Hertrich, S. Neumayer, and G. Steidl, *Convolutional proximal neural networks and plug-and-play algorithms*, Linear Algebra Appl., 631 (2021), pp. 203–234.
[69] Y. Huang et al., *Some investigations on robustness of deep learning in limited angle tomography*, in MICCAI, Springer, New York, 2018, pp. 145–153.
[70] S. Ji and J. Ye, *An accelerated gradient method for trace norm minimization*, in Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 457–464.
[71] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, *Deep convolutional neural network for inverse problems in imaging*, IEEE Trans. Image Process., 26 (2017), pp. 4509–4522.
[72] A. Jones, A. Tamtögl, I. Calvo-Almazán, and A. Hansen, *Continuous compressed sensing for surface dynamical processes with helium atom scattering*, Sci. Rep., 6 (2016), 27776.
[73] M. Kabanava, R. Kueng, H. Rauhut, and U. Terstiege, *Stable low-rank matrix recovery via null space properties*, Inf. Inference, 5 (2016), pp. 405–441.
[74] F. Knoll et al., *Advancing machine learning for MR image reconstruction with an open competition: Overview of the* 2019 *fastMRI challenge*, Magn. Reson. Med., 84 (2020), pp. 3054–3070.
[75] E. Kobler, A. Effland, K. Kunisch, and T. Pock, *Total Deep Variation: A Stable Regularizer for Inverse Problems*, arXiv:2006.08789, 2020.
[76] Y. Koren, R. Bell, and C. Volinsky, *Matrix factorization techniques for recommender systems*, Computer, 42 (2009), pp. 30–37.
[77] F. Krahmer and D. Stöger, *On the convex geometry of blind deconvolution and matrix completion*, Comm. Pure Appl. Math., 74 (2021), pp. 790–832.

[78] F. Krahmer and R. Ward, *Stable and robust sampling strategies for compressive imaging*, IEEE Trans. Image Process., 23 (2013), pp. 612–622.
[79] F. Krahmer and R. Ward, *Stable and robust sampling strategies for compressive imaging*, IEEE Trans. Image Process., 23 (2014), pp. 612–622.
[80] R. Kueng, H. Rauhut, and U. Terstiege, *Low rank matrix recovery from rank one measurements*, Appl. Comput. Harmon. Anal., 42 (2017), pp. 88–116.
[81] G. Kutyniok and W.-Q. Lim, *Optimal compressive imaging of fourier data*, SIAM J. Imaging Sci., 11 (2018), pp. 507–546.
[82] R. M. Larsen, *PROPACK—Software for Large and Sparse SVD Calculations*, URL http://sun. stanford. edu/rmunk/PROPACK, 2004.
[83] C. Li and B. Adcock, *Compressed sensing with local structure: Uniform recovery guarantees for the sparsity in levels class*, Appl. Comput. Harmon. Anal., 46 (2019), pp. 453–477.
[84] Z. Lin, M. Chen, and Y. Ma, *The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices*, arXiv:1009.5055, 2010.
[85] J. Liu, X. Chen, Z. Wang, and W. Yin, *ALISTA: Analytic weights are as good as learned weights in LISTA*, in Proceedings of ICLR, 2018.
[86] Y.-K. Liu, *Universal low-rank matrix recovery from Pauli measurements*, Adv. Neural Inf. Process. Syst, 24 (2011), pp. 1638–1646.
[87] M. Lustig, D. Donoho, and J. M. Pauly, *Sparse MRI: The application of compressed sensing for rapid MR imaging*, Magn. Reson. Med., 58 (2007), pp. 1182–1195.
[88] A. I. Lvovsky and M. G. Raymer, *Continuous-variable optical quantum-state tomography*, Rev. Modern Phys., 81 (2009), 299.
[89] J. Ma and M. März, *A Multilevel Based Reweighting Algorithm with Joint Regularizers for Sparse Recovery*, arXiv:1604.06941, 2016.
[90] S. Ma, D. Goldfarb, and L. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Math. Program., 128 (2011), pp. 321–353.
[91] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd ed., Academic Press, New York, 2008.
[92] M. T. McCann, K. H. Jin, and M. Unser, *Convolutional neural networks for inverse problems in imaging: A review*, IEEE Signal Process Mag., 34 (2017), pp. 85–95.
[93] V. Monga, Y. Li, and Y. C. Eldar, *Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing*, IEEE Signal Process Mag., 38 (2021), pp. 18–44.
[94] M. J. Muckley et al., *State-of-the-art Machine Learning MRI Reconstruction in* 2020*: Results of the Second fastMRI Challenge*, arXiv:2012.06318, 2020.
[95] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, *The cosparse analysis model and algorithms*, Appl. Comput. Harmon. Anal., 34 (2013), pp. 30–56.
[96] D. Needell and R. Ward, *Near-optimal compressed sensing guarantees for total variation minimization*, IEEE Trans. Image Process., 22 (2013), pp. 3941–3949.
[97] D. Needell and R. Ward, *Stable image reconstruction using total variation minimization*, SIAM J. Imaging Sci., 6 (2013), pp. 1035–1058.
[98] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Appl. Optim. 87, Springer, New York, 2003.
[99] Y. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program., 103 (2005), pp. 127–152.
[100] M. Neyra-Nesterenko and B. Adcock, *Stable, Accurate and Efficient Deep Neural Networks for Inverse Problems with Analysis-Sparse Models*, 2022, https://arxiv.org/abs/2203.00804.
[101] S. Parisotto, J. Lellmann, S. Masnou, and C. Schonlieb, *Higher-order total directional variation: Imaging applications*, SIAM J. Imaging Sci., 13 (2020), pp. 2063–2104.
[102] C. Poon, *On the role of total variation in compressed sensing*, SIAM J. Imaging Sci., 8 (2015).
[103] J. Rasch and A. Chambolle, *Inexact first-order primal–dual algorithms*, Comput. Optim. Appl., 76 (2020), pp. 381–430.
[104] B. Recht, *A simpler approach to matrix completion*, J. Mach. Learn. Res., 12 (2011).
[105] B. Recht, M. Fazel, and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Rev., 52 (2010), pp. 471–501.
[106] J. D. Rennie and N. Srebro, *Fast maximum margin matrix factorization for collaborative prediction*, in Proceedings of ICML, 2005, pp. 713–719.

[107] C. A. Riofrio, D. Gross, S. T. Flammia, T. Monz, D. Nigg, R. Blatt, and J. Eisert, *Experimental quantum compressed sensing for a seven-qubit system*, Nature Comm., 8 (2017), pp. 1–8.
[108] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.
[109] V. Roulet, N. Boumal, and A. d'Aspremont, *Computational complexity versus statistical performance on sparse recovery problems*, Inf. Inference, 9 (2020), pp. 1–32.
[110] V. Roulet and A. d'Aspremont, *Sharpness, restart, and acceleration*, SIAM J. Optim., 30 (2020), pp. 262–289.
[111] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[112] C. Schwemmer, G. Tóth, A. Niggebaum, T. Moroder, D. Gross, O. Gühne, and H. Weinfurter, *Experimental comparison of efficient tomography schemes for a six-qubit state*, Phys. Rev. Lett., 113 (2014), 040503.
[113] I. W. Selesnick and M. A. Figueiredo, *Signal restoration with overcomplete wavelet transforms: Comparison of analysis and synthesis priors*, in Wavelets XIII, Proc. SPIE 7446, International Society for Optics and Photonics, 2009.
[114] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning*, Cambridge University Press, Cambridge, UK, 2014.
[115] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, *Phase retrieval with application to optical imaging: a contemporary overview*, IEEE Signal Process Mag., 32 (2015).
[116] A. M. Stuart, *Inverse problems: A Bayesian perspective*, Acta Numer., 19 (2010), pp. 451–559.
[117] K.-C. Toh and S. Yun, *An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems*, Pac. J. Optimi., 6 (2010), p. 15.
[118] C. Tomasi and T. Kanade, *Shape and motion from image streams under orthography: A factorization method*, Int. J. Comput. Vis., 9 (1992), pp. 137–154.
[119] Y. Traonmilin and R. Gribonval, *Stable recovery of low-dimensional cones in Hilbert spaces: One RIP to rule them all*, Appl. Comput. Harmon. Anal., 45 (2018), pp. 170–205.
[120] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
[121] M. Udell and A. Townsend, *Why are big data matrices approximately low rank?*, SIAM J. Math. Data Sci., 1 (2019), pp. 144–160.
[122] G. Wang, J. C. Ye, K. Mueller, and J. A. Fessler, *Image reconstruction is a new frontier of machine learning*, IEEE Trans. Med. Imaging, 37 (2018), pp. 1289–1296.

