Semi-Random Sparse Recovery in Nearly-Linear Time

author names withheld

Under Review for OPT 2022

Abstract

Sparse recovery is one of the most fundamental and well-studied inverse problems. Standard statistical formulations of the problem are provably solved by general convex programming techniques and more practical, fast (nearly-linear time) iterative methods. However, these latter "fast algorithms" have previously been observed to be brittle in various real-world settings.

We investigate the brittleness of fast sparse recovery algorithms to generative model changes through the lens of studying their robustness to a "helpful" semi-random adversary, a framework which tests whether an algorithm overfits to input assumptions. We consider the following basic model: let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be a measurement matrix which contains an *unknown* subset of rows $\mathbf{G} \in \mathbb{R}^{m \times d}$ which are bounded and satisfy the restricted isometry property (RIP), but is otherwise arbitrary. Letting $x^\star \in \mathbb{R}^d$ be s-sparse, and given either exact measurements $b = \mathbf{A}x^\star$ or noisy measurements $b = \mathbf{A}x^\star + \xi$, we design algorithms recovering $x^\star$ information-theoretically optimally in nearly-linear time. We extend our algorithm to hold for weaker generative models relaxing our planted RIP row subset assumption to a natural weighted variant, and show that our method's guarantees naturally interpolate the quality of the measurement matrix to, in some parameter regimes, run in sublinear time.

Our approach differs from that of prior fast iterative methods with provable guarantees under semi-random generative models [23, 54], which typically separate the problem of learning the planted instance from the estimation problem, i.e. they attempt to first learn the planted "good" instance (in our case, $\mathbf{G}$). However, natural conditions which make sparse recovery tractable, such as RIP, are NP-hard to verify and hence first learning a sufficient row reweighting appears challenging. We eschew this approach and design a new iterative method, tailored to the geometry of sparse recovery, which is provably robust to our semi-random model. We hope our approach opens the door to new robust, efficient algorithms for other natural statistical inverse problems.

1. Introduction

Sparse recovery. Sparse recovery is one of the most fundamental and well-studied inverse problems, with numerous applications in prevalent real-world settings [39]. In its most basic form, we are given an entrywise Gaussian measurement matrix $\mathbf{G} \in \mathbb{R}^{m \times d}$ and measurements $b = \mathbf{G}x^\star$ for an unknown s-sparse $x^\star \in \mathbb{R}^d$; the goal of the problem is to recover $x^\star$. Seminal works by Candès, Romberg, and Tao [17–19] showed that even when the linear system in $\mathbf{G}$ is extremely underconstrained, recovery is tractable so long as $m = \Omega(s \log d)$. Further they gave a polynomial-time algorithm known as *basis pursuit* based on linear programming recovering $x^\star$ in this regime.

Unfortunately, the runtime of linear programming solvers, while polynomial in the size of the input, can still be prohibitive in many high-dimensional real-world settings. Correspondingly, a number of alternative approaches which may broadly be considered first-order methods have been

© .

developed. These methods provably achieve similar recovery guarantees under standard generative models such as Gaussian measurements, with improved runtimes compared to the aforementioned convex programming methods. We refer to these first-order methods through as "fast" algorithms throughout and they may roughly be placed in the following (potentially non-disjoint) categories.

- **Greedy algorithms**, e.g. [61, 67, 69], seek to greedily find elements in the support of the true $x^\star$ using different combinatorial search criteria.
- **Non-convex iterative algorithms**, e.g. [15, 16, 44, 60, 66], directly optimize a (potentially non-convex) objective over a non-convex domain.
- **Convex first-order methods**, e.g. [2, 8, 9, 25, 27, 43, 68] quickly solve the convex objective underlying basis pursuit using first-order methods.

We note that theoretically, recent advances by [80, 81], also obtain fast runtimes for the relevant linear program. The fastest IPM for the noiseless sparse recovery objective runs in time[1] $\widetilde{O}(nd + n^{2.5})$ which is nearly-linear when $\mathbf{A}$ is dense and $n \ll d^{2/3}$. For a range of (superlogarithmic, but sublinear) n, these runtimes are no longer nearly-linear; further, these IPMs are second-order and our focus is on designing first-order methods, which are potentially more practical.[2]

It has often been observed empirically that fast first-order methods can have large error, or fail to converge, in real-world settings [29, 48, 70] where convex programming-based algorithms (while potentially computationally cumbersome) perform well statistically [3, 84]. This may be surprising, given that in theory, fast algorithms essentially match the statistical performance of the convex programming-based algorithms under standard generative assumptions. While there have been many proposed explanations for this behavior, one compelling argument is that fast iterative methods used in practice are more brittle to changes in modeling assumptions. We adopt this viewpoint in this paper, and develop fast sparse recovery algorithms which achieve optimal statistical rates under a *semi-random adversarial model* [14, 41], a popular framework for investigating the robustness of learning algorithms under changes to the data distribution.

Semi-random adversaries. Semi-random adversaries are a framework for reasoning about algorithmic robustness to distributional shift. They are defined in statistical settings, and one common type of semi-random adversary is one which corresponds to generative models where data has been corrupted in a "helpful" or "monotone" way. Such a monotone semi-random adversary takes a dataset from which learning is information-theoretically tractable, and augments it with additional information; this additional information may not break the problem more challenging from an information-theoretic perspective,[3] but may affect the performance of algorithms in other ways. In this paper, we consider a semi-random adversary which makes the *computational problem* more difficult without affecting the problem information-theoretically, by returning a consistent superset of the unaugmented observations. This contrasts with other adversarial models such as gross corruption [4, 47, 77, 78], where corruptions may be arbitrary, and the corrupted measurements incorrect. It may be surprising that a "helpful" adversary has any implications whatsoever on a learning problem, from either an information-theoretic or computational standpoint.

1. We use $\widetilde{O}$ to hide polylogarithmic factors in problem parameters for brevity of exposition throughout the paper.
2. We also note that these IPM results do not immediately apply to natural (nonlinear) convex programs for sparse recovery under noisy observations, see Section E.
3. There are notable exceptions, e.g. the semi-random stochastic block model of [65].

Typically, convex programming methods for statistical recovery problems are robust to these sorts of perturbations — in brief, this is because constraints to a convex program that are met by an optimum point does not change the optimality of that point. However, greedy and non-convex methods — such as popular practical algorithms for sparse linear regression — can be susceptible to semi-random adversaries. Variants of this phenomenon have been reported in many common statistical estimation problems, such as stochastic block models and broadcast tree models [65], PAC learning [13], matrix completion [23, 64], and principal component regression [11]. This can be quite troubling, as semi-random noise can be thought of as a relatively mild form of generative model misspecification: in practice, the true distribution is almost always different from the models considered in theory. Consequently, an algorithm's non-robustness to semi-random noise is suggestive that the algorithm may be more unreliable in real-world settings.

We consider a natural semi-random adversarial model for sparse recovery (see e.g. page 284 of [5]), which extends the standard restricted isometry property (RIP) assumption, which states that applying matrix $\mathbf{A}$ approximately preserves the ℓ_2 norm of sparse vectors. Concretely, throughout the paper we say matrix $\mathbf{A}$ satisfies the (s, c)*-restricted isometry (RIP) property* if for all s-sparse vectors v, $\frac{1}{c}\|v\|_2^2 \leq \|\mathbf{A}v\|_2^2 \leq c\|v\|_2^2$. We state a basic version of our adversarial model here, and defer the full statement to Definition 4.[4] We defer stating the notation used in the paper to Section A.

Definition 1 (pRIP matrix) *Let $m, n, d \in \mathbb{N}$ be known with $n \geq m$. We say $\mathbf{A} \in \mathbb{R}^{n\times d}$ is ρ-pRIP (planted RIP) if there is an (unknown) $\mathbf{G} \in \mathbb{R}^{m\times d}$ such that each row of $\mathbf{G}$ is also a row of $\mathbf{A}$ and $\frac{1}{\sqrt{m}}\mathbf{G}$ is $(\Theta(s), \Theta(1))$-RIP for appropriate constants, and $\|\mathbf{G}\|_{\max} \leq \rho$. When $\rho = \widetilde{O}(1)$ for brevity we say $\mathbf{A}$ is pRIP.*

Under the problem parameterizations used in this paper, standard RIP matrix constructions satisfy $\rho = \widetilde{O}(1)$ with high probability. For example, when $\mathbf{G}$ is entrywise Gaussian and $m = \Theta(s \log d)$, with high probability setting $\rho = O(\sqrt{\log d})$ is compatible with Definition 1.

pRIP matrices can naturally be thought of as arising from a semi-random adversarial model as follows. First, an RIP matrix $\mathbf{G} \in \mathbb{R}^{m\times d}$ is generated, for example from a standard ensemble (e.g. Gaussian or subsampled Hadamard). An adversary inspects $\mathbf{G}$, and forms $\mathbf{A} \in \mathbb{R}^{n\times d}$ by reshuffling and arbitrarily augmenting rows of $\mathbf{G}$. Whenever we refer to a "semi-random adversary" in the remainder of the introduction, we mean the adversary provides us a pRIP measurement matrix $\mathbf{A}$.

The key problem we consider is recovering an unknown s-sparse vector $x^\star \in \mathbb{R}^d$ given measurements $b \in \mathbb{R}^n$ through $\mathbf{A}$. We consider both the *noiseless* or *exact* setting where $b = \mathbf{A}x^\star$ and the *noisy* setting where $b = \mathbf{A}x^\star + \xi$ for bounded ξ. In the noiseless setting, the semi-random adversary hence only gives the algorithm additional *consistent* measurements of the unknown s-sparse vector $x^\star$. In this sense, the adversary is "helpful," as it returns a superset of information which is sufficient for sparse recovery (formally, this adversary cannot break the standard restricted nullspace condition which underlies the success of convex programming). We note n may be much larger than m, i.e. we impose no constraint on how many measurements the adversary adds.

4. When clear from context, as it will be throughout the main sections of the paper, s will always refer to the sparsity of a vector $x^\star \in \mathbb{R}^d$ in an exact or noisy recovery problem through $\mathbf{A} \in \mathbb{R}^{n\times d}$. For example, the parameter s in Definition 1 is the sparsity of the vector in an associated sparse recovery problem.

2. Our results

We devise algorithms which match the nearly-linear runtimes and optimal recovery guarantees of faster algorithms on fully random data, but which retain both their runtime and the robust statistical performances of convex programming methods against semi-random adversaries. In this sense, our algorithms obtain the "best of both worlds." We compare more extensively to existing algorithms under Definition 1 in the following section. We first state our result under noiseless observations.

Theorem 2 (informal, see Theorem 5) *Let $x^\star \in \mathbb{R}^d$ be an unknown s-sparse vector. Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be pRIP. There is an algorithm, which given $\mathbf{A}$ and $b = \mathbf{A}x^\star$, runs in time $\widetilde{O}(nd)$, and outputs $x^\star$ with high probability.*

Since our problem size is nd, Theorem 2's runtime is nearly-linear in the input. We extend our algorithm to handle noisy observations, namely perturbed linear measurements from a pRIP matrix.

Theorem 3 (informal, see Theorem 16) *Let $x^\star \in \mathbb{R}^d$ be an unknown s-sparse vector, and let $\xi \in \mathbb{R}^n$ be arbitrary. Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be pRIP. There is an algorithm, which given $\mathbf{A}$ and $b = \mathbf{A}x^\star + \xi$, runs in time $\widetilde{O}(nd)$, and with high probability outputs x satisfying $\|x - x^\star\|_2 \le O(\frac{1}{\sqrt{m}}\left\|\xi_{(m)}\right\|_2)$, where $\xi_{(m)}$ denotes the largest m entries of ξ by absolute value, with other coordinates set to 0.*

The error scaling of Theorem 3 is optimal in the semi-random setting. Indeed, when there is no semi-random noise, the guarantees of Theorem 3 exactly match the standard statistical guarantees in the fully-random setting for sparse recovery, up to constants; for example, when $\mathbf{A} = \sqrt{m}\mathbf{I}$ (which is clearly RIP, in fact an exact isometry, after rescaling), it is information-theoretically impossible to obtain a better ℓ_2 error.[5] The error bound of Theorem 3 is similarly optimal in the semi-random setting because in the worst case, the largest entries of ξ may correspond to the rows of the RIP matrix from which recovery is information-theoretically possible.

Performance of existing algorithms. To contextualize Theorems 2 and 3, we discuss the performance of existing algorithms for sparse recovery under the semi-random adversarial model of Definition 1. First, it can be easily verified that our semi-random adversary never changes the information-theoretic tractability of sparse recovery. In the noiseless setting for example, the performance of the minimizer to the classical convex program based on ℓ_1 minimization, $\min_{\mathbf{A}x=b}\|x\|_1$, is unchanged in the presence of pRIP matrices (as $x^\star$ is still consistent with the constraint set, and in particular a RIP constraint set), and hence the semi-random problem can be solved in polynomial time via convex programming. This suggests the main question we address: can we design a near-linear time algorithm obtaining optimal statistical guarantees under pRIP measurements?

As alluded to previously, standard greedy and non-convex methods we have discussed may fail to converge to the true solution against appropriate semi-random adversaries. We give explicit counterexamples to several popular methods such as orthogonal matching pursuit and iterative hard thresholding in Section E. Further, it seems likely that similar counterexamples also break other, more complex methods commonly used in practice, e.g. matching pursuit [61] and CoSaMP [66].

Additionally, while fast "convex" iterative algorithms (e.g. first-order methods for solving objectives underlying polynomial-time convex programming approaches) will never fail to converge to the correct solution given pRIP measurements, the analyses which yield fast runtimes for these

5. In the literature it is often standard to scale down the sensing matrix $\mathbf{A}$ by $\sqrt{m}$; this is why our error bound is similarly scaled. However, this scaling is more convenient for our analysis, especially when stating weighted results.

algorithms [2, 68] rely on properties such as restricted smoothness and strong convexity (a specialization of standard conditioning assumptions to numerically sparse vectors). These hold under standard generative models but again can be broken by pRIP measurements; consequently, standard convergence analyses of "convex" first-order methods may yield arbitrarily poor rates.