SUPPLEMENTARY MATERIALS: WARPd: A Linearly Convergent First-Order Primal-Dual Algorithm for Inverse Problems with Approximate Sharpness Conditions*

Matthew J. Colbrook†

SM1. Connections with previous work. Additional to this section, we provide connections with previous work that are specific to each of sections 3 to 6 throughout the paper. We do not cover here the vast literature on NN techniques, discussed in subsection 1.1.

First-order methods: There are numerous specialized algorithms for various instances of (1.2) and related problems [SM10, SM11, SM26, SM29, SM50], as well as general-purpose solvers [SM12]. A common approach is to apply some form of smoothing and use Nesterov's acceleration [SM42], which achieves an objective function suboptimality of δ in $\mathcal{O}(\delta^{-1/2})$ steps for the smoothed problem, in combination with techniques such as continuation for the smoothing parameter. Increasing the smoothing typically improves the numerical performance of underlying solvers but at the expense of accuracy, and balancing this precise trade-off is difficult [SM12]. We will not attempt to survey this vast area but point the reader to [SM22, SM9]. The complexity of first-order methods is usually controlled by smoothness assumptions on the objective function, such as Lipschitz continuity of its gradient. Additional assumptions on the objective function such as strong and uniform convexity provide, respectively, linear and faster polynomial rates of convergence [SM41]. For example, using variants of the classical strong convexity assumption, linear convergence results have been obtained for LASSO [SM5, SM52]. However, strong or uniform convexity are often too restrictive in many applications. For results on asymptotic linear convergence of standard methods (e.g., proximal gradient methods) for certain continuously differentiable (but non strongly convex) objective functions, see [SM35, SM39, SM51].

Łojasiewicz-type inequalities: Achieving linear convergence for restarted first-order methods typically requires a Łojasiewicz-type or "sharpness" inequality such as

$$\gamma d(\hat{x}, X^*)^\beta \leq f(\hat{x}) - f^*, \tag{SM1.1}$$

also known as a Hölderian error bound, with knowledge of γ and β [SM31, SM47, SM48].[1] Here f is the objective function (with optimal value f^*) and $d(\cdot, X^*)$ denotes the distance to the set of minimizers. For example, Nemirovskii and Nesterov [SM40] linked a "strict minimum" condition similar to (SM1.1) with faster convergence rates using restart schemes for smooth objective functions. Hölderian error bounds were first introduced by Hoffman [SM33] to study systems of linear inequalities, and extended to convex optimization in [SM46, SM37, SM8, SM18, SM17]. Łojasiewicz showed that (SM1.1) holds generically for real analytic and subanalytic functions [SM36], and Bolte, Daniilidis, and Lewis extended this result to

*Supplementary material for SIIMS MS#M145500.
https://doi.org/10.1137/21M1455000

†Centre Sciences des Données, École Normale Supérieure, Paris, France (m.colbrook@damtp.cam.ac.uk).

[1] See [SM45] for a restart technique where the sharpness parameters are not learned or assumed known.

nonsmooth subanalytic convex functions [SM14]. As noted in the main text, a key difference between (1.5) and (SM1.1), and hence also between the restart scheme of this paper and the above cited work is that we only assume *approximate* control of the distance via the objective function difference. This gives us greater generality and allows us to tackle the case of *noisy measurements*, as well as prove *robustness* of our results (e.g., when considering sparse recovery, we cover approximately sparse vectors).

For further use of Łojasiewicz-type inequalities for first-order methods (e.g., assessing asymptotic rates of convergence), see [SM16, SM15, SM7, SM30]. Further works on restart schemes include [SM28], which showed that generic restart schemes can offer linear convergence given a rough estimate of the behavior of the function around its minimizers, and [SM43], which developed a heuristic analysis for restarts based on ripples or bumps in the trace of the objective value.

The example of sparse recovery: The use of (1.5) is closely related to [SM13], who were one of the first to realize how key assumptions in compressed sensing – such as the *robust nullspace property* – help bound the error of the approximation to a minimizer (produced by an optimization algorithm) in terms of error bounds on the approximation to the objective function. For example, [SM47] achieves linear convergence, using the restarted NESTA algorithm [SM11], for exact recovery (noiseless) of real-valued sparse vectors if $A \in \mathbb{R}^{m \times N}$ satisfies the null space property of order s. Under this assumption, if x is s-sparse and $A\hat{x} = Ax$, then one has

$$\|\hat{x} - x\|_{l^2} \lesssim \|\hat{x}\|_{l^1} - \|x\|_{l^1}. \tag{SM1.2}$$

The restart scheme in [SM47] is based on a careful reduction in the smoothing parameter, chosen by analyzing a combination of the error bounds for NESTA and (SM1.2). Though our methods are entirely different (e.g., we do not rely on smoothing, and we must take into account the additional error term owing to the approximate sharpness), for the specific case of sparse recovery discussed in section 3, our results can be considered a generalization of [SM47] to allow measurement noise, approximate sparsity, and structured compressed sensing.

Finally, the author of the current paper developed a restart scheme similar to WARPd-SR (see subsection 2.4) based on the Square-Root LASSO decoder for the specific case of sparse recovery ($B = 0$ and $\mathcal{J}(x) = \|x\|_{l^1_w}$, see section 3) from Fourier and binary measurements in [SM24]. The outcome was stable and accurate NNs, where unrolled iterations led to Fast Iterative REstarted NETworks (FIRENETs). Theorem 1.3 continues in this direction and provides foundations for stable and accurate NNs for a much broader class of problems. It was also shown in [SM24] that there are fundamental computability barriers for solving l^1 minimization if certain conditions, such as (1.5), are not met (here, we mean computing a minimizing vector as opposed to vectors that nearly minimize the objective function).

Primal-dual algorithms: WARPd uses iterations of Chambolle and Pock's primal-dual algorithm [SM23, SM21] and a novel restart scheme. The primal-dual hybrid gradient (PDHG) algorithm is a popular method to solve saddle point problems [SM27, SM44, SM20]. The linear convergence of primal-dual methods under different conditions is widely studied. For example, see [SM25] for bilinear problems (with a focus on training GANs) and [SM49] for partially strongly convex functions. Recently, [SM6] developed an adaptive restart scheme for PDHG applied to linear programming and showed linear convergence.

SM2. Proof of Theorem 2.3. For $\eta > 0$, define

$$\widehat{G}_\eta(\hat{x}, x, b) := \underbrace{\mathcal{J}(\hat{x}) + \mathcal{F}(B\hat{x}) + \eta\|A\hat{x} - b\|_{\mathcal{Y}_1^*} - \mathcal{J}(x) - \mathcal{F}(Bx) - \eta\|Ax - b\|_{\mathcal{Y}_1^*}}_{\text{objective function difference with } \lambda = \eta^{-1}}.$$

Analogous to (2.3), we use $\widehat{\mathrm{PD}}_{\boldsymbol{\tau}}$ to denote the exact updates. The following theorem bounds the gap $\widehat{G}_{\hat{C}_2}$ for exact primal-dual updates with $\lambda = 1/\hat{C}_2$. A version of Proposition 2.2 for WARPd-SR also holds in the Hilbert space case, whose proof is almost identical and hence omitted.

Proposition SM2.1 (Bounds on $\widehat{G}_{\hat{C}_2}$ for primal-dual updates). *Suppose that the step sizes τ_1, τ_2 and τ_3 satisfy $\tau_1(\tau_2\|A\|^2 + \tau_3\|B\|^2) < 1$. Let $x_0 \in \mathcal{X}$, $y_1^{(0)} = 0 \in \mathcal{Y}_1$, $y_2^{(0)} = 0 \in \mathcal{Y}_2$, and*

$$(x^{(j+1)}, y_1^{(j+1)}, y_2^{(j+1)}) = \widehat{\mathrm{PD}}_{\boldsymbol{\tau}}(x^{(j)}, y_1^{(j)}, y_2^{(j)}), \quad j = 0, \dots, k-1.$$

Define the ergodic averages

$$X_k = \frac{1}{k}\sum_{j=1}^{k} x^{(j)}, \quad [Y_k]_1 = \frac{1}{k}\sum_{j=1}^{k} y_1^{(j)}, \quad [Y_k]_2 = \frac{1}{k}\sum_{j=1}^{k} y_2^{(j)}.$$

Then for any $x \in \mathcal{X}$,

$$\widehat{G}_{\hat{C}_2}(X_k, x, b)/\hat{C}_2 \le \frac{2}{k}\left(\frac{D_{\mathcal{X}}(x, x^{(0)})}{\tau_1} + \frac{g_1(1)}{\tau_2} + \frac{g_2(1/\hat{C}_2)}{\tau_3}\right), \tag{SM2.1}$$

where g_1 and g_2 satisfy (1.3) and (1.4), respectively.

Proof. Recall the definition of $\widehat{\mathcal{L}}$ in (2.21). Since $\tau_1(\tau_2\|A\|^2 + \tau_3\|B\|^2) < 1$, a simple adaptation of [SM23, Theorem 1, Remark 2] shows that for any $x \in \mathcal{X}$, $y_1 \in \mathcal{Y}_1$ and $y_2 \in \mathcal{Y}_2$,

$$\widehat{\mathcal{L}}(X_k, y_1, y_2) - \widehat{\mathcal{L}}(x, [Y_k]_1, [Y_k]_2) \le \frac{2}{k}\left(\frac{D_{\mathcal{X}}(x, x^{(0)})}{\tau_1} + \frac{D_{\mathcal{Y}_1}(y_1, 0)}{\tau_2} + \frac{D_{\mathcal{Y}_2}(y_2, 0)}{\tau_3}\right). \tag{SM2.2}$$

Let x be feasible and y_1 be any unit norm vector such that $\|AX_k - b\|_{\mathcal{Y}_1^*} = \langle AX_k - b, y_1\rangle_{\mathbb{R}}$. Writing out the difference on the left-hand side of (SM2.2), and recalling (1.3),

$$\begin{aligned}
&\mathcal{J}(X_k)/\hat{C}_2 - \langle Bx, [Y_k]_2\rangle_{\mathbb{R}} + \mathcal{F}^*(\hat{C}_2[Y_k]_2)/\hat{C}_2 \\
&+ \|AX_k - b\|_{\mathcal{Y}_1^*} - \mathcal{J}(x)/\hat{C}_2 + \langle BX_k, y_2\rangle_{\mathbb{R}} \\
&-\mathcal{F}^*(\hat{C}_2 y_2)/\hat{C}_2 - \langle Ax - b, [Y_k]_1\rangle_{\mathbb{R}} + \chi_{\mathcal{B}_{\mathcal{Y}_1}}([Y_k]_1) \le \frac{2}{k}\left(\frac{D_{\mathcal{X}}(x, x^{(0)})}{\tau_1} + \frac{g_1(1)}{\tau_2} + \frac{D_{\mathcal{Y}_2}(y_2, 0)}{\tau_3}\right).
\end{aligned} \tag{SM2.3}$$

The left-hand side must be finite. It follows that $\|[Y_k]_1\|_{\mathcal{Y}_1} \le 1$ and hence that $-\|Ax - b\|_{\mathcal{Y}_1^*} \le -\langle Ax - b, [Y_k]_1\rangle_{\mathbb{R}}$. We now take the supremum of the left-hand side of (SM2.3) over y_2 with $D_{\mathcal{Y}_2}(y_2, 0) \le g_2(1/\hat{C}_2)$ and recall that $\mathcal{F}$ satisfies (1.4). It follows from (SM2.3) that

(SM2.4)
$$\mathcal{J}(X_k)/\hat{C}_2 + \mathcal{F}(BX_k)/\hat{C}_2 + \|AX_k - b\|_{\mathcal{Y}_1^*} - \mathcal{J}(x)/\hat{C}_2 - \langle Bx, [Y_k]_2\rangle_{\mathbb{R}} + \mathcal{F}^*(\hat{C}_2[Y_k]_2) - \|Ax-b\|_{\mathcal{Y}_1^*} \leq \frac{2}{k}\left(\frac{D_{\mathcal{X}}(x,x^{(0)})}{\tau_1} + \frac{g_1(1)}{\tau_2} + \frac{g_2(1/\hat{C}_2)}{\tau_3}\right).$$

Since $-\mathcal{F}(Bx)/\hat{C}_2 \leq -\langle Bx, [Y_k]_2\rangle_{\mathbb{R}} + \mathcal{F}^*(\hat{C}_2[Y_k]_2)$, (SM2.4) yields (SM2.1). ∎

A careful study of the proof shows that $\|[Y_k]_1\|_{\mathcal{Y}_1} \leq 1$. In addition, if we are in the setting of Example 1.1 with $\mathcal{F}(Bx) = \|Bx\|_{l^1}$, then $\|[Y_k]_2\|_{l^\infty} \leq 1/\hat{C}_2$. This provides a way of bounding the dual variables for warm restarts. For brevity, we omit the details.

Proof of Theorem 2.3. Consider the setup in the statement of Theorem 2.3. Let $\psi_k = \texttt{InnerIt-SR}(x_0, \tau_1, \tau_2, \tau_3, k)$ denote the exact updates described in Algorithm 2.4 (or Algorithm 2.5) for $\lambda = 1/\hat{C}_2$. Suppose that the initial starting vector x_0 satisfies $\sqrt{2D_{\mathcal{X}}(\mathbb{x}, x_0)} \leq \hat{C}_1(\delta+\omega)$ for some $\omega > 0$. Combining this with (SM2.1), we have

(SM2.5)
$$\widehat{G}_{\hat{C}_2}(\psi_k, \mathbb{x}, b) \leq \frac{\hat{C}_2}{k}\left(\frac{\hat{C}_1^2(\delta+\omega)^2}{\tau_1} + \frac{2g_1(1)}{\tau_2} + \frac{2g_2(1/\hat{C}_2)}{\tau_3}\right).$$

Let $\tau \in (0,1)$, and suppose that $\tau_1(\tau_2 L_A^2 + \tau_3 L_B^2) = \tau^2$. In this case, the optimal choices of τ_1, τ_2 and τ_3 that minimize the right-hand side of (SM2.5) are

$$\tau_1(\omega) = \frac{\tau\hat{C}_1(\delta+\omega)}{L_A\sqrt{2g_1(1)} + L_B\sqrt{2g_2(1/\hat{C}_2)}}, \quad \tau_2(\omega) = \frac{\tau\sqrt{2g_1(1)}}{L_A\hat{C}_1(\delta+\omega)}, \quad \tau_3(\omega) = \frac{\tau\sqrt{2g_2(1/\hat{C}_2)}}{L_B\hat{C}_1(\delta+\omega)}$$

With this choice, we have that

(SM2.6)
$$\widehat{G}_{\hat{C}_2}(\psi_k, \mathbb{x}, b) \leq \frac{2\hat{C}_1\hat{C}_2}{k}\left(\frac{L_A\sqrt{2g_1(1)} + L_B\sqrt{2g_2(1/\hat{C}_2)}}{\tau}\right)(\delta+\omega).$$

For $\nu \in (0,1)$ that we optimize later, set $k = \lceil 2\hat{C}_1\hat{C}_2(L_A\sqrt{2g_1(1)} + L_B\sqrt{2g_2(1/\hat{C}_2)})/(\nu\tau)\rceil$ so that (SM2.6) implies that $\widehat{G}_{\hat{C}_2}(\psi_k, \mathbb{x}, b) \leq \nu(\delta+\omega)$.