One intuitive explanation for why faster methods fail is that they depend on conditions such as incoherence [38] or RIP [18], which can be destroyed by a semi-random adversary. For instance, RIP states that if S is any subset of $m = \Theta(s)$ columns of $\mathbf{A}$, and $\mathbf{A}_S$ is the submatrix formed by taking those columns of $\mathbf{A}$, then $\mathbf{A}_S^\top \mathbf{A}_S$ is an approximate isometry (i.e. it is well-conditioned). While it is well-known that RIP is satisfied with high probability when $\mathbf{A}$ consists of $\Theta(s \log d)$ Gaussian rows, it is straightforward to see that augmenting $\mathbf{A}$ with additional rows can ruin the condition number of submatrices of this form. In contrast, convex methods work under weaker assumptions such as the restricted nullspace property (RNP), which cannot be destroyed by the augmentation used by pRIP matrices. Though these weaker conditions (e.g. RNP) suffice for algorithms based on convex programming, known analyses of near-linear time "fast" algorithms require additional instance structure, such as incoherence or RIP. Thus, it is plausible that fast algorithms for sparse recovery are less robust to the sorts of distributional changes that may occur in practice.

Beyond submatrices. Our methods naturally extend to a more general setting (see Definition 4, wherein we define "weighted RIP" (wRIP) matrices, a generalization of Definition 1). Rather than assuming there is a RIP submatrix $\mathbf{G}$, we only assume that there is a (nonnegative) reweighting of the rows of $\mathbf{A}$ so that the reweighted matrix is "nice," i.e. it satisfies RIP. Definition 1 corresponds to the special case of this assumption where the weights are constrained to be either 0 or 1 (and hence indicates a row subset). In our technical sections (Sections C and D), our results are stated for this more general semi-random model, i.e. sparse recovery from wRIP measurements.

Towards instance-optimal guarantees. While the performance of the algorithms in Theorems 2 and 3 is already nearly-optimal in the worst case semi-random setting, one can still hope to improve our runtime and error bounds in certain scenarios. Our formal results, Theorems 5 and 16, provide these types of fine-grained instance-optimal guarantees in several senses.

In the noiseless setting (Theorem 5), if it happens to be that the entire matrix $\mathbf{A}$ is RIP (and not just $\mathbf{G}$), then standard techniques based on subsampling the matrix can be used to solve the problem in time $\widetilde{O}(sd)$ with high probability. Theorem 5's runtime smoothly interpolates between the two regimes of a worst-case adversary and an adversary which gives us additional random measurements from an RIP ensemble. Roughly speaking, if there exists a (a priori unknown) submatrix of $\mathbf{A}$ of $m \gg \widetilde{\Theta}(s)$ rows which is RIP, then we show that our algorithm runs in *sublinear* time $\widetilde{O}(nd \cdot \frac{s}{m})$, which is $\widetilde{O}(sd)$ when $m \approx n$. We show this holds in our weighted semi-random model (under wRIP measurements, Definition 4) as well, where the runtime depends on the ratio of the ℓ_1 norm of the (best) weight vector to its ℓ_∞ norm, a continuous proxy for the number of RIP rows under pRIP.

We show a similar interpolation holds in the noisy measurement setting, both in the runtime sense discussed previously, and in a statistical sense. In particular, Theorem 16 achieves (up to logarithmic factors) the same interpolating runtime guarantee of Theorem 5, but further attains a squared ℓ_2 error which is roughly the average of the m largest squared elements of the noise ξ (see Theorem 3). This bound thus improves as $m \gg \widetilde{\Theta}(s)$; we show it naturally extends to weighted RIP matrices (Definition 4, generalizing Definition 1), depending on the ℓ_∞-ℓ_1 ratio of the weights.

Organization. We provide notation used throughout the paper and supplementary material in Section A, and a brief technical overview of our algorithms in Section B. The remaining proofs of our results, and counterexamples to existing methods, can be found in Sections C, D, E, and F.

3. Related work

Sparse recovery. Sparse recovery, and variants thereof, are fundamental statistical and algorithmic problems which have been studied in many settings, e.g. signal processing [7, 15, 38, 53, 74], and compressed sensing [17–19, 37, 73]. A full review of the sparse recovery literature is out of the scope of the present paper; we refer the reader to e.g. [28, 39, 52, 75] for more extensive surveys.

Within the sparse recovery literature, arguably the closest line of work to ours is that which aims to design efficient algorithms which work when the restricted condition number of the sensing matrix is large. Indeed, it is known that many non-convex methods fail when the restricted condition number of the sensing matrix is far from 1, which is often the case in applications [48]. To address this, several works [48, 75] have designed novel non-convex methods which still converge, when the restricted condition number of the matrix is much larger than 1. However, these methods still require that the restricted condition number is bounded, whereas in our setting, the restricted condition number could be arbitrarily large due to the generality of our adversary.

Another related line of work considers the setting where, instead of having a sensing matrix with rows which are drawn from an isotropic Gaussian, rows are drawn from $\mathcal{N}(0, \mathbf{\Sigma})$, for potentially ill-conditioned $\mathbf{\Sigma}$ [10, 12, 26, 48, 49, 51, 71, 79, 83]. This setting is related to our semi-random adversarial model, in that the information-theoretic content of the problem does not change, but obtaining efficient algorithms which match the optimal statistical rates is very challenging. However, there does not appear to be any further concrete connection between this "ill-conditioned covariance" setting and the semi-random model we consider. Indeed, the ill-conditioned setting appears to be qualitatively much more difficult for algorithms: [49] shows evidence that there are in fact no efficient algorithms that achieve the optimal statistical rates, without additional assumptions on $\mathbf{\Sigma}$. In contrast in the semi-random setting, polynomial-time convex programming approaches, while having potentially undesirable superlinear runtimes, still obtain optimal statistical guarantees.

Finally as discussed earlier in the introduction, there is a large body of work on efficient algorithms for sparse recovery in an RIP matrix (or a matrix satisfying weaker or stronger analogous properties). These works e.g. [2, 8, 9, 15–19, 25, 27, 43, 44, 60, 61, 66–69] are typically based on convex programming or different iterative first-order procedures.

Semi-random models. Semi-random models were originally introduced in a sequence of innovative papers [14, 41] in the context of graph coloring. In theoretical computer science, semi-random models have been explored in many settings, for instance, for various graph-structured [21, 41, 42, 55, 57] and constraint satisfaction problems [50]. More recently, they have also been studied for learning tasks such as clustering problems and community detection [22, 40, 45, 56, 58, 59, 63, 65], matrix completion [23], and linear regression [54]. We refer the reader to [72] for a more thorough overview of this vast literature. We remark that our investigation of the semi-random sparse recovery problem is heavily motivated by two recent works [23, 54] which studied the robustness of *fast iterative methods* to semi-random modeling assumptions. Finally, the *Massart noise* model in PAC learning [62] is a semi-random variant of the random classification noise model. However, this setting appears quite different from ours: until recently, polynomial-time algorithms were not known to be achievable for several fundamental problems under Massart noise [20, 30–36, 82].

References

[1] Alekh Agarwal, Sahand N. Negahban, and Martin J. Wainwright. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 37–45, 2010.

[2] Alekh Agarwal, Sahand Negahban, and Martin J Wainwright. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, pages 2452–2482, 2012.

[3] Abhishek Aich and P Palanisamy. On application of omp and cosamp algorithms for doa estimation problem. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 1983–1987. IEEE, 2017.

[4] Frank J Anscombe. Rejection of outliers. *Technometrics*, 2(2):123–146, 1960.

[5] Pranjal Awasthi and Aravindan Vijayaraghavan. Towards learning sparsely used dictionaries with arbitrary supports. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 283–296. IEEE Computer Society, 2018.

[6] Afonso S Bandeira, Edgar Dobriban, Dustin G Mixon, and William F Sawin. Certifying the restricted isometry property is hard. *IEEE transactions on information theory*, 59(6):3448–3450, 2013.

[7] Richard G Baraniuk, Volkan Cevher, Marco F Duarte, and Chinmay Hegde. Model-based compressive sensing. *IEEE Transactions on information theory*, 56(4):1982–2001, 2010.

[8] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[9] Stephen Becker, Jérôme Bobin, and Emmanuel J Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.

[10] Pierre C Bellec. The noise barrier and the large signal bias of the lasso and other convex estimators. *arXiv preprint arXiv:1804.01230*, 2018.

[11] Aditya Bhaskara, Aravinda Kanchana Ruwanpathirana, and Maheshakya Wijewardena. Principal component regression with semirandom observations via matrix completion. In *International Conference on Artificial Intelligence and Statistics*, pages 2665–2673. PMLR, 2021.

[12] Peter J Bickel, Ya'acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of statistics*, 37(4):1705–1732, 2009.

[13] Avrim Blum. Machine learning: My favorite results, directions, and open problems. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 2–2. IEEE, 2003.

[14] Avrim Blum and Joel Spencer. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms*, 19(2):204–234, 1995.

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

[16] Thomas Blumensath and Mike E Davies. Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal of selected topics in signal processing*, 4(2):298–309, 2010.

[17] Emmanuel J Candes and Justin K Romberg. Signal recovery from random projections. In *Computational Imaging III*, volume 5674, pages 76–86. International Society for Optics and Photonics, 2005.

[18] Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12):5406–5425, 2006.

[19] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

[20] Sitan Chen, Frederic Koehler, Ankur Moitra, and Morris Yau. Classification under misspecification: Halfspaces, generalized linear models, and connections to evolvability. *arXiv preprint arXiv:2006.04787*, 2020.

[21] Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. *arXiv preprint arXiv:1210.3335*, 2(5), 2012.

[22] Yudong Chen, Ali Jalali, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. *The Journal of Machine Learning Research*, 15(1):2213–2238, 2014.

[23] Yu Cheng and Rong Ge. Non-convex matrix completion against a semi-random adversary. In *Conference On Learning Theory*, pages 1362–1394. PMLR, 2018.

[24] Michael B. Cohen, Yin Tat Lee, Gary L. Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 9–21. ACM, 2016.

[25] Patrick L Combettes and Valérie R Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[26] Arnak S Dalalyan, Mohamed Hebiri, and Johannes Lederer. On the prediction performance of the lasso. *Bernoulli*, 23(1):552–581, 2017.

[27] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11): 1413–1457, 2004.

[28] Mark A Davenport, Marco F Duarte, Yonina C Eldar, and Gitta Kutyniok. Introduction to compressed sensing., 2012.

[29] Mark A Davenport, Deanna Needell, and Michael B Wakin. Signal space cosamp for sparse recovery with redundant dictionaries. *IEEE Transactions on Information Theory*, 59(10):6820–6829, 2013.

[30] Ilias Diakonikolas and Daniel M Kane. Hardness of learning halfspaces with massart noise. *arXiv preprint arXiv:2012.09720*, 2020.

[31] Ilias Diakonikolas, Themis Gouleakis, and Christos Tzamos. Distribution-independent pac learning of halfspaces with massart noise. *arXiv preprint arXiv:1906.10075*, 2019.

[32] Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning halfspaces with massart noise under structured distributions. In *Conference on Learning Theory*, pages 1486–1513. PMLR, 2020.

[33] Ilias Diakonikolas, Russell Impagliazzo, Daniel Kane, Rex Lei, Jessica Sorrell, and Christos Tzamos. Boosting in the presence of massart noise. *arXiv preprint arXiv:2106.07779*, 2021.

[34] Ilias Diakonikolas, Daniel M Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Threshold phenomena in learning halfspaces with massart noise. *arXiv preprint arXiv:2108.08767*, 2021.

[35] Ilias Diakonikolas, Daniel M Kane, and Christos Tzamos. Forster decomposition and learning halfspaces with noise. *arXiv preprint arXiv:2107.05582*, 2021.

[36] Ilias Diakonikolas, Jongho Park, and Christos Tzamos. Relu regression with massart noise. *arXiv preprint arXiv:2109.04623*, 2021.

[37] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4): 1289–1306, 2006.

[38] David L Donoho and Philip B Stark. Uncertainty principles and signal recovery. *SIAM Journal on Applied Mathematics*, 49(3):906–931, 1989.

[39] Yonina C Eldar and Gitta Kutyniok. *Compressed sensing: theory and applications*. Cambridge university press, 2012.

[40] Micha Elsner and Warren Schudy. Bounding and comparing methods for correlation clustering beyond ilp. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 19–27, 2009.

[41] Uriel Feige and Joe Kilian. Heuristics for semirandom graph problems. *Journal of Computer and System Sciences*, 63(4):639–671, 2001.

[42] Uriel Feige and Robert Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Structures & Algorithms*, 16(2):195–208, 2000.

[43] Mário AT Figueiredo and Robert D Nowak. An em algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 12(8):906–916, 2003.

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

[45] Amir Globerson, Tim Roughgarden, David Sontag, and Cafer Yildirim. Tight error bounds for structured prediction. *arXiv preprint arXiv:1409.5834*, 2014.

[46] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[47] Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pages 73–101, 1964.

[48] Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. In *NIPS*, 2014.

[49] Jonathan Kelner, Frederic Koehler, Raghu Meka, and Dhruv Rohatgi. On the power of preconditioning in sparse linear regression. *arXiv preprint arXiv:2106.09207*, 2021.

[50] Alexandra Kolla, Konstantin Makarychev, and Yury Makarychev. How to play unique games against a semi-random adversary: Study of semi-random models of unique games. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 443–452. IEEE, 2011.

[51] Vladimir Koltchinskii and Stanislav Minsker. l_1-penalization in functional linear regression with subgaussian design. *Journal de l'Ecole polytechnique-Mathématiques*, 1:269–330, 2014.

[52] Gitta Kutyniok. Theory and applications of compressed sensing. *GAMM-Mitteilungen*, 36(1): 79–101, 2013.

[53] Shlomo Levy and Peter K Fullagar. Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution. *Geophysics*, 46(9):1235–1243, 1981.

[54] Jerry Li, Aaron Sidford, Kevin Tian, and Huishuai Zhang. Well-conditioned methods for ill-conditioned systems: Linear regression with semi-random noise, 2020.

[55] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 367–384, 2012.

[56] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Sorting noisy data with partial information. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 515–528, 2013.

[57] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Constant factor approximation for balanced cut in the pie model. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 41–49, 2014.

[58] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Correlation clustering with noisy partial information. In *Conference on Learning Theory*, pages 1321–1342. PMLR, 2015.

[59] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Learning communities in the presence of errors. In *Conference on learning theory*, pages 1258–1291. PMLR, 2016.

[60] Arian Maleki and David L Donoho. Optimally tuned iterative reconstruction algorithms for compressed sensing. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):330–341, 2010.

[61] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

[62] Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5):2326–2366, 2006.

[63] Claire Mathieu and Warren Schudy. Correlation clustering with noisy input. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 712–728. SIAM, 2010.

[64] Ankur Moitra. What does robustness say about algorithms. ICML '17 Tutorial, 2017. URL https://people.csail.mit.edu/moitra/docs/robusttutorialpt2.pdf.