We now describe the restart scheme. From (2.20) and the assumption $\mathcal{J}(\cdot) + \mathcal{F}(B\cdot) \geq 0$,

$$\widehat{G}_{\hat{C}_2}(0, \mathbb{x}, b) = \mathcal{J}(0)+\mathcal{F}(0)+\hat{C}_2\|b\|_{\mathcal{Y}_1^*} - \mathcal{J}(\mathbb{x}) - \mathcal{F}(B\mathbb{x}) - \hat{C}_2\|A\mathbb{x}-b\|_{\mathcal{Y}_1^*} \leq \mathcal{J}(0)+\mathcal{F}(0)+\hat{C}_2\|b\|_{\mathcal{Y}_1^*}.$$

It follows from (1.5) that $\sqrt{2D_{\mathcal{X}}(\mathbb{x},0)} \leq \hat{C}_1(\delta+\omega_0)$ with $\omega_0 = \mathcal{J}(0)+\mathcal{F}(0)+\hat{C}_2\|b\|_{\mathcal{Y}_1^*}$. Given $n \in \mathbb{N}$, for $j = 1, \ldots, n-1$ set $\omega_j = \nu(\delta+\omega_{j-1})$. By summing a geometric series, this implies that $\omega_n \leq \frac{\nu\delta}{1-\nu} + \nu^n[\mathcal{J}(0)+\mathcal{F}(0)+\hat{C}_2\|b\|_{\mathcal{Y}_1^*}]$. We define $\phi_n(b)$ iteratively via

$$\phi_1(b) = \psi_k(0, \tau_1(\omega_0), \tau_2(\omega_0), \tau_3(\omega_0)), \quad \phi_j(b) = \psi_k(\phi_{j-1}(b), \tau_1(\omega_{j-1}), \tau_2(\omega_{j-1}), \tau_3(\omega_{j-1})),$$

for $j = 1, \dots, n$. The choice of ω_j and the above argument inductively show that

$$\sqrt{2D_{\mathcal{X}}(\mathbb{x}, \phi_n(b))} \leq \hat{C}_1(\delta + \omega_n) \leq \hat{C}_1 \left(\delta + \frac{\nu\delta}{1-\nu} + \nu^n \left[\mathcal{J}(0) + \mathcal{F}(0) + \hat{C}_2 \|b\|_{\mathcal{Y}_1^*} \right] \right).$$

For $T = kn$ inner iterations, the error term ν^n is equal to $\exp(Tk^{-1}\log(\nu))$. If we ignore the ceiling function in the choice of k, the optimal choice of $\nu = e^{-1}$ is found via differentiation. This choice yields (1.7). ∎

SM3. Proofs of results and further details for section 3. We begin with the proof of results from section 3. The following two lemmas are taken from the compressed sensing literature [SM1].

Lemma SM3.1 (**rNSPL implies l^1_w distance bound**). *Suppose that A has the weighted rNSPL of order $(\mathbf{s}, \mathbf{M})$ with constants $0 < \rho < 1$ and $\gamma > 0$. Let $x, \hat{x} \in \mathbb{C}^N$, then*

$$\|\hat{x} - x\|_{l^1_w} \leq \frac{1+\rho}{1-\rho} \left(2\sigma_{\mathbf{s},\mathbf{M}}(x)_{l^1_w} + \|\hat{x}\|_{l^1_w} - \|x\|_{l^1_w} \right) + \frac{2\gamma}{1-\rho} \sqrt{\xi} \|A(\hat{x} - x)\|_{l^2}. \tag{SM3.1}$$

Lemma SM3.2 (**rNSPL implies l^2 distance bound**). *Suppose that A has the weighted rNSPL of order $(\mathbf{s}, \mathbf{M})$ with constants $0 < \rho < 1$ and $\gamma > 0$. Let $x, \hat{x} \in \mathbb{C}^N$, then*

$$\|\hat{x} - x\|_{l^2} \leq \left(\rho + \frac{(1+\rho)\kappa^{1/4}}{2} \right) \frac{\|\hat{x} - x\|_{l^1_w}}{\sqrt{\xi}} + \left(1 + \frac{\kappa^{1/4}}{2} \right) \gamma \|A(\hat{x} - x)\|_{l^2}. \tag{SM3.2}$$

Combining these two lemmas, we can prove Lemma SM3.3.

Lemma SM3.3. *Suppose that A has the weighted rNSPL of order $(\mathbf{s}, \mathbf{M})$ with constants $0 < \rho < 1$ and $\gamma > 0$. Then the assumption (1.5) holds with*

$$\begin{aligned} C_1 &= \left(\rho + \frac{(1+\rho)\kappa^{1/4}}{2} \right) \frac{1+\rho}{\sqrt{\xi}(1-\rho)}, \\ C_2 &= \frac{\left(1 + \frac{\kappa^{1/4}}{2}\right)\gamma + \left(\rho + \frac{(1+\rho)\kappa^{1/4}}{2}\right)\frac{2\gamma}{(1-\rho)}}{C_1} = \frac{\gamma}{C_1} \cdot \frac{2 + 2\rho + (3+\rho)\kappa^{1/4}}{2(1-\rho)}, \\ c(\mathbb{x}, b) &= 2\sigma_{\mathbf{s},\mathbf{M}}(\mathbb{x})_{l^1_w} + C_2 \left(\|A\mathbb{x} - b\|_{l^2} + \epsilon \right). \end{aligned}$$

Moreover,

$$\|\hat{x} - \mathbb{x}\|_{l^1_w} \leq \frac{1+\rho}{1-\rho} \left(G_{C_2}(\hat{x}, \mathbb{x}, b) + c(\mathbb{x}, b) \right). \tag{SM3.3}$$

Proof of Lemma SM3.3. We first substitute (SM3.1) into the right-hand side of (SM3.2) to obtain (for $x = \mathbb{x}$)

$$\|\hat{x} - \mathbb{x}\|_{l^2} \leq C_1 \left(\|\hat{x}\|_{l^1_w} - \|\mathbb{x}\|_{l^1_w} \right) + 2C_1 \sigma_{\mathbf{s},\mathbf{M}}(\mathbb{x})_{l^1_w} + C_1 C_2 \|A(\hat{x} - \mathbb{x})\|_{l^2}.$$

Using $\|A(\hat{x}-\mathbb{x})\|_{l^2} \leq \|A\hat{x}-b\|_{l^2} - \epsilon + \|A\mathbb{x}-b\|_{l^2} + \epsilon$, and rearranging, we arrive at (1.5) for the stated choice of C_1, C_2 and c. For the final part, note that

$$\frac{2\gamma}{1+\rho}\sqrt{\xi} = \frac{\left(\rho + \frac{(1+\rho)\kappa^{1/4}}{2}\right)\frac{2\gamma}{(1-\rho)}}{C_1} \leq C_2.$$