[65] Ankur Moitra, William Perry, and Alexander S Wein. How robust are reconstruction thresholds for community detection? In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 828–841, 2016.

[66] Deanna Needell and Joel A Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and computational harmonic analysis*, 26(3):301–321, 2009.

[67] Deanna Needell and Roman Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE Journal of selected topics in signal processing*, 4(2):310–316, 2010.

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

[69] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993.

[70] Luisa F Polania, Rafael E Carrillo, Manuel Blanco-Velasco, and Kenneth E Barner. Exploiting prior knowledge in compressed sensing wireless ecg systems. *IEEE journal of Biomedical and Health Informatics*, 19(2):508–519, 2014.

[71] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Restricted eigenvalue properties for correlated gaussian designs. *The Journal of Machine Learning Research*, 11:2241–2259, 2010.

[72] Tim Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2021.

[73] Mark Rudelson and Roman Vershynin. Sparse reconstruction by convex relaxation: Fourier and gaussian measurements. In *2006 40th Annual Conference on Information Sciences and Systems*, pages 207–212. IEEE, 2006.

[74] Fadil Santosa and William W Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986.

[75] Ludwig Schmidt. *Algorithms above the noise floor*. PhD thesis, Massachusetts Institute of Technology, 2018.

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

[77] John W Tukey. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, pages 448–485, 1960.

[78] John W Tukey. Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, volume 2, pages 523–531, 1975.

[79] Sara Van de Geer and Johannes Lederer. The lasso, correlated design, and improved oracle inequalities. In *From Probability to Statistics and Back: High-Dimensional Models and Processes–A Festschrift in Honor of Jon A. Wellner*, pages 303–316. Institute of Mathematical Statistics, 2013.

[80] Jan van den Brand, Yin Tat Lee, Aaron Sidford, and Zhao Song. Solving tall dense linear programs in nearly linear time. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 775–788, 2020.

[81] Jan van den Brand, Yin Tat Lee, Yang P. Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, mdps, and ℓ_1-regression in nearly linear time for dense instances. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 859–869, 2021.

[82] Chicheng Zhang and Yinan Li. Improved algorithms for efficient active learning halfspaces with massart and tsybakov noise. *arXiv preprint arXiv:2102.05312*, 2021.

[83] Yuchen Zhang, Martin J Wainwright, and Michael I Jordan. Optimal prediction for sparse linear models? lower bounds for coordinate-separable m-estimators. *Electronic Journal of Statistics*, 11(1):752–799, 2017.

[84] Zhimin Zhang, Shoushui Wei, Dingwen Wei, Liping Li, Feng Liu, and Chengyu Liu. Comparison of four recovery algorithms used in compressed sensing for ecg signal processing. In *2016 Computing in Cardiology Conference (CinC)*, pages 401–404. IEEE, 2016.

Appendix A. Preliminaries

General notation. We let $[n] := \{i \in \mathbb{N}, 1 \leq i \leq n\}$. The ℓ_p norm of a vector is denoted $\|\cdot\|_p$, and the sparsity (number of nonzero entries) of a vector is denoted $\|\cdot\|_0$. For a vector $v \in \mathbb{R}^d$ and $k \in [d]$, we let $v_{(k)}$ be the vector equalling v on the largest k entries of v in absolute value (with other coordinates zeroed out). The all-zeroes vector of dimension n is denoted 0_n. The nonnegative probability simplex in dimension n (i.e. $\|p\|_1 = 1, p \in \mathbb{R}^n_{\geq 0}$) is denoted Δ^n.

For mean $\mu \in \mathbb{R}^d$ and positive semidefinite covariance $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$, $\mathcal{N}(\mu, \mathbf{\Sigma})$ denotes the corresponding multivariate Gaussian. $i \sim_{\text{unif.}} S$ denotes a uniform random sample from set S. For $N \in \mathbb{N}$ and $p \in \Delta^n$ we we use $\text{Multinom}(N, p)$ to denote the probability distribution corresponding to N independent draws from $[n]$ as specified by p.

Sparsity. We say v is *s-sparse* if $\|v\|_0 \leq s$. We define the *numerical sparsity* of a vector by $\text{NS}(v) := \|v\|_1^2 / \|v\|_2^2$. Note that from the Cauchy-Schwarz inequality, if $\|v\|_0 \leq s$, then $\text{NS}(v) \leq s$.

Matrices. Matrices are in boldface throughout. The zero and identity matrix of appropriate dimension from context are $\mathbf{0}$ and $\mathbf{I}$. For a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, we let its rows be $\mathbf{A}_{i:}$, $i \in [n]$ and its columns be $\mathbf{A}_{:j}$, $j \in [d]$. The set of $d \times d$ symmetric matrices is $\mathbb{S}^d$, and its positive definite and positive semidefinite restrictions are $\mathbb{S}^d_{\succ 0}$ and $\mathbb{S}^d_{\succeq 0}$. We use the Loewner partial order $\preceq$ on $\mathbb{S}^d$. The largest entry of a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ is denoted $\|\mathbf{A}\|_{\max} := \max_{i\in[n], j\in[d]} |\mathbf{A}_{ij}|$. When a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ is clear from context, we refer to its rows as $\{a_i\}_{i\in[n]}$.

Short-flat decompositions. Throughout we frequently use the notion of "short-flat decompositions." We say $v \in \mathbb{R}^d$ has a (C_2, C_∞) *short-flat decomposition* if $v = p + e$ for some $e \in \mathbb{R}^d$ with $\|e\|_2 \leq C_2$ and $p \in \mathbb{R}^d$ with $\|p\|_\infty \leq C_\infty$. Further, we use $\text{trunc}(v, c) \in \mathbb{R}^d$ for $c \in \mathbb{R}_{\geq 0}$ to denote the vector which coordinatewise $[\text{trunc}(v, c)]_i = \text{sgn}(v_i) \max(|v_i| - c, 0)$ (i.e. the result of adding or subtracting at most c from each coordinate to decrease the coordinate's magnitude). Note that $v \in \mathbb{R}^d$ has a (C_2, C_∞) short-flat decomposition if and only if $\|\text{trunc}(v, C_\infty)\|_2 \leq C_2$ (in which case $p = \text{trunc}(v, C_\infty)$ and $e = v - p$ is such a decomposition).

Restricted isometry property. We say that matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ satisfies the (s, c)*-restricted isometry property (RIP)* or (more concisely) $\mathbf{A}$ *is* (s, c)*-RIP*, if for all s-sparse vectors $v \in \mathbb{R}^d$,

$$\frac{1}{c} \|v\|_2^2 \leq \|\mathbf{A}v\|_2^2 \leq c \|v\|_2^2 .$$

Appendix B. Technical overview

Our overall approach for semi-random sparse recovery is fairly different from two recent works in the literature which designed fast iterative methods succeeding under a semi-random adversarial model [23, 54]. In particular, these two algorithms were both based on the following natural framework, which separates the "planted learning" problem (e.g. identifying the planted benign matrix) from the "estimation" task (e.g. solving a linear system or regression problem).

1. Compute a set of weights for the data (in linear regression for example, these are weights on each of the rows of a measurement matrix $\mathbf{A}$), such that after re-weighting, the data fits the input assumptions of a fast iterative method which performs well on a fully random instance.

2. Apply said fast iterative algorithm on the reweighted data in a black-box manner.

To give a concrete example, [54] studied the standard problem of *overdetermined* linear regression with a semi-random adversary, where a measurement matrix $\mathbf{A}$ is received with the promise that $\mathbf{A}$ contains a "well-conditioned core" $\mathbf{G}$. The algorithm of [54] first learned a re-weighting of the rows of $\mathbf{A}$ by a diagonal matrix $\mathbf{W}^{\frac{1}{2}}$, such that the resulting system in $\mathbf{A}^\top \mathbf{W}\mathbf{A}$ is well-conditioned and hence can be solved using standard first-order methods.

In the case of semi-random sparse recovery, there appear to be significant barriers to reweighting approaches (which we will shortly elaborate on). We take a novel direction that involves designing a new nearly-linear time iterative method for sparse recovery tailored to the geometry of the problem.

Why not (globally) reweight the rows? There are several difficulties immediately encountered when trying to use the aforementioned reweighting framework for sparse recovery. First of all, there is no effective analog of condition number for an underdetermined linear system. The standard assumption on the measurement matrix $\mathbf{A}$ to make sparse recovery tractable for fast iterative methods is that $\mathbf{A}$ satisfies RIP, i.e. $\mathbf{A}$ is roughly an isometry when restricted to $O(s)$-sparse vectors. However, RIP is NP-hard to verify [6] and this may suggest that it is computationally hard to try, say, learning a reweighting of the rows of $\mathbf{A}$ such that the resulting reweighted matrix is guaranteed to be RIP (though it would be very interesting if this were achievable). More broadly, almost all explicit conditions (e.g. RIP, incoherence etc.) which make sparse recovery tractable for fast algorithms are conditions about subsets of the *columns* of $\mathbf{A}$. Thus, any approach which reweights rows of $\mathbf{A}$ such that column subsets of the reweighted matrix satisfy an appropriate condition results in optimization problems that seems challenging to solve in nearly-linear time.

We circumvent these difficulties in two steps. First, we propose a new analysis of an iterative method based on (reweighted) projected gradient descent, which obtains a fast convergence rate whenever each step satisfies certain locally verifiable properties. Next, our algorithm computes a sequence of *local reweightings* (which can be different in each iteration) of the rows of our measurement matrix, such that each local reweighting satisfies our requisite progress conditions for that step. We use the existence of a global reweighting satisfying RIP to demonstrate that each local reweighting subproblem has a good solution, and design an efficient method for computing each local reweighting. Our framework of bypassing hardness of computing a global reweighting to recover planted statistical information, by instead designing an iterative method capable of exploiting local reweightings with (computationally tractable) certifiable progress conditions, is quite general, and we hope it will find uses in semi-random settings beyond our particular problem.

The geometry of sparse recovery. We now explain our new approach, and how we derive deterministic conditions on the steps of an iterative method which certify progress by exploiting the geometry of sparse recovery. We focus on the clean observation setting in this technical overview. Suppose that we wish to solve a sparse regression problem $\mathbf{A}x^\star = b$ where $x^\star$ is s-sparse, and we are given $\mathbf{A}$ and b. To fix a scale, suppose for simplicity that we know $\|x^\star\|_1 = \sqrt{s}$ and $\|x^\star\|_2 = 1$. Also, assume for the purpose of conveying intuition that $\mathbf{A}$ is pRIP, and that the planted matrix $\mathbf{G}$ in Definition 1 is an entrywise random Gaussian matrix.

We next consider a natural family of iterative algorithms for solving the system $\mathbf{A}x^\star = b$. For simplicity, assume that our current iterate is $x_t = 0$ (such that b is the residual vector $b - \mathbf{A}x_t$; more generally, in the following description b will be replaced by the current residuals). Inspired by gradient methods for solving regression, we consider "first-order" algorithms of the following

form:

$$\begin{aligned} y_t &\leftarrow x_t + \mathbf{A}^\top u \\ x_{t+1} &\leftarrow \mathbf{\Pi}(y_t) \end{aligned} \tag{1}$$

where $u \in \mathbb{R}^n$ are coefficients to be computed (for example, a natural attempt could be to set u to be a multiple of $\mathbf{A}x_t - b$, resulting in the step being a scaled gradient of $\|\mathbf{A}x - b\|_2^2$ at x_t) and $\mathbf{\Pi}$ denotes projection onto the (convex) ℓ_1 ball of radius $\sqrt{s}$. Our goal will be to make constant-factor progress in terms of $\|x_t - x^\star\|_2$ in each application of (1), to yield a $\widetilde{O}(1)$ iteration method. Note that x_{t+1} must at minimum make constant factor progress in the direction $x^\star - x_t$ (in terms of decreasing the projection onto this direction) if we hope to make constant factor progress in overall distance to $x^\star$. In other words, we must have

$$\langle x_{t+1} - x_t, x^\star - x_t \rangle = \Omega(\|x^\star - x_t\|_2^2) = \Omega(1)\,.$$

First, observe that by the definition of our step and shifting so that $x_t = 0$, the point y_t satisfies

$$\langle y_t - x_t, x^\star - x_t \rangle = \langle \mathbf{A}^\top u, x^\star - x_t \rangle = u^\top (b - \mathbf{A}x_t) = u^\top b,$$

so to obtain a corresponding progress lower bound for the move to y_t, we require

$$u^\top b = \Omega(1). \tag{2}$$

Of course, this condition alone is not enough, for two reasons: the step also moves in directions orthogonal to $x^\star - x_t$, and we have not accounted for the projection step $\mathbf{\Pi}$. If all of the rows of $\mathbf{A}$ were random, then standard Gaussian concentration implies that we expect $\|b\|_2 = \sqrt{\frac{n}{d}}$, and thus to satisfy (2) we need $\|u\|_2 \geq \sqrt{\frac{d}{n}}$. This also implies

$$\|y_t - x_t\|_2 = \left\|\mathbf{A}^\top u\right\|_2 \approx \sqrt{\frac{d}{n}},$$

since for Gaussian $\mathbf{A}$, we expect that its rows are roughly orthogonal. Moreover, since $\sqrt{\frac{d}{n}} \gg 1$ in the typical underconstrained setting, almost all of the step from x_t to y_t is actually orthogonal to the desired "progress direction" parallel to $x^\star - x_t$. This appears to be a serious problem, because in order to argue that our algorithm makes progress, we need to argue that the ℓ_1 projection step $x_{t+1} = \mathbf{\Pi}(y_t)$ "undoes" this huge ℓ_2 movement orthogonal to the progress direction (see Figure 1).

Our key geometric insight is that the ℓ_1 ball is very thin in most directions (i.e. has small diameter in these directions), but is thick (i.e. has large diameter) in directions that correspond to "numerically sparse" vectors, namely vectors with bounded ℓ_1 to ℓ_2 ratios. Crucially, the movement of our step in the progress direction parallel to $x^\star - x_t$ is numerically sparse because $x^\star - x_t$ is itself $O(s)$-numerically sparse by assumption. However, for Gaussian $\mathbf{A}$, the motion in the subspace orthogonal to the progress direction ends up being essentially random (and thus is both non-sparse and ℓ_∞-bounded). Formally, we leverage this decomposition to show that the ℓ_1 projection keeps most of the forward movement in the progress direction $x^\star - x_t$ but effectively filters out much of the orthogonal motion, as demonstrated by Figure B.