Combining this with (SM3.1), we see that

$$\|\hat{x}-\mathbb{x}\|_{l^1_w} \leq \frac{1+\rho}{1-\rho}\left(2\sigma_{\mathbf{s},\mathbf{M}}(\mathbb{x})_{l^1_w} + \|\hat{x}\|_{l^1_w} - \|\mathbb{x}\|_{l^1_w} + C_2\|A(\hat{x}-\mathbb{x})\|_{l^2}\right).$$

Again, using $\|A(\hat{x}-\mathbb{x})\|_{l^2} \leq \|A\hat{x}-b\|_{l^2} - \epsilon + \|A\mathbb{x}-b\|_{l^2} + \epsilon$, we arrive at (SM3.3). ∎

Proof of Theorem 3.3. The only result that does not follow directly from Theorem 1.2 and Lemma SM3.3 is the bound on $\|\phi_n(b)-\mathbb{x}\|_{l^1_w}$. However, the proof of Theorem 1.2 shows that

$$G_{C_2}\left(\phi_n(b),\mathbb{x},b\right) + c(\mathbb{x},b) \leq \frac{\delta}{1-\exp(-1)} + C_2\|b\|_{l^2}\cdot\exp\left(-T(n)\left\lceil 2eL_A\gamma\frac{2+2\rho+(3+\rho)\kappa^{1/4}}{2\tau(1-\rho)}\right\rceil^{-1}\right).$$

Combining this with (SM3.3) gives the required result. ∎

For completeness, we now describe the sampling setup for the example in subsection 3.2. We first recall the concept of multilevel random subsampling.

Definition SM3.4 (Multilevel random subsampling [SM4]). *Let $\mathbf{N} = (N_1,\dots,N_l) \in \mathbb{N}^l$, where $1 \leq N_1 < \cdots < N_l = N$ and $\mathbf{m} = (m_1,\dots,m_l) \in \mathbb{N}^l$ with $m_k \leq N_k - N_{k-1}$ for $k = 1,\dots,l$, and $N_0 = 0$. For each $k = 1,\dots,l$, let $\mathcal{I}_k = \{N_{k-1}+1,\dots,N_k\}$ if $m_k = N_k - N_{k-1}$ and if not, let $t_{k,1},\dots,t_{k,m_k}$ be chosen uniformly and independently from the set $\{N_{k-1}+1,\dots,N_k\}$ (with possible repeats), and set $\mathcal{I}_k = \{t_{k,1},\dots,t_{k,m_k}\}$. If $\mathcal{I} = \mathcal{I}_{\mathbf{N},\mathbf{m}} = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_l$ we refer to $\mathcal{I}$ as an $(\mathbf{N},\mathbf{m})$-multilevel subsampling scheme.*

Definition SM3.5 (Multilevel subsampled unitary matrix). *A matrix $A \in \mathbb{C}^{m\times N}$ is an $(\mathbf{N},\mathbf{m})$-multilevel subsampled unitary matrix if $A = P_{\mathcal{I}}DU$ for a unitary matrix $U \in \mathbb{C}^{N\times N}$ and $(\mathbf{N},\mathbf{m})$-multilevel subsampling scheme $\mathcal{I}$. Here, D is a diagonal scaling matrix with*

$$D_{ii} = \sqrt{\frac{N_k - N_{k-1}}{m_k}}, \quad i = N_{k-1}+1,\dots,N_k, \quad k = 1,\dots,l$$

and $P_{\mathcal{I}}$ denotes the projection onto the span of the basis vectors indexed by $\mathcal{I}$.

Let $Q = 2^r$ for $r \in \mathbb{N}$, and consider vectors on $\mathbb{C}^Q$ or d-dimensional tensors on $\mathbb{C}^{Q\times\cdots\times Q}$. To keep notation consistent with the main text, we set $N = Q^d$ so that the objective is to recover a vectorized $\mathbb{x} \in \mathbb{C}^N$.[2] Let $V \in \mathbb{C}^{N\times N}$ be either the matrix $F^{(d)}$ or $W^{(d)}$, corresponding

[2]The following can also be generalized to rectangles (i.e., $\mathbb{C}^{2^{r_1}\times\cdots\times 2^{r_d}}$ with possibly different $r_1,\dots,r_d$) or dimensions that are not powers of two.

to the d-dimensional discrete Fourier or Walsh–Hadamard transform. In the Fourier case, we divide the different frequencies $\{-Q/2+1,\dots,Q/2\}^d$ into dyadic bands. For $d=1$, we let $B_1=\{0,1\}$ and $B_k=\{-2^{k-1}+1,\dots,-2^{k-2}\}\cup\{2^{k-2}+1,\dots,2^{k-1}\}$ for $k=2,\dots,r$. In the binary case, we define the frequency bands $B_1=\{0,1\}$ and $B_k=\{2^{k-1},\dots,2^k-1\}$ for $k=2,\dots,r$ in the one-dimensional case. In the general d-dimensional case for Fourier or binary sampling, we set $B_{\mathbf{k}}^{(d)}=B_{k_1}\times\ldots\times B_{k_d}$ for $\mathbf{k}=(k_1,\dots,k_d)\in\mathbb{N}^d$. To recover a sparse representation, we consider the Haar wavelet coefficients for simplicity, though similar statements can be made for higher order Daubechies wavelets [SM3] and [SM38, Table 1]. We denote the discrete Haar Wavelet transform by $\Phi\in\mathbb{C}^{N\times N}$. We consider a multilevel subsampled unitary matrix (Definition SM3.5), with $U=V\Psi^*$. Given $\{m_{\mathbf{k}=(k_1,\dots,k_d)}\}_{k_1,\dots,k_d=1}^r$, we use multilevel random sampling with $m_{\mathbf{k}}$ measurements chosen from $B_{\mathbf{k}}^{(d)}$ according to Definition SM3.4. This corresponds to $l=r^d$ and the N_i's can be chosen given a suitable ordering of the Fourier/Walsh basis. The sparsity in levels structure (Definition 3.1) is chosen to correspond to the r wavelet levels. Finally, we define

$$\mathcal{M}_{\mathcal{F}}=\sum_{j=1}^{\|\mathbf{k}\|_{l^\infty}}s_j\prod_{i=1}^{d}2^{-|k_i-j|}+\sum_{j=\|\mathbf{k}\|_{l^\infty}+1}^{r}s_j2^{-2(j-\|\mathbf{k}\|_{l^\infty})}\prod_{i=1}^{d}2^{-|k_i-j|},\quad \mathcal{M}_{\mathcal{W}}=s_{\|\mathbf{k}\|_{l^\infty}}\prod_{i=1}^{d}2^{-|k_i-\|\mathbf{k}\|_{l^\infty}|}.$$

The following result was proven in [SM24].

Theorem SM3.6. *Consider the above setup of recovering a d-dimensional tensor $c\in\mathbb{C}^{Q^d}$ ($N=Q^d$) from subsampled Fourier or binary measurements Vc, such that A is a multilevel subsampled unitary matrix with respect to $U=V\Psi^*$. Let $\epsilon_{\mathbb{P}}\in(0,1)$ and $\mathcal{L}=d\cdot r^2\cdot\log(2m)\cdot\log^2\left(s\cdot\kappa(\mathbf{s},\mathbf{M},w)\right)+\log(\epsilon_{\mathbb{P}}^{-1})$. Suppose that:*

(a) *In the Fourier case, $m_{\mathbf{k}}\gtrsim\kappa(\mathbf{s},\mathbf{M},w)\cdot\mathcal{M}_{\mathcal{F}}(\mathbf{s},\mathbf{k})\cdot\mathcal{L}$.*

(b) *In the binary case, $m_{\mathbf{k}}\gtrsim\kappa(\mathbf{s},\mathbf{M},w)\cdot\mathcal{M}_{\mathcal{W}}(\mathbf{s},\mathbf{k})\cdot\mathcal{L}$.*

Then with probability at least $1-\epsilon_{\mathbb{P}}$, A satisfies the weighted rNSPL of order $(\mathbf{s},\mathbf{M})$ with constants $\rho=1/16$ and $\gamma=\sqrt{3/2}$.