This geometric intuition is the basis for the deterministic conditions we require of the steps of our iterative method, to guarantee that it is making progress. More precisely, the main condition we

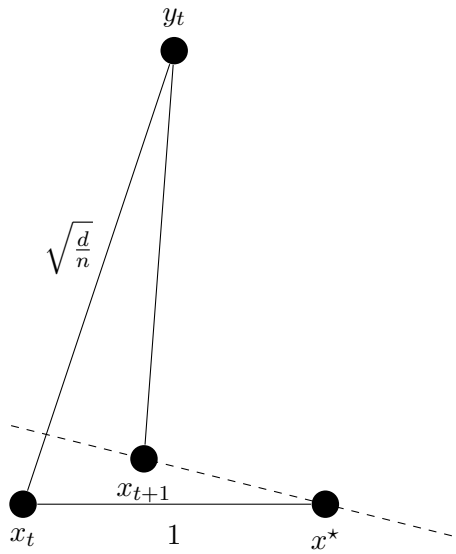

Figure 1: The effect of ℓ_1 projection on iterate progress. The dashed line represents a facet of the ℓ_1-ball around x_t of radius $\|x_t - x^\star\|_1$.

require of our step in each iteration, parameterized by the coefficients u used to induce (1), is that $y_t - x_t$ has a "short-flat" decomposition into two vectors $p + e$ where

$$\|p\|_2 = O(1) \text{ is "short" and } \|e\|_\infty = O\left(\frac{1}{\sqrt{s}}\right) \text{ is "flat".}$$

Intuitively, the short component is a proxy for the progress component (in the direction towards $x^\star$), and the flat component is a proxy for the orthogonal component, which should be filtered out by the projection step onto an ℓ_1 ball. The above bounds are rescaled appropriately in our actual method. We state these requirements formally (in the clean observation case, for example) in Definition 6, where we define a "step oracle" which is guaranteed to make constant-factor progress towards $x^\star$. By combining the above short-flat decomposition requirement with a progress requirement such as (2), we can show that as long as we can repeatedly implement a satisfactory step, our method is guaranteed to converge rapidly.

This framework for sparse recovery effectively reduces the learning problem to the problem of implementing an appropriate step oracle. Note that a valid step always exists by only looking at the Gaussian rows. To complete our algorithm, we give an implementation of this step oracle (i.e. actually solving for a valid step) which runs in nearly-linear time even when the data is augmented by a semi-random adversary (that is, our measurement matrix is pRIP or wRIP rather than RIP), and demonstrate that our framework readily extends to the noisy observation setting. Our step oracle is motivated by stochastic gradient methods. In particular, we track potentials corresponding to the progress made by our iterative method and the short-flat decomposition, and show that uniformly sampling rows of a pRIP (or wRIP) matrix $\mathbf{A}$ and taking steps in the direction of these rows which maximally improve our potentials rapidly implements a step oracle.

Appendix C. Exact recovery

In this section, we give an algorithm for solving the underconstrained linear system $\mathbf{A}x^\star = b$ given the measurement matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ (for $n \le d$) and responses $b \in \mathbb{R}^n$ (i.e. noiseless or "exact" regression), and $x^\star$ is s-sparse. Our algorithm succeeds when $\mathbf{A}$ is weighted RIP (wRIP), i.e. it satisfies Definition 4, a weighted generalization of Definition 1.

Definition 4 (wRIP matrix) *Let $w^\star_\infty \in [0,1]$. We say $\mathbf{A} \in \mathbb{R}^{n\times d}$ is $(\rho, w^\star_\infty)$-wRIP if $\|\mathbf{A}\|_{\max} \le \rho$, and there exists a weight vector $w^\star \in \Delta^n$ satisfying $\|w^\star\|_\infty \le w^\star_\infty$, such that $\mathbf{diag}\,(w^\star)^{\frac{1}{2}}\,\mathbf{A}$ is $(\Theta(s), \Theta(1))$-RIP for appropriate constants. When $\rho = \widetilde{O}(1)$ for brevity we say $\mathbf{A}$ is $w^\star_\infty$-wRIP.*

As discussed after Definition 1, a wRIP matrix can be thought of as arising from a "semi-random model" because it strictly generalizes our previously-defined pRIP matrix notion in Definition 1 with $w^\star_\infty = \frac{1}{m}$, by setting $w^\star$ to be $\frac{1}{m}$ times the zero-one indicator vector of rows of $\mathbf{G}$. The main result of this section is the following theorem regarding sparse recovery with wRIP matrices.

Theorem 5 *Let $\delta \in (0,1)$, $r > 0$, and suppose $R_0 \ge \|x^\star\|_2$ for s-sparse $x^\star \in \mathbb{R}^d$. Then with probability at least $1-\delta$, Algorithm 1 using Algorithm 2 as a step oracle takes as input a $(\rho, w^\star_\infty)$-wRIP matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $b = \mathbf{A}x^\star$, and computes $\hat{x}$ satisfying $\|\hat{x} - x^\star\|_2 \le r$ in time*

$$O\left(\left(nd\log^3(nd\rho)\log\left(\frac{1}{\delta}\cdot\log\frac{R_0}{r}\right)\log\left(\frac{R_0}{r}\right)\right)\cdot\left(w^\star_\infty s\rho^2\log d\right)\right).$$

Under the wRIP assumption, Theorem 5 provides a natural interpolation between the fully random and semi-random generative models. To build intuition, if a pRIP matrix contains a planted RIP matrix with $\widetilde{O}(s)$ rows (the information-theoretically minimum size), then by setting $w^\star_\infty \approx \frac{1}{\widetilde{O}(s)}$, we obtain a near-linear runtime of $\widetilde{O}(nd)$. However, in the fully random regime where $w^\star_\infty \approx \frac{1}{n}$ (i.e. all of $\mathbf{A}$ is RIP), the runtime improves $\widetilde{O}(sd)$ which is sublinear for $n \gg s$.

The roadmap of our algorithm and its analysis are as follows.

1. In Section C.1, we give an algorithm (Algorithm 1) which iteratively halves an upper bound on the radius to $x^\star$, assuming that either an appropriate step oracle (see Definition 6) based on short-flat decompositions can be implemented for each iteration, or we can certify that the input radius bound is now too loose. This algorithm is analyzed in Lemma 9.

2. We state in Assumption 1 a set of conditions on a matrix-vector pair $(\mathbf{A}, \Delta)$ centered around the notion of short-flat decompositions, which suffice to provide a sufficient step oracle implementation with high probability in nearly-linear time. In Section C.2 we analyze this implementation (Algorithm 2) in the proof of Lemma 7 assuming the inputs satisfy Assumption 1.

3. In Section C.3, we show Assumption 1, with appropriate parameters, follows from $\mathbf{A}$ being wRIP. This is a byproduct of a general equivalence we demonstrate between RIP, restricted conditioning measures used in prior work [1], and short-flat decompositions.

C.1. Radius contraction using step oracles

In this section, we provide and analyze the main loop of our overall algorithm for proving Theorem 5. This procedure, HalfRadiusSparse, takes as input an s-sparse vector x_{in} and a radius bound $R \geq \|x_{\text{in}} - x^\star\|_2$ and returns an s-sparse vector x_{out} with the guarantee $\|x_{\text{out}} - x^\star\|_2 \leq \frac{1}{2}R$. As a subroutine, it requires access to a "step oracle" $\mathcal{O}_{\text{step}}$, which we implement in Section C.2 under certain assumptions on the matrix $\mathbf{A}$.

Definition 6 (Step oracle) *We say that $\mathcal{O}_{\text{step}}$ is a $(C_{\text{prog}}, C_2, \delta)$-step oracle for $\Delta \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{n\times d}$, if the following holds. Whenever there is $v \in \mathbb{R}^d$ with $\frac{1}{4} \leq \|v\|_2 \leq 1$ and $\|v\|_1 \leq 2\sqrt{2s}$ such that $\Delta = \mathbf{A}v$, with probability $\geq 1 - \delta$, $\mathcal{O}_{\text{step}}$ returns $w \in \mathbb{R}^n_{\geq 0}$ such that the following two conditions hold. First,*

$$\sum_{i\in[n]} w_i \Delta_i^2 \geq C_{\text{prog}}. \tag{3}$$

Second, there exists a $(C_2, \frac{C_{\text{prog}}}{6\sqrt{s}})$ short-flat decomposition of $\mathbf{A}^\top \mathbf{diag}\,(w)\,\Delta$:

$$\left\| \text{trunc}\left(\mathbf{A}^\top \mathbf{diag}\,(w)\,\Delta, \frac{C_{\text{prog}}}{6\sqrt{s}}\right)\right\|_2 \leq C_2. \tag{4}$$

Intuitively, (4) guarantees that we can write $\gamma = p + e$ where p denotes a "progress" term which we require to be sufficiently short in the ℓ_2 norm, and e denotes an "error" term which we require to be small in ℓ_∞. We prove that under certain assumptions on the input $\mathbf{A}$ (stated in Assumption 1 below), we can always implement a step oracle with appropriate parameters.

Assumption 1 *The matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ satisfies the following. There is a weight vector $w^\star \in \Delta^n$ satisfying $\|w^\star\|_\infty \leq w^\star_\infty$, a constant L, $\rho \geq 1$, and a constant K (which may depend on L) such that for all $v \in \mathbb{R}^d$ with $\frac{1}{4} \leq \|v\|_2 \leq 1$ and $\|v\|_1 \leq 2\sqrt{2s}$ we have, defining $\Delta = \mathbf{A}v$:*

1. *$\mathbf{A}$ is entrywise bounded by $\pm\rho$, i.e. $\|\mathbf{A}\|_{\max} \leq \rho$.*
2. $$\frac{1}{L} \leq \sum_{i\in[n]} w_i^\star \Delta_i^2 \leq L. \tag{5}$$
3. *For $\mathbf{W}^\star := \mathbf{diag}\,(w^\star)$, there is a $(L, \frac{1}{K\sqrt{s}})$ short-flat decomposition of $\mathbf{A}^\top \mathbf{W}^\star \Delta = \sum_{i\in[n]} w_i^\star \Delta_i a_i$:*
$$\left\| \text{trunc}\left(\mathbf{A}^\top \mathbf{W}^\star \Delta, \frac{1}{K\sqrt{s}}\right)\right\|_2 \leq L. \tag{6}$$

Our Assumption 1 may also be stated in a scale-invariant way (i.e. with (5), (6) scaling with $\|v\|_2$), but it is convenient in our analysis to impose a norm bound on v. Roughly, the second property in Assumption 1 is (up to constant factors) equivalent to the "restricted strong convexity" and "restricted smoothness" assumptions of [1], which were previously shown for specific measurement matrix constructions such as random Gaussian matrices. The use of the third property in Assumption 1 (the existence of short-flat decompositions for numerically sparse vectors) in designing an efficient algorithm is a key contribution of our work. Interestingly, we show in Section C.3 that these assumptions are up to constant factors equivalent to RIP.

More specifically, we show that when $\mathbf{A}$ is wRIP, we can implement a step oracle for $\Delta = \mathbf{A}v$ where $v = \frac{1}{R}(x - x^\star)$ for some iterate x of Algorithm 1, which either makes enough progress to advance the algorithm or certifies that v is sufficiently short, by using numerical sparsity properties of v. We break this proof into two parts. In Lemma 7, we show that Assumption 1 suffices to implement an appropriate step oracle; this is proven in Section C.2. In Lemma 8, we then demonstrate the wRIP assumption with appropriate parameters implies our measurement matrix satisfies Assumption 1, which we prove by way of a more general equivalence in Section C.3.

Lemma 7 *Suppose* $\mathbf{A}$ *satisfies Assumption 1. Algorithm 2 is a* $(C_{\text{prog}}, C_2, \delta)$ *step oracle* StepOracle *for* $(\Delta, \mathbf{A})$ *with* $C_{\text{prog}} = \Omega(1)$, $C_2 = O(1)$ *running in time*

$$O\left(\left(nd\log^3(nd\rho)\log\frac{1}{\delta}\right)\cdot\left(w_\infty^\star s\rho^2\log d\right)\right).$$

Lemma 8 *Suppose* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *is* $(\rho, w_\infty^\star)$*-wRIP with a suitable choice of constants in the RIP parameters in Definition 4. Then,* $\mathbf{A}$ *also satisfies Assumption 1.*

We now give our main algorithm HalfRadiusSparse, assuming access to the step oracle $\mathcal{O}_{\text{step}}$ from Section C.2 with appropriate parameters, and that $\mathbf{A}$ obeys Assumption 1.

Algorithm 1: HalfRadiusSparse$(x_{\text{in}}, R, \mathcal{O}_{\text{step}}, \delta, \mathbf{A}, b)$

Input: s-sparse $x_{\text{in}} \in \mathbb{R}^d$, $R \geq \|x_{\text{in}} - x^\star\|_2$ for s-sparse $x^\star \in \mathbb{R}^d$, $(C_{\text{prog}}, C_2, \delta)$-step oracle $\mathcal{O}_{\text{step}}$ for all $(\Delta, \mathbf{A})$ with $\Delta \in \mathbb{R}^n$, $\delta \in (0,1)$, $\mathbf{A} \in \mathbb{R}^{n\times d}$, $b = \mathbf{A}x^\star \in \mathbb{R}^n$;
Output: s-sparse vector x_{out} that satisfies $\|x_{\text{out}} - x^\star\|_2 \leq \frac{1}{2}R$ with probability $\geq 1 - T\delta$;
Set $x_0 \leftarrow x_{\text{in}}$, $\mathcal{X} \leftarrow \{x \in \mathbb{R}^d \mid \|x - x_{\text{in}}\|_1 \leq \sqrt{2s}R\}$ $T \leftarrow \left\lceil \frac{12C_2^2}{C_{\text{prog}}^2} \right\rceil$, $\eta \leftarrow \frac{C_{\text{prog}}}{2C_2^2}$;
for $0 \leq t \leq T-1$ **do**
  $w_t \leftarrow \mathcal{O}_{\text{step}}(\Delta_t, \mathbf{A})$ for $\Delta_t \leftarrow \frac{1}{R}(\mathbf{A}x_t - b)$, $\gamma_t \leftarrow \mathbf{A}^\top \mathbf{diag}\,(w_t)\,\Delta_t = \sum_{i\in[n]}[w_t]_i[\Delta_t]_i a_i$;
  if $\sum_{i\in[n]}[w_t]_i[\Delta_t]_i^2 < C_{\text{prog}}$ or $\left\|\text{trunc}(\gamma_t, \frac{C_{\text{prog}}}{6\sqrt{s}})\right\|_2 > C_2$ **then**
    Return: $x_{\text{out}} \leftarrow [x_t]_{(s)}$
  end
  else $x_{t+1} \leftarrow \text{argmin}_{x\in\mathcal{X}} \|x - x_t - \eta R\gamma_t\|_2$;
end
Return: $x_{\text{out}} \leftarrow [x_t]_{(s)}$

Lemma 9 *Assume* $\mathbf{A}$ *satisfies Assumption 1. With probability at least* $1 - T\delta$*, Algorithm 1 succeeds (i.e.* $\|x_{out} - x^\star\|_2 \leq \frac{1}{2}R$*).*

Proof Throughout this proof, condition on the event that all step oracles succeed (which provides the failure probability via a union bound). We first observe that $x^\star \in \mathcal{X}$ because of Cauchy-Schwarz, the $2s$-sparsity of $x_{\text{in}} - x^\star$, and the assumption $\|x_{\text{in}} - x^\star\|_2 \leq R$.

Next, we show that in every iteration t of Algorithm 1,

$$\|x_{t+1} - x^\star\|_2^2 \leq \left(1 - \frac{C_2^2}{4C_{\text{prog}}^2}\right) \|x_t - x^\star\|_2^2 . \tag{7}$$

As $x^\star \in \mathcal{X}$, the optimality conditions of x_{t+1} as minimizing $\|x - (x_t - \eta R\gamma_t)\|_2^2$ over $\mathcal{X}$ imply

$$\begin{aligned} &2 \langle x_{t+1} - x_t + \eta R\gamma_t, x_{t+1} - x^\star \rangle \leq 0 \\ \implies &\|x_t - x^\star\|_2^2 - \|x_{t+1} - x^\star\|_2^2 \geq 2\eta R \langle \gamma_t, x_{t+1} - x^\star \rangle + \|x_t - x_{t+1}\|_2^2 . \end{aligned} \tag{8}$$

Hence, it suffices to lower bound the right-hand side of the above expression. Let $\gamma_t = p_t + e_t$ denote the $(C_2, \frac{C_{\text{prog}}}{6\sqrt{s}})$ short-flat decomposition of γ_t which exists by Definition 6 assuming the step oracle succeeded. We begin by observing

$$\begin{aligned} 2\eta R \langle \gamma_t, x_{t+1} - x_t \rangle + \|x_t - x_{t+1}\|_2^2 &= 2\eta R \langle e_t, x_{t+1} - x_t \rangle + 2\eta R \langle p_t, x_{t+1} - x_t \rangle + \|x_t - x_{t+1}\|_2^2 \\ &\geq -2\eta R \|e_t\|_\infty \|x_{t+1} - x_t\|_1 - \eta^2 R^2 \|p_t\|_2^2 \\ &\geq -\eta R^2 C_{\text{prog}} - \eta^2 R^2 C_2^2. \end{aligned} \tag{9}$$

The first inequality followed from Hölder on the first term, Cauchy-Schwarz on the second term, and then applying Young's inequality on the latter two terms in the preceding line. The second followed from the ℓ_1 radius of $\mathcal{X}$, and the bounds on e_t and p_t from (4). Next, from Definition 6, for $\Delta = \Delta_t = \frac{1}{R}(\mathbf{A}x_t - b)$ and $v = \frac{1}{R}(x_t - x^\star)$,

$$2\eta R \langle \gamma_t, x_t - x^\star \rangle = 2\eta R \sum_{i \in [n]} w_i \Delta_i \langle a_i, v \rangle = 2\eta R^2 \sum_{i \in [n]} w_i \Delta_i^2 \geq 2\eta R^2 C_{\text{prog}}. \tag{10}$$

Finally, (7) follows from combining (8), (9), and (10), with our choice of η, and the fact that inducting on this lemma implies the ℓ_2 distance to $x^\star$ of the iterates is monotone decreasing.

Next, we claim that regardless of whether Algorithm 1 terminates on Line 7 or Line 11, we have $\|x_t - x^\star\|_2 \leq \frac{1}{4}R$. Note that the vector $v = \frac{1}{R}(x_t - x^\star)$ satisfies $\mathbf{A}v = \Delta := \frac{1}{R}(\mathbf{A}x_t - b)$. By assumption the condition $\|v\|_1 \leq 2\sqrt{2s}$ is met (since $x_t, x^\star \in \mathcal{X}$), and upon iterating (7) on our radius bound assumption, this implies that the condition $\|v\|_2 \leq 1$ is met. Hence, if the algorithm terminated on Line 7, we must have $\|v\|_2 \leq \frac{1}{4}R \implies \|x_t - x^\star\|_2 \leq \frac{1}{4}R$, as otherwise the termination condition would have been false. On the other hand, by (7), after T steps we have

$$\|x_T - x^\star\|_2^2 \leq \exp\left(-\frac{TC_2^2}{4C_{\text{prog}}^2}\right) \|x_0 - x^\star\|_2^2 \leq \frac{1}{16}R^2.$$

We conclude that at termination, $\|x_t - x^\star\|_2 \leq \frac{1}{4}R$. Now, s-sparsity of $x^\star$ and the definition of $x_{\text{out}} = \text{argmin}_{\|x\|_0 \leq s} \|x - x_t\|_2$ imply the desired

$$\|x_{\text{out}} - x^\star\|_2 \leq \|x_{\text{out}} - x_t\|_2 + \|x^\star - x_t\|_2 \leq 2\|x^\star - x_t\|_2 \leq \frac{1}{2}R. \tag{11}$$

■

C.2. Designing a step oracle

In this section, we design a step oracle $\mathcal{O}_{\text{step}}(\Delta, \mathbf{A})$ (see Definition 6) under Assumption 1 on the input matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$. Our step oracle iteratively builds a weight vector $\bar{w} \in \mathbb{R}^n_{\geq 0}$. It will be convenient to define

$$\gamma_{\bar{w}} := \sum_{i\in[n]} \bar{w}_i \Delta_i a_i. \tag{12}$$

Note that a valid step oracle always exists (although it is unclear how to implement the following solution): namely, setting $\bar{w} = w^\star$ satisfies the oracle assumptions by the second and third conditions in Assumption 1. In order to ensure Algorithm 5 is indeed a step oracle, we track two potentials for some μ, C we will define in Algorithm 2:

$$\begin{gathered}\Phi_2(\bar{w}) := \sum_{i\in[n]} \bar{w}_i \Delta_i^2 \text{ and } \Phi_{\text{sqmax}}(\bar{w}) := \left(\min_{\|p\|_2 \leq L\|\bar{w}\|_1} \text{sqmax}_\mu\left(\gamma_{\bar{w}} - p\right)\right) + \frac{\|\bar{w}\|_1}{4CLs}, \\ \text{where } \text{sqmax}_\mu(x) := \mu^2 \log\left(\sum_{j\in[d]} \exp\left(\frac{x_j^2}{\mu^2}\right)\right).\end{gathered} \tag{13}$$

Intuitively, $\Phi_2(\bar{w})$ corresponds to progress on (3), and $\Phi_{\text{sqmax}}(\bar{w})$ is intended to track the bounds (4). We note the following fact about the sqmax function which follows from direct calculation.

Fact 1 *For all $x \in \mathbb{R}^d$, $\|x\|_\infty^2 \leq \text{sqmax}_\mu(x)$, and $\text{sqmax}_\mu(x) \geq \mu^2 \log(d)$.*

Also it will be important to note that $\Phi_{\text{sqmax}}(\bar{w})$ can be computed to high precision efficiently. We state this claim in the following and defer a full proof to Section F; we give a subroutine which performs a binary search on a Lagrange multiplier on the ℓ_2 constraint on p, and then solves for each optimal p_j using another binary search based on the Lagrange multiplier value.

Lemma 10 *Let $\delta > 0$ and $\theta \geq 0$. For any vector $\gamma \in \mathbb{R}^d$, we can solve the optimization problem*

$$\min_{\|p\|_2 \leq \theta} \text{sqmax}_\mu(\gamma - p)$$

to additive accuracy δ in time

$$O\left(d \log^2\left(\frac{\|\gamma\|_2^2}{\mu\sqrt{\delta}}\right)\right).$$

We state the full implementation of our step oracle as Algorithm 2 below.

Our main helper lemma bounds the expected increase in Φ_{sqmax} from choosing a row of $\mathbf{A}$ uniformly at random, and choosing a step size according to $w^\star$. We do not know $w^\star$, but we argue that our algorithm makes at least this much expected progress. Define the decomposition promised by (6):

$$p^\star := \text{trunc}\left(\mathbf{A}^\top \mathbf{W}^\star \Delta, \frac{1}{K\sqrt{s}}\right), \; e^\star := \mathbf{A}^\top \mathbf{W}^\star \Delta - p^\star.$$

Furthermore, define for all $i \in [n]$,

$$z^{(i)} := \eta w_i^\star (\Delta_i a_i - p^\star), \tag{15}$$

where $p^\star$ is given above. We use $\{z^{(i)}\}_{i\in[n]}$ as certificates of Φ_{sqmax}'s growth in the following.

Algorithm 2: StepOracle$(\Delta, \mathbf{A}, \delta)$

Input: $\Delta \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n\times d}$ satisfying Assumption 1, $\delta \in (0,1)$;

Output: w such that if there is $v \in \mathbb{R}^d$ with $\frac{1}{4} \le \|v\|_2 \le 1$ and $\|v\|_1 \le 2\sqrt{2s}$ such that $\Delta = \mathbf{A}v$, with probability $\ge 1-\delta$, (3), (4) are satisfied with $Cp = 1,\ C_2 = O(1)$.

$C \leftarrow 200$, $\mu \leftarrow \frac{1}{\sqrt{Cs\log d}}$, $\eta \leftarrow \frac{1}{Kw^\star_\infty s\rho^2 \log d}$, $N' \leftarrow \lceil \log_2 \frac{1}{\delta} \rceil$;

for $0 \le k \le N'$ **do**

  $w_0 \leftarrow 0_n$, $N \leftarrow \lceil \frac{5Ln}{\eta} \rceil$;

  for $0 \le t \le N$ **do**

    if $\Phi_2(w_t) \ge 1$ **then** **Return:** $w \leftarrow w_t$;

    Sample $i \sim_{\text{unif.}} [n]$;

    Compute (using Lemma 10) $d_t \in [0, \eta w^\star_\infty]$ maximizing to additive $O(\frac{\eta}{n})$

$$\Gamma_t(d) := \Phi_2(w_t + de_i) - Cs\Phi_{\text{sqmax}}(w_t + de_i) \tag{14}$$

    $w_{t+1} \leftarrow w_t + d_t e_i$;

  end

end

Return: $w \leftarrow 0_n$;

Lemma 11 *Assume that the constant K in Assumption 1 is sufficiently large, and that $\Delta = \mathbf{A}v$ where v satisfies the norm conditions in Assumption 1. Then for any $\bar{w} \in \mathbb{R}^n_{\ge 0}$ such that $\Phi_{\text{sqmax}}(\bar{w}) \le C^2\mu^2 \log d$, and $\eta \le \frac{1}{Kw^\star_\infty s\rho^2 \log d}$, we have*

$$\mathbb{E}_{i\sim_{\text{unif.}}[n]}\left[\Phi_{\text{sqmax}}(\bar{w} + \eta w^\star_i)\right] \le \Phi_{\text{sqmax}}(\bar{w}) + \frac{1}{2CLs}\cdot\frac{\eta}{n}.$$

Proof We assume for simplicity $L \ge 2\sqrt{2}$ as otherwise we may set $L \leftarrow \max(2\sqrt{2}, L)$ and (5) remains true. Let $p_{\bar{w}}$ be the minimizing argument in the definition of $\Phi_{\text{sqmax}}(\bar{w})$ in (13). For any $i \in [n]$, $p_{\bar{w}} + (\eta w^\star_i)p^\star$ is a valid argument for the optimization problem defining $\Phi_{\text{sqmax}}(\bar{w} + \eta w^\star_i)$. This follows since $\|p^\star\|_2 \le L$, and since $\|w\|_1$ grows by $\eta w^\star_i$. Next, define

$$F(x) := \sum_{j\in[d]} \exp\left(\frac{x_j^2}{\mu^2}\right) \tag{16}$$

such that $\Phi_{\text{sqmax}}(\bar{w}) = \mu^2 \log F(x) + \frac{\|\bar{w}\|_1}{4CLs}$ for $x = \gamma_{\bar{w}} - p_{\bar{w}}$. As discussed earlier, since $\|p_{\bar{w}} + (\eta w^\star_i)p^\star\|_2 \le \|p_{\bar{w}}\|_2 + \eta w^\star_i L$, we conclude

$$\Phi_{\text{sqmax}}(\bar{w} + \eta w^\star_i) \le \mu^2 \log F(x + z^{(i)}) + \frac{\|\bar{w} + \eta w^\star_i\|_1}{4CLs}. \tag{17}$$

We next compute

$$\begin{aligned}\frac{1}{n}\sum_{i\in[n]} F(x+z^{(i)}) &= \frac{1}{n}\sum_{j\in[d]} \exp\left(\frac{x_j^2}{\mu^2}\right)\left(\sum_{i\in[n]} \exp\left(\frac{2x_j z_j^{(i)} + (z_j^{(i)})^2}{\mu^2}\right)\right) \\ &\le \frac{1}{n}F(x)\max_{j\in[d]}\left(\sum_{i\in[n]} \exp\left(\frac{2x_j z_j^{(i)} + (z_j^{(i)})^2}{\mu^2}\right)\right).\end{aligned} \tag{18}$$

We now bound the right-hand side of this expression. For any $i \in [n]$ and $j \in [d]$, recalling (15),

$$\left|z_j^{(i)}\right| \le \eta w_i^\star(|\Delta_i| \, \|a_i\|_\infty + \|p^\star\|_2) \le \eta w_\infty^\star L(\sqrt{s}\rho^2 + 1). \tag{19}$$

The second inequality used our bounds from Assumption 1; note that for $\Delta = \mathbf{A}v$ where v satisfies the norm conditions in Assumption 1, $|\Delta_i| \le \rho \, \|v\|_1 \le 2\sqrt{2s}\rho$. Hence, if we choose a sufficiently large constant K in Assumption 1, we have

$$\frac{1}{\mu}\left|z_j^{(i)}\right| \le \frac{\sqrt{C}}{K\sqrt{s\log d}\rho^2} \cdot \left(L(\sqrt{s}\rho^2 + 1)\right) \le \frac{1}{4C\sqrt{\log d}}.$$