The sampling conditions are optimized by minimizing $\kappa(\mathbf{s},\mathbf{M},w)$. Up to a constant scale, this corresponds to the choice $w_{(j)}=\sqrt{s/s_j}$. Up to log-factors, the measurement condition then becomes equivalent to the currently best-known oracle estimator (where one assumes apriori knowledge of the support of the vector) [SM2, Prop. 3.1]. Theorem SM3.6 gives us an immediate example of being able to apply Theorem 3.3, as is done in the main text.

SM4. Proofs of results in section 4 and section 5. We first consider the uniform recovery guarantees in section 4, for which we make use of the following theorem that generalizes Lemma SM3.3.

Theorem SM4.1 ([SM34, Theorem 3.2]). *Let $p\in[1,2]$ and suppose that $A:\mathbb{C}^{n_1\times n_2}\to\mathbb{C}^m$ satisfies the Frobenius-robust rank null space property of order r with constants $\rho\in(0,1)$ and $\gamma>0$. Then for any $M,\widehat{M}\in\mathbb{C}^{n_1\times n_2}$,*

(SM4.1)
$$\|\widehat{M}-M\|_p\le\frac{(1+\rho)^2}{(1-\rho)r^{\frac{p-1}{p}}}\left(2\|M_c\|_1+\|\widehat{M}\|_1-\|M\|_1\right)+\frac{\gamma(3+\rho)}{1-\rho}r^{\frac{1}{p}-\frac{1}{2}}\|A(\widehat{M}-M)\|_{l^2}.$$

In particular, taking the $p = 2$ case in (SM4.1) and using $\|A(\widehat{M} - M)\|_{l^2} \leq \|A(\widehat{M}) - b\|_{l^2} - \epsilon + \|A(M) - b\|_{l^2} + \epsilon$, we see that (1.5) is satisfied with (for $M = \mathbb{x}$)

$$C_1 = \frac{(1+\rho)^2}{(1-\rho)r^{\frac{1}{2}}}, \quad C_2 = \frac{\gamma(3+\rho)r^{\frac{1}{2}}}{(1+\rho)^2}, \quad c(\mathbb{x}, b) = 2\|\mathbb{x}_c\|_1 + \frac{\gamma(3+\rho)r^{\frac{1}{2}}}{(1+\rho)^2}\left(\|A(\mathbb{x}) - b\|_{l^2} + \epsilon\right).$$

We can now finish the proof of Theorem 4.2.

Proof of Theorem 4.2. The proof of Theorem 1.2 shows that

$$G_{C_2}\left(\phi_n(b), \mathbb{x}, b\right) + c(\mathbb{x}, b) \leq \frac{\delta}{1 - \exp(-1)} + C_2\|b\|_{l^2} \cdot \exp\left(-T(n)\left\lceil \frac{2eL_A\gamma}{\tau}\frac{(3+\rho)}{(1-\rho)}\right\rceil^{-1}\right),$$

where we have used $C_1C_2 = \frac{\gamma(3+\rho)}{1-\rho}$. The result now follows from (SM4.1) in Theorem SM4.1. ∎

We now turn to the non-uniform recovery guarantees in section 5 for matrix completion. We will need the following non-symmetric pinching lemma.

Lemma SM4.2 ([SM32]). *Let $P_1 \in \mathbb{C}^{n_1 \times n_1}$ and $P_2 \in \mathbb{C}^{n_2 \times n_2}$ be two orthogonal projection matrices. Then for any $M \in \mathbb{C}^{n_1 \times n_2}$,*

$$\|M\|_1 \geq \|P_1 M P_2\|_1 + \|P_1^{\perp} M P_2^{\perp}\|_1. \tag{SM4.2}$$

We are now ready to prove Theorem 5.3. We use the following convention for the Hilbert–Schmidt inner product:

$$\operatorname{tr}(M_2^* M_1) = \langle M_2, M_1 \rangle.$$

Proof of Theorem 5.3. Let $\widehat{M} \in \mathbb{C}^{n_1 \times n_2}$ and $\Delta = \widehat{M} - \mathbb{x}$. Using Lemma SM4.2 and $P^{\perp}\mathbb{x}Q^{\perp} = 0$,

$$\|\widehat{M}\|_1 \geq \|P\widehat{M}Q\|_1 + \|P^{\perp}\widehat{M}Q^{\perp}\|_1 = \|P\widehat{M}Q\|_1 + \|P^{\perp}\Delta Q^{\perp}\|_1.$$

Let $\Delta_T^{\perp} = P^{\perp}\Delta Q^{\perp}$ and $\Delta_T = \Delta - \Delta_T^{\perp}$. Using the fact that $P\widehat{M}Q = \mathbb{x} + P\Delta Q$, we have

$$\|\widehat{M}\|_1 \geq \|\mathbb{x} + P\Delta Q\|_1 + \|\Delta_T^{\perp}\|_1. \tag{SM4.3}$$

Since $\|UV^*\| \leq 1$ and $\langle UV^*, \mathbb{x}\rangle = \|\mathbb{x}\|_1$, we have that

$$\|\mathbb{x}\|_1 - |\langle UV^*, P\Delta Q\rangle| \leq |\langle UV^*, \mathbb{x} + P\Delta Q\rangle| \leq \|\mathbb{x} + P\Delta Q\|_1. \tag{SM4.4}$$

Writing out the inner product in terms of the trace, we see that

$$\langle UV^*, P\Delta Q\rangle = \operatorname{tr}\left(VU^*UU^*\Delta VV^*\right) = \operatorname{tr}\left(VU^*\Delta\right) = \langle UV^*, \Delta\rangle,$$

where we use the cyclic property of tr and $U^*U = V^*V = I_r$. We then use the decomposition

$$\langle UV^*, \Delta\rangle = \langle Y, \Delta\rangle + \langle UV^* - Y, \Delta\rangle. \tag{SM4.5}$$

Using the definition of the adjoint, the first inner product on the right-hand side of (SM4.5) is equal to $\langle z, A(\Delta)\rangle$. The second inner product can be written as

$$\langle UV^* - Y, \Delta\rangle = \langle UV^* - Y, \Delta_T\rangle - \langle P_{T_{\mathbb{x}}^\perp} Y, \Delta_T^\perp\rangle,$$

where we have used the fact that $\Delta_T^\perp = P_{T_{\mathbb{x}}^\perp}\Delta_T^\perp$ and $P_{T_{\mathbb{x}}^\perp} UV^* = 0$. Note also that

$$|\langle UV^* - Y, \Delta_T\rangle| = |\langle UV^* - P_{T_{\mathbb{x}}} Y, \Delta_T\rangle| \le \alpha_1 \|\Delta_T\|_2.$$

We then combine these arguments and use (5.2) to obtain

$$|\langle UV^*, P\Delta Q\rangle| \le \|z\|_{l^2}\|A(\Delta)\|_{l^2} + \alpha_1\|\Delta_T\|_2 + \alpha_2\|\Delta_T^\perp\|_2.$$

Combining this with (SM4.3) and (SM4.4) yields

$$\|\widehat{M}\|_1 \ge \|\mathbb{x}\|_1 + \|\Delta_T^\perp\|_1 - \|z\|_{l^2}\|A(\Delta)\|_{l^2} - \alpha_1\|\Delta_T\|_2 - \alpha_2\|\Delta_T^\perp\|_2.$$

Since $\|\cdot\|_2 \le \|\cdot\|_1$ and $\alpha_2 < 1$, we obtain the inequality

$$(1-\alpha_2)\|\Delta_T^\perp\|_2 - \alpha_1\|\Delta_T\|_2 \le \|\widehat{M}\|_1 - \|\mathbb{x}\|_1 + \|z\|_{l^2}\|A(\Delta)\|_{l^2}. \tag{SM4.6}$$