Also by the assumption that $\Phi_{\text{sqmax}}(\bar{w}) \le C^2\mu^2 \log d$ we must have that for all $j \in [d]$,

$$\frac{|x_j|}{\mu} \le C\sqrt{\log d}.$$

Now, using $\exp(c) \le 1 + c + c^2$ for $|c| \le 1$, we get

$$\begin{aligned}\sum_{i\in[n]} \exp\left(\frac{2x_j z_j^{(i)} + (z_j^{(i)})^2}{\mu^2}\right) &\le \sum_{i\in[n]}\left(1 + \frac{2x_j z_j^{(i)}}{\mu^2} + \frac{(z_j^{(i)})^2}{\mu^2} + \left(\frac{2x_j z_j^{(i)} + (z_j^{(i)})^2}{\mu^2}\right)^2\right) \\ &\le \sum_{i\in[n]}\left(1 + \frac{2x_j z_j^{(i)}}{\mu^2} + 10C^2\log d \cdot \frac{(z_j^{(i)})^2}{\mu^2}\right).\end{aligned} \tag{20}$$

We control the first-order term via the observation that $\sum_{i\in[n]} z^{(i)} = \eta e^\star$ which is ℓ_∞-bounded from (6), so taking the constant K in Assumption 1 sufficiently large, we have

$$\begin{aligned}&\left|\sum_{i\in[n]} \frac{z_j^{(i)}}{\mu}\right| \le \frac{\eta}{\mu}\|e^\star\|_\infty \le \frac{\eta\sqrt{C\log d}}{K} \\ \implies &\left|\sum_{i\in[n]} \frac{2x_j z_j^{(i)}}{\mu^2}\right| \le 2C\sqrt{\log d} \cdot \frac{\eta\sqrt{C\log d}}{K} \le \frac{\eta\log d}{8L}.\end{aligned} \tag{21}$$

In the last inequality we assumed $K \ge 16C^{1.5}L$. We control the second-order term by using $(a+b)^2 \le 2a^2 + 2b^2$, $\|p^\star\|_\infty \le \|p^\star\|_2 \le L$, and (5):

$$\sum_{i\in[n]}\left(z_j^{(i)}\right)^2 \le 2\eta^2 w_\infty^\star\left(\sum_{i\in[n]} w_i^\star [p^\star]_j^2 + \sum_{i\in[n]} w_i^\star \Delta_i^2 \rho^2\right) \le 2\eta^2 w_\infty^\star(L\rho^2 + L^2). \tag{22}$$

Putting together (20), (21), and (22), with the definition of μ, we conclude for sufficiently large K,

$$\frac{1}{n}\sum_{i\in[n]}\exp\left(\frac{2x_jz_j^{(i)}+(z_j^{(i)})^2}{\mu^2}\right)\leq 1+\frac{\eta\log d}{8Ln}+\frac{2\eta^2w_\infty^\star(L\rho^2+L^2)}{n\mu^2}$$
$$\leq 1+\frac{\eta\log d}{4Ln}.$$

Hence, combining the above with (18), and using $\log(1+c)\leq c$ for all c,

$$\mu^2\log\left(\frac{1}{n}\sum_{i\in[n]}F(x+z^{(i)})\right)\leq\mu^2\log F(x)+\frac{\mu^2\eta\log d}{4Ln}=\mu^2\log F(x)+\frac{1}{Cs}\cdot\frac{\eta}{4Ln}. \tag{23}$$

Finally, we compute via (17) and concavity of log,

$$\mathbb{E}_{i\sim_{\text{unif.}}[n]}[\Phi_{\text{sqmax}}(\bar{w}+\eta w_i^\star)]\leq\frac{\mu^2}{n}\sum_{i\in[n]}\log F(x+z^{(i)})+\frac{\|\bar{w}\|_1}{4CLs}+\frac{1}{4CLs}\left(\frac{1}{n}\sum_{i\in[n]}\eta w_i^\star\right)$$
$$\leq\mu^2\log\left(\frac{1}{n}\sum_{i\in[n]}F(x+z^{(i)})\right)+\frac{\|\bar{w}\|_1}{4CLs}+\frac{1}{4CLs}\cdot\frac{\eta}{n}$$
$$\leq\mu^2\log F(x)+\frac{\|\bar{w}\|_1}{4CLs}+\frac{1}{Cs}\cdot\frac{\eta}{2Ln}=B(\bar{w})+\frac{1}{Cs}\cdot\frac{\eta}{2Ln}.$$

In the last line, we used the bound (23). ■

Finally, we can complete the analysis of Algorithm 2.

Lemma 12 *Suppose* $\mathbf{A}$ *satisfies Assumption 1. Algorithm 2 is a* $(C_{\text{prog}}, C_2, \delta)$ *step oracle* StepOracle *for* $(\Delta, \mathbf{A})$ *with* $C_{\text{prog}}=\Omega(1)$, $C_2=O(1)$ *running in time*

$$O\left(\left(nd\log^3(nd\rho)\log\frac{1}{\delta}\right)\cdot\left(w_\infty^\star s\rho^2\log d\right)\right).$$

Proof

It suffices to prove Algorithm 2 meets its output guarantees in this time. Throughout this proof, we consider one run of Lines 5-10 of the algorithm, and prove that it successfully terminates on Line 7 with probability $\geq\frac{1}{2}$ assuming $\mathbf{A}$ satisfies Assumption 1 and that $\Delta=\mathbf{A}v$ for v satisfying the norm bounds in Assumption 1. This yields the failure probability upon repeating N' times.

For the first part of this proof, we assume we can exactly compute Δ_t, and carry out the proof accordingly. We discuss issues of approximation tolerance at the end, when bounding the runtime.

Correctness. We use the notation $A_t:=\Phi_2(w_t)$, $B_t:=\Phi_{\text{sqmax}}(w_t)$, and $\Phi_t:=A_t-CsB_t$. We first observe that A_t is 1-Lipschitz, meaning it can only increase by 1 in any given iteration; this follows from $\eta w_\infty^\star\Delta_i^2\leq\frac{1}{8s\rho^2}\Delta_i^2\leq 1$, since $\Delta_i^2=\langle a_i,v\rangle^2\leq 8s\rho^2$ by ℓ_∞-ℓ_1 Hölder.

Suppose some run of Lines 5-13 terminates by returning on Line 8 in iteration T, for $0\leq T\leq N$. The termination condition implies that $A_T\geq 1=C_{\text{prog}}$, so to show that the algorithm satisfies

Definition 6, it suffices to show existence of a short-flat decomposition in the sense of (4). Clearly, Φ_t is monotone non-decreasing in t, since we may always force $\Gamma_t = 0$ by choosing $d_t = 0$. Moreover, $\Phi_0 = -CsB_0 = -Cs\mu^2 \log d = -1$. The above Lipschitz bound implies that $A_T \leq 2$, since $A_{T-1} \leq 1$ by the termination condition; hence,

$$A_T - CsB_T = \Phi_T \geq \Phi_0 = -1 \implies B_T \leq \frac{A_T + 1}{Cs} \leq \frac{3}{Cs} \leq C^2\mu^2 \log d.$$

Note that the above inequality and nonnegativity of sqmax_μ imply that $\frac{\|w_T\|_1}{4LCs} \leq \frac{3}{Cs}$, so $\|w_T\|_1 \leq 12L$. For the given value of $C = 200$, and the first inequality in Fact 1, the definition of the first summand in B implies there is a short-flat decomposition meeting (4) with $C_2 = L\|w_T\|_1 = O(1)$.

Hence, we have shown that Definition 6 is satisfied whenever the algorithm returns on Line 7. We make one additional observation: whenever $\Phi_t \geq 0$, the algorithm will terminate. This follows since on such an iteration,

$$A_t \geq CsB_t \geq CsB_0 = Cs\mu^2 \log d = 1,$$

since clearly the function B is minimized by the all-zeroes weight vector, attaining value $\mu^2 \log d$.

Success probability. We next show that with probability at least $\frac{1}{2}$, the loop in Lines 5-10 will terminate. Fix an iteration t. When sampling $i \in [n]$, the maximum gain in Φ_t for $d_t \in [0, \eta w_\infty^\star]$ is at least that attained by setting $d_t = \eta w_i^\star$, and hence

$$\mathbb{E}[\Phi_{t+1} - \Phi_t \mid A_t \leq 1] \geq \frac{\eta}{Ln} - \frac{\eta}{2Ln} = \frac{\eta}{2Ln}. \tag{24}$$

Here, we used that the expected gain in A_t by choosing $d_t = \eta w_i^\star$ over a uniformly sampled $i \in [n]$ is lower bounded by $\frac{\eta}{Ln}$ via (5), and the expected gain in CsB_t is upper bounded by Lemma 11.

Let Z_t be the random variable equal to $\Phi_t - \Phi_0$, where we freeze the value of $w_{t'}$ for all $t' \geq t$ if the algorithm ever returns on Line 8 in an iteration t. Notice that $Z_t \leq 2$ always: whenever $Z_t \geq 1$, we have $\Phi_t \geq 0$ so the algorithm will terminate, and Z_t is 1-Lipschitz because A_t is. Moreover, whenever we are in an iteration t where $\Pr[A_t \geq 1] \leq \frac{1}{2}$, applying (24) implies

$$\mathbb{E}[Z_{t+1} - Z_t] = \mathbb{E}[Z_{t+1} - Z_t \mid A_t \leq 1] \Pr[A_t \leq 1] \geq \frac{\eta}{4Ln}.$$

Clearly, $\Pr[A_t \geq 1]$ is a monotone non-decreasing function of t, since A_t is monotone. After $N \geq \frac{5Ln}{\eta}$ iterations, if we still have $\Pr[A_t \geq 1] \leq \frac{1}{2}$, we would obtain a contradiction since recursing the above display yields $\mathbb{E}[Z_N] > 2$. This yields the desired success probability.

Runtime. The cost of each iteration is dominated by the following computation in Line 9: we wish to find $d \in [0, \eta w_\infty^\star]$ maximizing to additive $O(\frac{\eta}{n})$ the following objective:

$$\Phi_2(w + de_i) - Cs\Phi_{\text{sqmax}}(w + de_i).$$

We claim the above function is a concave function of d. First, we show Φ_{sqmax} is convex (and the result will then follow from linearity of Φ_2). To see this, for two values w_i and w_i', let the corresponding maximizing arguments in the definition of $\Phi_{\text{sqmax}}(\bar{w} + w_i)$ and $\Phi_{\text{sqmax}}(\bar{w} + w_i')$ be denoted p and p'. Then, $\frac{1}{2}(p + p')$ is a valid argument for $\bar{w} + \frac{1}{2}(w_i + w_i')$, and by convexity of sqmax_μ and linearity of the ℓ_1 portion, we have the conclusion.

Next, note that all $|\Delta_i|$ are bounded by $2\sqrt{2s}\rho$ (proven after (19)) and all a_{ij} are bounded by ρ by assumption. It follows that the restriction of Φ_2 to a coordinate is $8s\rho^2$-Lipschitz. Moreover the linear portion of Φ_{sqmax} is clearly $\frac{1}{4CLs}$-Lipschitz in any coordinate. Finally we bound the Lipschitz constant of the sqmax part of Φ_{sqmax}. It suffices to bound Lipschitzness for any fixed p of

$$\text{sqmax}_\mu\left(\gamma_{\bar{w}} - p + d_i\Delta_i a_i\right)$$

because performing the minimization over p involved in two $\text{sqmax}(\gamma_{\bar{w}} - p + d\Delta_i a_i)$ and $\text{sqmax}(\gamma_{\bar{w}} - p + d'\Delta_i a_i)$ can only bring the function values closer together. By direct computation the derivative of the above quantity with respect to d_i is

$$\sum_{j\in[d]} \Delta_i a_{ij}\left(2\left[\gamma_{\bar{w}} - p + d_i\Delta_i a_i\right]_j\right) q_j$$

for some probability density vector $q \in \Delta^d$. Further we have

$$\left|\gamma_{\bar{w}} - p + d_i\Delta_i a_i\right|_j \le O\left(\sqrt{s}\rho^2\right) + O(1) + 2\sqrt{2s}\rho^2 \cdot (\eta w_\infty^\star).$$

Here we used our earlier proof that we must only consider values of $\|p\|_2 = O(1)$ throughout the algorithm (since $\|w_t\|_1 = O(1)$ throughout) and this also implies no coordinate of $\gamma_{\bar{w}}$ can be larger than $(\max_{i\in[n]}|\Delta_i|)(\max_{i\in[n],j\in[d]}|a_{ij}|)\|\bar{w}\|_1$ by definition of $\gamma_{\bar{w}}$. Combined with our bounds on linear portions this shows Φ_2 and Φ_{sqmax} are poly$(nd\rho)$-Lipschitz.

Hence, we may evaluate to the desired $O(\frac{\eta}{n})$ accuracy by approximate minimization of a Lipschitz convex function over an interval (Lemma 33, [24]) with a total cost of $O(d\log^3(nd\rho))$. Here we use the subroutine of Lemma 10 in Lemma 33 of [24], with evaluation time $O(d\log^2(nd\rho))$.

The algorithm then runs in NN' iterations, each bottlenecked by the cost of approximating Γ_t; combining these multiplicative factors yields the runtime. We note that we do not precompute $\Delta = \mathbf{A}v$; we can compute coordinates of Δ in time $O(d)$ as they are required by Algorithm 2. ■

C.3. Equivalence between Assumption 1 and RIP

The main result of this section is an equivalence between Assumption 1 and the weighted restricted isometry property, which requires two helper tools to prove. The first is a "shelling decomposition."

Lemma 13 *Let $v \in \mathbb{R}^d$ have $\text{NS}(v) \le \sigma$. Then if we write $v = \sum_{l\in[k]} v^{(l)}$ where $v^{(1)}$ is obtained by taking the s largest coordinates of v, $v^{(2)}$ is obtained by taking the next s largest coordinates and so on (breaking ties arbitrarily so that the supports are disjoint), we have*

$$\sum_{2\le l\le k}\left\|v^{(l)}\right\|_2 \le \sqrt{\frac{\sigma}{s}}\|v\|_2.$$

Proof Note that the decomposition greedily sets $v^{(l)}$ to be the s largest coordinates (by absolute value) of $v - \sum_{l'\in[l-1]} v^{(l')}$, zeroing all other coordinates and breaking ties arbitrarily. This satisfies

$$\left\|v^{(l+1)}\right\|_2 \le \sqrt{s}\left\|v^{(l+1)}\right\|_\infty \le \frac{1}{\sqrt{s}}\left\|v^{(l)}\right\|_1.$$

The last inequality follows since every entry of $v^{(l)}$ is larger than the largest of $v^{(l+1)}$ in absolute value. Finally, summing the above equation and using disjointness of supports yields

$$\sum_{2\le l\le k}\left\|v^{(l)}\right\|_2 \le \frac{1}{\sqrt{s}}\left\|v\right\|_1 \le \sqrt{\frac{\sigma}{s}}\left\|v\right\|_2.$$

■

The second bounds the largest entries of image vectors from the transpose of an RIP matrix.