We now note that

$$\|A(\Delta_T)\|_{l^2} - \|A(\Delta_T^\perp)\|_{l^2} \le \|A(\Delta)\|_{l^2}.$$

Due to (5.3) and the fact that $\Delta_T \in T_{\mathbb{x}}$, it follows that

$$\gamma\|\Delta_T\|_2 - \|A\|\|\Delta_T^\perp\|_2 \le \|A(\Delta)\|_{l^2}. \tag{SM4.7}$$

Combining (SM4.6) and (SM4.7), we have

$$\|\Delta_T\|_2 + \|\Delta_T^\perp\|_2 \le \frac{\gamma + \|A\|}{(1-\alpha_2)\gamma - \alpha_1\|A\|}\left[\|\widehat{M}\|_1 - \|\mathbb{x}\|_1 + \left(\frac{\alpha_1 + 1 - \alpha_2}{\gamma + \|A\|} + \|z\|_{l^2}\right)\left\|A\left(\widehat{M} - \mathbb{x}\right)\right\|_{l^2}\right].$$

The inequality (5.4) now follows. Finally, using

$$\|A(\widehat{M} - \mathbb{x})\|_{l^2} \le \|A(\widehat{M}) - b\|_{l^2} - \epsilon + \epsilon + \|A(\mathbb{x}) - b\|_{l^2},$$

(1.5) is satisfied with the given values of C_1, C_2 and $c(\mathbb{x}, b)$. ∎

SM5. Proof of Theorem 6.2. We first need some results that follow from straightforward adaptations of the arguments laid out in [SM19, Section 2]. For completeness, we have provided the details of the necessary modifications. In what follows, we let $\mathbb{x}, \hat{x} \in \mathbb{C}^N$ and set $h = \mathbb{x} - \hat{x}$. Let T_0 denote the set of the largest s coefficients of $D^*\mathbb{x}$ in magnitude, and let D_T denote the matrix D restricted to the columns indexed by T. We divide the coordinates T_0^c into sets of size t (chosen later) in order of decreasing magnitude of $D^*_{T_0^c}h$. Call these sets $T_1, T_2, \ldots$ and set $T_{01} = T_0 \cup T_1$. We also collapse the notation $\delta_s(A, D)$ to δ_s.

First, an application of the triangle inequality yields

$$\|D^*\mathbb{x} - D^*h\|_{l^1} \le \|D^*\mathbb{x}\|_{l^1} + (\|D^*\hat{x}\|_{l^1} - \|D^*\mathbb{x}\|_{l^1}),$$

which implies that

$$\|D^*_{T_0^c}h\|_{l^1} \leq 2\|D^*_{T_0^c}\mathbb{x}\|_{l^1} + \|D^*_{T_0}h\|_{l^1} + (\|D^*\hat{x}\|_{l^1} - \|D^*\mathbb{x}\|_{l^1}). \tag{SM5.1}$$

The following lemma is a direct generalization of [SM19, Lemma 2.2], and we have omitted the proof since it simply makes use of (SM5.1) instead of [SM19, Lemma 2.1] (which assumes that $\hat{x}$ solves (6.1) so that the bracketed term on the right-hand side of (SM5.1) can be dropped).

Lemma SM5.1. *Setting $\rho = s/t$ and $\eta = 2\|D^*_{T_0^c}\mathbb{x}\|_{l^1}/\sqrt{s}$, we have*

$$\sum_{j\geq 2} \|D^*_{T_j}h\|_{l^2} \leq \sqrt{\rho}(\|D^*_{T_0}h\|_{l^2} + \eta) + \frac{1}{\sqrt{t}}(\|D^*\hat{x}\|_{l^1} - \|D^*\mathbb{x}\|_{l^1}). \tag{SM5.2}$$

The next result we tweak is [SM19, Lemma 2.4], where the following is proven via the same string of inequalities, but with Lemma SM5.1 replacing [SM19, Lemma 2.2].

Lemma SM5.2. *As a consequence of D-RIP, the following holds:*

$$\sqrt{1-\delta_{s+t}}\|D_{T_{01}}D^*_{T_{01}}h\|_{l^2} - \sqrt{\rho(1+\delta_t)}(\|h\|_{l^2} + \eta) \leq \|Ah\|_{l^2} + \frac{\sqrt{1+\delta_t}}{\sqrt{t}}(\|D^*\hat{x}\|_{l^1} - \|D^*\mathbb{x}\|_{l^1}). \tag{SM5.3}$$

Similarly, we obtain the following by adapting the proof of [SM19, Lemma 2.5].

Lemma SM5.3. *The vector h satisfies*

$$\|h\|^2_{l^2} \leq \|h\|_{l^2}\|D_{T_{01}}D^*_{T_{01}}h\|_{l^2} + \left[\sqrt{\rho}(\|D^*_{T_0}h\|_{l^2} + \eta) + \frac{1}{\sqrt{t}}(\|D^*\hat{x}\|_{l^1} - \|D^*\mathbb{x}\|_{l^1})\right]^2. \tag{SM5.4}$$

Using these results, we now depart from the argument in [SM19] and prove Theorem 6.2.

Proof of Theorem 6.2. To simplify the notation, we set

$$E_1 = \eta + \frac{1}{\sqrt{\rho t}}(\|D^*\hat{x}\|_{l^1} - \|D^*\mathbb{x}\|_{l^1}).$$

If $h = 0$, then there is nothing to prove, so we assume that $\|h\|_{l^2} > 0$. The inequality (SM5.4) together with $\|D^*_{T_0}h\|_{l^2} \leq \|h\|_{l^2}$ then implies that

$$\|h\|_{l^2} \leq \|D_{T_{01}}D^*_{T_{01}}h\|_{l^2} + \rho\|h\|_{l^2} + 2\rho E_1 + \rho\frac{E_1^2}{\|h\|_{l^2}}. \tag{SM5.5}$$

If $\|h\|_{l^2} \geq (\rho + \sqrt{\rho^2+\rho})E_1$, then we must have

$$2\rho E_1 + \rho\frac{E_1^2}{\|h\|_{l^2}} \leq (\rho + \sqrt{\rho^2+\rho})E_1.$$

Combining with (SM5.5), it follows, even in the case $\|h\|_{l^2} < (\rho + \sqrt{\rho^2+\rho})E_1$, that

$$\|h\|_{l^2} \leq \|D_{T_{01}}D^*_{T_{01}}h\|_{l^2} + \rho\|h\|_{l^2} + (\rho + \sqrt{\rho^2+\rho})E_1. \tag{SM5.6}$$

Combining Lemma SM5.2 and (SM5.6),

$$\|h\|_{l^2} \leq \left(\frac{\sqrt{\rho(1+\delta_t)}}{\sqrt{1-\delta_{s+t}}} + \rho \right) \|h\|_{l^2} + \frac{\|Ah\|_{l^2}}{\sqrt{1-\delta_{s+t}}} + \left(\rho + \sqrt{\rho^2+\rho} + \frac{\sqrt{\rho(1+\delta_t)}}{\sqrt{1-\delta_{s+t}}} \right) E_1. \tag{SM5.7}$$

We then use $\|Ah\|_{l^2} \leq \|A\hat{x} - b\|_{l^2} - \epsilon + \|A\mathbb{x} - b\|_{l^2} + \epsilon$. Rearranging (SM5.7) gives (1.5) with the parameters in (6.3). ∎

REFERENCES

[1] B. Adcock, V. Antun, and A. C. Hansen, *Uniform recovery in infinite-dimensional compressed sensing and applications to structured binary sampling*, arXiv:1905.00126, (2019).

[2] B. Adcock, C. Boyer, and S. Brugiapaglia, *On oracle-type local recovery guarantees in compressed sensing*, Inf. Inference, (2018).

[3] B. Adcock and A. Hansen, *Compressive Imaging: Structure, Sampling, Learning*, CUP, 2021.

[4] B. Adcock, A. C. Hansen, C. Poon, and B. Roman, *Breaking the coherence barrier: A new theory for compressed sensing*, in Forum Math. Sigma, vol. 5, CUP, 2017.

[5] A. Agarwal, S. Negahban, and M. J. Wainwright, *Fast global convergence of gradient methods for high-dimensional statistical recovery*, Ann. Statist., (2012), pp. 2452–2482.

[6] D. Applegate, O. Hinder, H. Lu, and M. Lubin, *Faster first-order primal-dual methods for linear programming using restarts and sharpness*, arXiv:2105.12715, (2021).

[7] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, *Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-łojasiewicz inequality*, Math. Oper. Res., 35 (2010), pp. 438–457.

[8] A. Auslender and J.-P. Crouzeix, *Global regularity theorems*, Math. Oper. Res., 13 (1988), pp. 243–253.

[9] A. Beck, *First-Order Methods in Optimization*, SIAM, 2017.

[10] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[11] S. Becker, J. Bobin, and E. J. Candès, *NESTA: A fast and accurate first-order method for sparse recovery*, SIAM J. Imaging Sci., 4 (2011), pp. 1–39.

[12] S. R. Becker, E. J. Candès, and M. C. Grant, *Templates for convex cone problems with applications to sparse signal recovery*, Math. Program. Comput., 3 (2011), p. 165.

[13] A. Ben-Tal and A. Nemirovski, *Lectures on modern convex optimization*, (2020/2021), https://www2.isye.gatech.edu/~nemirovs/.

[14] J. Bolte, A. Daniilidis, and A. Lewis, *The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems*, SIAM J. Optim., 17 (2007), pp. 1205–1223.

[15] J. Bolte, T. P. Nguyen, J. Peypouquet, and B. W. Suter, *From error bounds to the complexity of first-order descent methods for convex functions*, Math. Program., 165 (2017), pp. 471–507.

[16] J. Bolte, S. Sabach, and M. Teboulle, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Math. Program., 146 (2014), pp. 459–494.

[17] J. Burke and S. Deng, *Weak sharp minima revisited Part I: basic theory*, Control Cybernet., 31 (2002), pp. 439–469.

[18] J. V. Burke and M. C. Ferris, *Weak sharp minima in mathematical programming*, SIAM J. Control Optim., 31 (1993), pp. 1340–1359.

[19] E. J. Candes, Y. C. Eldar, D. Needell, and P. Randall, *Compressed sensing with coherent and redundant dictionaries*, Appl. Comput. Harmon. Anal., 31 (2011), pp. 59–73.

[20] A. Chambolle, M. J. Ehrhardt, P. Richtárik, and C. Schonlieb, *Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications*, SIAM J. Optim., 28 (2018).

[21] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.

[22] A. Chambolle and T. Pock, *An introduction to continuous optimization for imaging*, Acta Numer., 25 (2016), pp. 161–319.

[23] A. Chambolle and T. Pock, *On the ergodic convergence rates of a first-order primal–dual algorithm*, Math. Program., 159 (2016), pp. 253–287.

[24] M. J. Colbrook, V. Antun, and A. C. Hansen, *The difficulty of computing stable and accurate neural networks: On the barriers of deep learning and smale's 18th problem*, Proc. Natl. Acad., (to appear).

[25] C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng, *Training GANs with optimism*, arXiv:1711.00141, (2017).

[26] I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Comm. Pure Appl. Math., 57 (2004), pp. 1413–1457.

[27] E. Esser, X. Zhang, and T. F. Chan, *A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science*, SIAM J. Imaging Sci., 3 (2010).

[28] O. Fercoq and Z. Qu, *Restarting accelerated gradient methods with a rough strong convexity estimate*, arXiv:1609.07358, (2016).

[29] M. A. Figueiredo, R. D. Nowak, and S. J. Wright, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE J Sel Top Signal Process, 1 (2007), pp. 586–597.

[30] P. Frankel, G. Garrigos, and J. Peypouquet, *Splitting methods with variable metric for Kurdyka–łojasiewicz functions and general convergence rates*, J. Optim. Theory Appl., 165 (2015), pp. 874–900.

[31] R. M. Freund and H. Lu, *New computational guarantees for solving convex optimization problems with first order methods, via a function growth condition measure*, Math. Program., 170 (2018), pp. 445–477.

[32] T. Fuchs, D. Gross, P. Jung, F. Krahmer, R. Kueng, and D. Stöger, *Proof methods for robust low-rank matrix recovery*, arXiv:2106.04382, (2021).

[33] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Research Nat. Bur. Standards, 49 (1952).

[34] M. Kabanava, R. Kueng, H. Rauhut, and U. Terstiege, *Stable low-rank matrix recovery via null space properties*, Inf. Inference, 5 (2016), pp. 405–441.

[35] J. Liang, J. M. Fadili, and G. Peyré, *Local linear convergence of forward–backward under partial smoothness*, in NIPS, 2014.

[36] S. Lojasiewicz, *Une propriété topologique des sous-ensembles analytiques réels*, Les équations aux dérivées partielles, 117 (1963), pp. 87–89.

[37] O. L. Mangasarian, *A condition number for differentiable convex inequalities*, Math. Oper. Res., 10 (1985), pp. 175–179.

[38] A. Moshtaghpour, J. M. Bioucas-Dias, and L. Jacques, *Close encounters of the binary kind: Signal reconstruction guarantees for compressive Hadamard sampling with Haar wavelet basis*, IEEE Trans. Inform. Theory, 66 (2020), pp. 7253–7273.

[39] I. Necoara, Y. Nesterov, and F. Glineur, *Linear convergence of first order methods for non-strongly convex optimization*, Math. Program., 175 (2019), pp. 69–107.

[40] A. S. Nemirovskii and Y. E. Nesterov, *Optimal methods of smooth convex minimization*, USSR Comput. Math. Math. Phys., 25 (1985), pp. 21–30.

[41] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science & Business Media, 2003.

[42] Y. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program., 103 (2005), pp. 127–152.

[43] B. O'donoghue and E. Candes, *Adaptive restart for accelerated gradient schemes*, Found. Comput. Math., 15 (2015), pp. 715–732.

[44] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, *An algorithm for minimizing the Mumford–Shah functional*, in IEEE Int Conf Comput Vis, IEEE, 2009, pp. 1133–1140.

[45] J. Renegar and B. Grimmer, *A simple nearly optimal restart scheme for speeding up first-order methods*, Foundations of Computational Mathematics, (2021), pp. 1–46.

[46] S. M. Robinson, *An application of error bounds for convex programming in a linear space*, SIAM J. Control, 13 (1975), pp. 271–273.

[47] V. Roulet, N. Boumal, and A. d'Aspremont, *Computational complexity versus statistical performance on sparse recovery problems*, Inf. Inference, 9 (2020), pp. 1–32.

[48] V. Roulet and A. d'Aspremont, *Sharpness, restart, and acceleration*, SIAM J. Optim., 30 (2020), pp. 262–289.

[49] T. Valkonen and T. Pock, *Acceleration of the PDHGM on partially strongly convex functions*, J. Math. Imaging Vision, 59 (2017), pp. 394–414.

[50] E. Van Den Berg and M. P. Friedlander, *Probing the Pareto frontier for basis pursuit solutions*, SIAM J. Sci. Comput., 31 (2009), pp. 890–912.

[51] Z. Zhou and A. M.-C. So, *A unified approach to error bounds for structured convex optimization problems*, Math. Program., 165 (2017), pp. 689–728.

[52] Z. Zhou, Q. Zhang, and A. M.-C. So, ℓ_1*, p-norm regularization: Error bounds and convergence rate analysis of first-order methods*, in ICML, PMLR, 2015, pp. 1501–1510.