Lemma 14 *Let $\mathbf{A}\in\mathbb{R}^{n\times d}$ be (s,c)-RIP, and let $u\in\mathbb{R}^n$. Then,*

$$\left\|[\mathbf{A}^\top u]_{(s)}\right\|_2 \le c\left\|u\right\|_2.$$

Proof Let $v=[\mathbf{A}^\top u]_{(s)}$. The lemma is equivalent to showing $\left\|v\right\|_2\le c\left\|u\right\|_2$. Note that

$$\left\|v\right\|_2^2 = \left\langle v,\mathbf{A}^\top u\right\rangle \le \left\|\mathbf{A}v\right\|_2\left\|u\right\|_2 \le c\left\|v\right\|_2\left\|u\right\|_2.$$

The first inequality used Cauchy-Schwarz, and the second applied the RIP property of $\mathbf{A}$ to v, which is s-sparse by construction. The conclusion follows via dividing by $\left\|v\right\|_2$. ■

Using these helper tools, we now prove the main result of this section.

Lemma 15 *The following statements are true.*

1. *If $\mathbf{A}$ satisfies Assumption 1 with weight vector $w^\star$, then $(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}$ is (s,L)-RIP.*
2. *If the matrix $(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}$ is RIP with parameters*

$$\left(12800L^3K^2s,\frac{\sqrt{L}}{2}\right)$$

for $L\ge 1$, and $\left\|\mathbf{A}\right\|_{\max}\le\rho$, then $\mathbf{A}$ satisfies Assumption 1.

Proof We prove each equivalence in turn.

Assumption 1 implies RIP. The statement of RIP is scale-invariant, so we will prove it for all s-sparse unit vectors v without loss of generality. Note that such v satisfies the condition in Assumption 1, since $\left\|v\right\|_2=1$ and $\left\|v\right\|_1\le\sqrt{s}$ by Cauchy-Schwarz. Then, the second condition of Assumption 1 implies that for $\Delta=\mathbf{A}v$, we have the desired norm preservation:

$$\frac{1}{L}\le\left\|(\mathbf{W}^*)^{\frac{1}{2}}\mathbf{A}v\right\|_2^2=\sum_{i\in[n]}w_i^\star\Delta_i^2\le L.$$

Boundedness and RIP imply Assumption 1. Let $v \in \mathbb{R}^d$ satisfy $\frac{1}{4} \leq \|v\|_2 \leq 1$ and $\|v\|_1 \leq 2\sqrt{2s}$, and define $\Delta := \mathbf{A}v$. The first condition in Assumption 1 is immediate from our assumed entrywise boundedness on $\mathbf{A}$, so we begin by demonstrating the lower bound in (5). Let

$$s' = 12800L^3K^2s$$

and let $v^{(1)}, \ldots, v^{(k)}$ be the shelling decomposition into s'-sparse vectors given by Lemma 13, where $\sigma = 128s$ from the ℓ_1 and ℓ_2 norm bounds on v. By Lemma 13, we have

$$\left\|v^{(2)}\right\|_2 + \cdots + \left\|v^{(k)}\right\|_2 \leq \frac{0.1}{L}\|v\|_2 .$$

In particular, the triangle inequality then implies $0.9\|v\|_2 \leq \left\|v^{(1)}\right\|_2 \leq \|v\|_2$. Next, recall that $\sum_{i\in[n]} w_i^\star \Delta_i^2 = \left\|(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}v\right\|_2^2$. By applying the triangle inequality and since $(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}$ is RIP,

$$\begin{aligned}\left\|(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}v\right\|_2^2 &\geq \left(\left\|(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}v^{(1)}\right\|_2 - \sum_{l=2}^{k}\left\|(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}v^{(l)}\right\|_2\right)^2 \\ &\geq \left(\frac{5}{\sqrt{L}}\cdot 0.9\|v\|_2 - \frac{\sqrt{L}}{2}\cdot\frac{1}{L}\|v\|_2\right)^2 \geq \frac{16}{L}\|v\|_2^2 \geq \frac{1}{L}.\end{aligned}$$

In the second inequality, we applied the RIP assumption to each individual term, since all the vectors are s'-sparse. Similarly, to show the upper bound in (5), we have

$$\begin{aligned}\left\|(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}v\right\|_2^2 &\leq \left(\left\|(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}v^{(1)}\right\|_2 + \sum_{l=2}^{k}\left\|(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}v^{(l)}\right\|_2\right)^2 \\ &\leq \left(\frac{\sqrt{L}}{2}\cdot\|v\|_2 + \frac{\sqrt{L}}{2}\cdot\frac{1}{L}\|v\|_2\right)^2 \leq L.\end{aligned}$$

It remains to verify the final condition of Assumption 1. First, for $u := \mathbf{W}^{\frac{1}{2}}\mathbf{A}v$, by applying the shelling decomposition to v into s'-sparse vectors $\{v^{(l)}\}_{l\in[k]}$,

$$\|u\|_2 \leq \sum_{l\in[k]}\left\|(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}v^{(l)}\right\|_2 \leq \sqrt{L}\|v\|_2 . \tag{25}$$

Here, we used our earlier proof to bound the contribution of all terms but $v^{(1)}$. Applying Lemma 14 to the matrix $(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}$ and vector u, we have for $\Delta = \mathbf{A}v$,

$$\left\|\left[\mathbf{A}^\top(\mathbf{W}^\star)^{\frac{1}{2}}u\right]_{(s')}\right\|_2 = \left\|\left[\mathbf{A}^\top\mathbf{W}^\star\Delta\right]_{(s')}\right\|_2 \leq L.$$

By setting the ℓ_2 bounded component in the short-flat decomposition of $\mathbf{A}^\top\mathbf{W}^\star\Delta$ to be the top s' entries by magnitude, it remains to show the remaining coordinates are ℓ_∞ bounded by $\frac{1}{K\sqrt{s}}$. This follows from the definition of s' and (25), which imply that the $s'+1^{\text{th}}$ largest coordinate (in magnitude) cannot have squared value larger than $\frac{L^2}{s'} \leq \frac{1}{K^2 s}$ without contradicting (25). ■

Finally, it is immediate that Lemma 8 follows from Lemma 15.

C.4. Putting it all together

At this point, we have assembled the tools to prove our main result on exact recovery.

Theorem 5 *Let $\delta \in (0,1)$, $r > 0$, and suppose $R_0 \geq \|x^\star\|_2$ for s-sparse $x^\star \in \mathbb{R}^d$. Then with probability at least $1-\delta$, Algorithm 1 using Algorithm 2 as a step oracle takes as input a $(\rho, w_\infty^\star)$-wRIP matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $b = \mathbf{A}x^\star$, and computes $\hat{x}$ satisfying $\|\hat{x} - x^\star\|_2 \leq r$ in time*

$$O\left(\left(nd\log^3(nd\rho)\log\left(\frac{1}{\delta}\cdot\log\frac{R_0}{r}\right)\log\left(\frac{R_0}{r}\right)\right)\cdot\left(w_\infty^\star s\rho^2\log d\right)\right).$$

Proof With probability at least $1-\delta$, combining Lemma 7 and Lemma 8 implies that Assumption 1 holds for all $v \in \mathbb{R}^d$ where $\frac{1}{4} \leq \|v\|_2 \leq 1$ and $\|v\|_1 \leq 2\sqrt{2s}$, and that for $N = O(\log\frac{R_0}{r})$, we can implement a step oracle for N runs of Algorithm 1 in the allotted time, each with failure probability $1-\frac{\delta}{N}$. Moreover, Algorithm 1 returns in $O(1)$ iterations, and allows us to halve our radius upper bound. By taking a union bound on failure probabilities and repeatedly running Algorithm 1 N times, we obtain a radius upper bound of r with probability $\geq 1-\delta$. ■

Appendix D. Noisy recovery

In this section, we give an algorithm for solving a noisy sparse recovery problem in a wRIP matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ (where we recall Definition 4). In particular, we assume that we receive

$$b = \mathbf{A}x^\star + \xi, \tag{26}$$

for an arbitrary unknown $\xi \in \mathbb{R}^n$, and $x^\star \in \mathbb{R}^d$ is s-sparse. Throughout this section, we will define

$$m := \frac{1}{w_\infty^\star}, \tag{27}$$

where $w_\infty^\star$ is an entrywise bound on w in Definition 4. We define the (unknown) "noise floor"

$$R_\xi := \frac{1}{\sqrt{m}}\left\|\xi_{(m)}\right\|_2,$$

where we defined $\cdot_{(m)}$ in Section A. Our goal will be to return x such that $\|x - x^\star\|_2 = O(R_\xi)$. We now formally state the main result of this section here.

Theorem 16 *Let $\delta \in (0,1)$, and suppose $R_0 \geq \|x^\star\|_2$ for s-sparse $x^\star \in \mathbb{R}^d$. Further, suppose $\mathbf{A} \in \mathbb{R}^{n\times d}$ is $(\rho, w_\infty^\star)$-wRIP and $b = \mathbf{A}x^\star + \xi$, and $R_1 \geq R_\xi := \left\|\xi_{(m)}\right\|_2$. Then with probability at least $1-\delta$, Algorithm 3 using Algorithm 3 as a noisy step oracle computes $\hat{x}$ satisfying*

$$\|\hat{x} - x^\star\|_2 \leq R_{\text{final}} = \Theta(R_\xi),$$

in time

$$O\left(\left(ndw_\infty^\star s\log^4(nd\rho)\log^2\left(\frac{d}{\delta}\cdot\log\left(\frac{R_0}{R_{\text{final}}}\right)\log\left(\frac{R_1}{R_{\text{final}}}\right)\right)\right)\cdot\rho^2\log\left(\frac{R_0}{R_{\text{final}}}\right)\log\left(\frac{R_1}{R_{\text{final}}}\right)\right).$$

Similarly to Theorem 5, Theorem 16 provides a runtime guarantee which interpolates between the fully random and semi-random settings, and runs in sublinear time when e.g. the entire measurement matrix $\mathbf{A}$ satisfies RIP. Theorem 16 further provides a refined error guarantee as a function of the noise vector ξ, which again interpolates based on the "quality" of the weights w. This is captured through the parameter $m = \frac{1}{w_\infty^\star}$: when $m \approx n$, the squared error bound R_ξ^2 scales as the average squared entry of ξ, and more generally it scales as the average of the largest m entries.

We solve the noisy variant by essentially following the same steps as Section C and making minor modifications to the analysis; we give an outline of the section here. In Section D.1, we generalize the framework of Section C.1 to the setting where we only receive noisy observations (26), while our current radius is substantially above the noise floor. We then implement an appropriate step oracle for this outer loop in Section D.2, and prove that the relevant Assumption 2 used in our step oracle implementation holds when $\mathbf{A}$ is wRIP in Section D.3.

D.1. Radius contraction above the noise floor using step oracles

In this section, we give the main loop of our overall noise-tolerant algorithm, HalfRadiusSparseNoisy, which takes as input s-sparse x_{in} and a radius bound $R \geq \|x_{\text{in}} - x^\star\|_2$. It then returns an s-sparse vector x_{out} with the guarantee $\|x_{\text{out}} - x^\star\|_2 \leq \frac{1}{2}R$, *as long as* R is larger than an appropriate multiple of R_ξ. We give the analog of Definition 6 in this setting, termed a "noisy step oracle."

Definition 17 (Noisy step oracle) *We say that $\mathcal{O}_{\text{nstep}}$ is a $(C_{\text{prog}}, C_2, C_\xi, \delta)$-noisy step oracle for $\widetilde{\Delta} \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{n\times d}$ if the following holds. Whenever there is $v \in \mathbb{R}^d$ with $\frac{1}{12} \leq \|v\|_2 \leq 1$ such that $\widetilde{\Delta} = \mathbf{A}v + \xi$ where $\left\|\xi_{(m)}\right\|_2 \leq \frac{\sqrt{m}}{C_\xi}$, with probability $\geq 1-\delta$, $\mathcal{O}_{\text{nstep}}$ returns $w \in \mathbb{R}^n_{\geq 0}$ such that the following two conditions hold. First,*

$$\sum_{i\in[n]} w_i \widetilde{\Delta}_i \Delta_i \geq C_{\text{prog}}. \tag{28}$$

Second, there exists a $(C_2, \frac{C_{\text{prog}}}{6\sqrt{s}})$ short-flat decomposition of $\mathbf{A}^\top \mathbf{diag}\,(w)\,\Delta$:

$$\left\|\text{trunc}\left(\mathbf{A}^\top \mathbf{diag}\,(w)\,\Delta, \frac{C_{\text{prog}}}{6\sqrt{s}}\right)\right\|_2 \leq C_2.$$

We next characterize how a strengthened step oracle with appropriate parameters also is a noisy step oracle. First, we will need a definition.

Definition 18 *For distributions A, B on $\mathbb{R}^n$, we say A stochastically dominates B if there is a random variable C on $\mathbb{R}^n$ whose coordinates are always nonnegative such that the distribution of A is the same as the distribution of $B + C$ (where C may depend on the realization of B).*

We now formalize the properties of the strengthened step oracle that we will construct.

Definition 19 (Strong step oracle) *We say that $\mathcal{O}_{\text{step}}$ is a $(C_{\text{prog}}, C_2, C_\xi, \delta)$-strong step oracle for $\widetilde{\Delta} \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{n\times d}$ if it satisfies all the properties of a standard step oracle (Definition 6), as well as the following additional guarantees.*

1. *For the output weights w, we have*

$$\|w\|_1 \leq \frac{C_{\text{prog}} C_\xi^2}{4} \cdot \delta. \tag{29}$$

2. *The distribution of w output by the oracle is stochastically dominated by the distribution*

$$\frac{\delta}{4s\rho^2 \log \frac{d}{\delta}} \text{Multinom}\left(\left\lceil \frac{C_{\text{prog}} C_\xi^2 ns\rho^2 \log \frac{d}{\delta}}{m} \right\rceil, \left(\underbrace{\frac{1}{n}, \ldots, \frac{1}{n}}_{n}\right)\right)$$

   for some $\rho \geq 1$.

3. *Compared to Definition 6 (the step oracle definition), we have the stronger guarantees that $\mathbf{A}^\top \mathbf{diag}(w)\, \Delta$ admits a $(C_2, \frac{C_{\text{prog}}}{24\sqrt{s}})$ short-flat decomposition in (4), and obtains its guarantees using the bounds $\frac{1}{12} \leq \|v\|_2 \leq 1$ (instead of a lower bound of $\frac{1}{4}$).*

We next demonstrate that a strong step oracle is a noisy step oracle.

Lemma 20 *Suppose $\mathcal{O}_{\text{step}}$ is a $(C_{\text{prog}}, C_2, C_\xi, \delta)$-strong step oracle for $\widetilde{\Delta} \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{n \times d}$. Then, $\mathcal{O}_{\text{step}}$ is also a $(\frac{1}{4} C_{\text{prog}}, C_2, C_\xi, 2\delta)$-noisy step oracle for $(\widetilde{\Delta}, \mathbf{A})$.*

Proof In the definition of a noisy step oracle, we only need to check the condition that $\sum_{i \in [n]} w_i \widetilde{\Delta}_i \Delta_i \geq \frac{1}{4} C_{\text{prog}}$ for an arbitrary $\Delta = \widetilde{\Delta} - \xi$ where $\left\|\xi_{(m)}\right\|_2 \leq \sqrt{m} C_\xi^{-1}$, as all other conditions are immediate from Definition 19. Note that

$$\begin{aligned}\sum_{i \in [n]} w_i \widetilde{\Delta}_i \Delta_i &= \sum_{i \in [n]} w_i \widetilde{\Delta}_i (\widetilde{\Delta}_i - \xi_i) \\ &\geq \frac{1}{2} \sum_{i \in [n]} w_i \widetilde{\Delta}_i^2 - \frac{1}{2} \sum_{i \in [n]} w_i \xi_i^2 \,.\end{aligned}$$

where we used $a^2 - ab \geq \frac{1}{2} a^2 - \frac{1}{2} b^2$. The first sum above is at least $\frac{1}{2} C_{\text{prog}}$ by assumption. To upper bound the second sum, we will use the second property in the definition of a strong step oracle. Let $S \subset [n]$ be the set consisting of the m largest coordinates of ξ (with ties broken lexicographically). Let α be drawn from the distribution

$$\frac{\delta}{4s\rho^2 \log \frac{d}{\delta}} \text{Multinom}\left(\left\lceil \frac{C_{\text{prog}} C_\xi^2 ns\rho^2 \log \frac{d}{\delta}}{m} \right\rceil, \left(\underbrace{\frac{1}{n}, \ldots, \frac{1}{n}}_{n}\right)\right).$$

Note that with $1 - 0.1\delta$ probability, by a Chernoff bound, we have that $\sum_{i \in S} \alpha_i \geq \frac{1}{5} \delta C_{\text{prog}} C_\xi^2$. If this happens, then since S consists of the largest coordinates of ξ, any vector β such that $\beta \leq \alpha$ entrywise and $\|\beta\|_1 \leq \frac{1}{4} \delta C_{\text{prog}} C_\xi^2$ must have

$$\sum_{i \in [n]} \beta_i \xi_i^2 \leq \frac{5}{4} \sum_{i \in S} \alpha_i \xi_i^2.$$

Now note that for any S with $|S| = m$,

$$\mathbb{E}\left[\sum_{i\in S}\alpha_i\xi_i^2\right] \le \frac{\delta C_{\text{prog}} C_\xi^2}{4m}\cdot\left\|\xi_{(m)}\right\|_2^2 \le \frac{\delta C_{\text{prog}}}{4}.$$

Combining the above two inequalities and Markov's inequality and the fact that the distribution of α stochastically dominates the distribution of w, we deduce that with at least $1-\delta$ probability,

$$\sum_{i\in[n]} w_i\xi_i^2 \le \frac{1}{0.9\delta}\cdot\mathbb{E}\left[\max_{\substack{\beta\le\alpha\\ \|\beta\|_1\le\frac{1}{4}\delta C_{\text{prog}}C_\xi^2}}\sum_{i\in[n]}\beta_i\xi_i^2\right] \le \frac{C_{\text{prog}}}{2}.$$

Putting everything together, we conclude that we have

$$\sum_{i\in[n]} w_i\widetilde{\Delta}_i\Delta_i \ge \frac{C_{\text{prog}}}{4}$$

with failure probability at most 2δ, completing the proof. ■

In Section D.2, we prove that if $\mathbf{A}$ satisfies Assumption 2 (a slightly different assumption than Assumption 1) then with high probability we can implement a strong step oracle with appropriate parameters. This is stated more formally in the following; recall m is defined in (27).

Assumption 2 *The matrix $\mathbf{A}\in\mathbb{R}^{n\times d}$ satisfies the following. There is a weight vector $w^\star\in\Delta^n$ with $\|w^\star\|_\infty \le w^\star_\infty = \frac{1}{m}$, a constants L, $\rho\ge 1$, and constants K, C_ξ (which may depend on L) such that for all $v\in\mathbb{R}^d$, $\xi\in\mathbb{R}^n$ with*

$$\frac{1}{4}\le\|v\|_2\le 1,\ \|v\|_1\le 2\sqrt{2s},\ \left\|\xi_{(m)}\right\|_2\le\frac{\sqrt{m}}{C_\xi}$$

we have, defining $\widetilde{\Delta} = \mathbf{A}v+\xi$:

1. *$\mathbf{A}$ is entrywise bounded by $\pm\rho$, i.e. $\|\mathbf{A}\|_{\max}\le\rho$.*
2. $$\frac{1}{L}\le\sum_{i\in[n]} w_i^\star\widetilde{\Delta}_i^2\le L. \tag{30}$$
3. *There is a $(L,\frac{1}{K\sqrt{s}})$ short-flat decomposition of $\mathbf{A}^\top\mathbf{W}^\star\widetilde{\Delta}$:*
$$\left\|\text{trunc}\left(\mathbf{A}^\top\mathbf{W}^\star\widetilde{\Delta},\frac{1}{K\sqrt{s}}\right)\right\|_2\le L. \tag{31}$$

Lemma 21 *Suppose* $\mathbf{A}$ *satisfies Assumption 2. Algorithm 5 is a* $(C_{\text{prog}}, C_2, C_\xi, \delta)$ *strong step oracle* StrongStepOracle *for* $(\widetilde{\Delta}, \mathbf{A})$ *with*

$$C_{\text{prog}} = \Omega(1),\ C_2 = O(1),\ C_\xi = O(1),\ \delta = \frac{1}{2}\left(\frac{C_2}{10^5 C_{\text{prog}}}\right)^2,$$

running in time

$$O\left(\left(nd\log^3(nd\rho)\log\frac{1}{\delta}\right)\cdot\left(w_\infty^\star s\rho^2\log^2\frac{d}{\delta}\right)\right).$$

Here, in contrast to the noiseless setting, we can only guarantee that the strong step oracle (and thus also the noisy step oracle) succeeds with constant probability. In our full algorithm, we boost the success probability of the oracle by running a logarithmic number of independent trials and aggregating the outputs. We also show that for an appropriate choice of constants in Definition 4, Assumption 2 is also satisfied, stated in Lemma 22 and proven in Section D.3.

Lemma 22 *Suppose* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *is* $(\rho, w_\infty^\star)$*-wRIP with a suitable choice of constants in the RIP parameters in Definition 4. Then,* $\mathbf{A}$ *also satisfies Assumption 2.*

Next, we give a guarantee regarding our geometric post-processing step, Algorithm 4.

Claim 1 Aggregate$(\mathcal{S}, R)$ *runs in* $O(k^2 d)$ *time and meets its output guarantees.*

Proof Let T be the subset of indices $i \in [k]$ such that $\|y_i - z\| \le \frac{R}{3}$. Whenever the algorithm tests y_i for some $i \in T$, it will be returned and satisfies the desired properties. Now consider any y_i returned by the algorithm. The ball of radius $\frac{2R}{3}$ around y_i intersects the ball of radius $\frac{R}{3}$ around z, since otherwise it can only contain at most $0.49k$ points. Thus, $\|y_i - z\|_2 \le R$. The runtime is dominated by the time it takes to do k^2 distance comparisons of points in $\mathbb{R}^d$. ■

We remark that is possible that for $k = \Omega(\log\frac{1}{\delta})$ as is the case in our applications, the runtime of Claim 1 can be improved to have a better dependence on k by subsampling the points and using low-rank projections for distance comparisons.

Lemma 23 *Assume* $\mathbf{A}$ *satisfies Assumption 2. Then, Algorithm 3 meets its output guarantees in time*

$$O\left(\left(nd\log^3(nd\rho)\right)\cdot\left(w_\infty^\star s\rho^2\log d\right)\cdot\log^2\frac{d}{\delta}\right).$$

Proof We claim that for each independent trial $j \in [N_{\text{trials}}]$, except with probability $1 - T\delta'$, the output x_T^j satisfies $\|x_T^j - x^\star\|_2 \le \frac{R}{6}$. Once we prove this, by Chernoff at least $0.51N_{\text{trials}}$ of the trials satisfy $\|x_T^j - x^\star\|_2 \le \frac{R}{6}$ except with probability at most δ, and then we are done by Claim 1.

It remains to prove the above claim. Fix a trial j, and drop the superscript j for notational convenience. In every iteration t, $\widetilde{\Delta} := \frac{1}{R}(\mathbf{A}x_t - b)$ is given to $\mathcal{O}_{\text{nstep}}$. Since $b = \mathbf{A}x^\star + \xi$, we have

$$\widetilde{\Delta} = \frac{1}{R}(\mathbf{A}(x - x^\star) + \xi) = \mathbf{A}v + \xi,$$

Algorithm 3: HalfRadiusSparseNoisy$(x_{\text{in}}, R, R_\xi, \mathcal{O}_{\text{nstep}}, \delta, \mathbf{A}, b)$

1 **Input:** s-sparse $x_{\text{in}} \in \mathbb{R}^d$, $R \geq \|x_{\text{in}} - x^\star\|_2$ for s-sparse $x^\star \in \mathbb{R}^d$, $(C_{\text{prog}}, C_2, C_\xi, \delta')$-noisy step oracle $\mathcal{O}_{\text{nstep}}$ for all $(\Delta, \mathbf{A})$ with $\Delta \in \mathbb{R}^n$, $\delta' \leq (10^{-4}\frac{C_{\text{prog}}}{C_2})^2$, $\mathbf{A} \in \mathbb{R}^{n\times d}$, $b = \mathbf{A}x^\star + \xi \in \mathbb{R}^n$ for $\left\|\xi_{(m)}\right\|_2 \leq R_\xi\sqrt{m}$, with $R \geq C_\xi R_\xi$;

2 **Output:** s-sparse vector x_{out} that satisfies $\|x_{\text{out}} - x^\star\|_2 \leq \frac{1}{2}R$ with probability $\geq 1-\delta$

3 $x_0 \leftarrow x_{\text{in}}$, $\mathcal{X} \leftarrow \{x \in \mathbb{R}^d \mid \|x - x_{\text{in}}\|_1 \leq \sqrt{2s}R\}$;

4 $T \leftarrow \left\lceil \frac{200C_2^2}{C_{\text{prog}}^2} \right\rceil$, $\eta \leftarrow \frac{C_{\text{prog}}}{2C_2^2}$;

5 $N_{\text{trials}} \leftarrow 10\log\frac{d}{\delta}$;

6 **for** $1 \leq j \leq N_{\text{trials}}$ **do**

7 $\quad x_0^j \leftarrow x_0$;

8 $\quad$ **for** $0 \leq t \leq T-1$ **do**

9 $\quad\quad w_t^j \leftarrow \mathcal{O}_{\text{nstep}}(\Delta_t^j, \mathbf{A})$ for $\Delta_t^j \leftarrow \frac{1}{R}(\mathbf{A}x_t^j - b)$, $\gamma_t^j \leftarrow \mathbf{A}^\top \mathbf{diag}\left([w_t^j]\right)\Delta_t^j = \sum_{i\in[n]}[w_t^j]_i[\Delta_t^j]_i a_i$;

10 $\quad\quad$ **if** (w_t^j, γ_t^j) *do not meet all of* (3), (4) *and the additional criteria in Definition 19* **then**

11 $\quad\quad\quad x_T^j \leftarrow [x_t^j]_{(s)}$;

12 $\quad\quad\quad$ **Break:** ;

13 $\quad\quad$ **end**

14 $\quad\quad x_{t+1}^j \leftarrow \text{argmin}_{x\in\mathcal{X}} \left\|x - x_t^j - \eta R\gamma_t\right\|_2$;

15 $\quad$ **end**

16 **end**

17 $x_T \leftarrow$ Aggregate$(\{x_T^1, \ldots, x_T^{N_{\text{trials}}}\}, \frac{R}{2})$;

18 **Return:** $x_{\text{out}} \leftarrow [x_T]_{(s)}$;

Algorithm 4: Aggregate$(\mathcal{S}, R)$

1 **Input:** $\mathcal{S} = \{y_i\}_{i\in[k]} \subset \mathbb{R}^d$, $R \geq 0$ such that for some unknown $z \in \mathbb{R}^d$, at least $0.51k$ points $y_i \in \mathcal{S}$ have $\|y_i - z\|_2 \leq \frac{R}{3}$;

2 **Output:** $\widetilde{z}$ with $\|\widetilde{z} - z\|_2 \leq R$;

3 **for** $1 \leq i \leq k$ **do**

4 $\quad$ **if** *at least* $0.51k$ *points* $y_j \in \mathcal{S}$ *satisfy* $\|y_i - y_j\|_2 \leq \frac{2R}{3}$ **then** **Return:** $\widetilde{z} \leftarrow y_i$;

5 **end**

for $\|v\|_2 \leq 1$, $\|v\|_1 \leq 2\sqrt{2s}$, and $\|\xi\|_{2,(m)} \leq \frac{\sqrt{m}}{C_\xi}$, where the last inequality used the assumed bounds

$$\left\|\xi_{(m)}\right\|_2 \leq R_\xi\sqrt{m}, \ \frac{R_\xi}{R} \leq \frac{1}{C_\xi}.$$

Hence, by the assumptions on $\mathcal{O}_{\text{nstep}}$, it will not fail for such inputs unless $\|v\|_2 \geq \frac{1}{12}$ is violated, except with probability $\leq \delta'$. If the check in Line 10 fails, then except with probability $\leq \delta'$, the conclusion $\|x_T - x^\star\|_2 \leq \frac{R}{6}$ follows analogously to Lemma 9, since $v = \frac{1}{R}(x - x^\star)$.

The other case's correctness follows identically to the proof of Lemma 9, except for one difference: to lower bound the progress term (10), we use the assumption (28) which shows

$$2\eta R \langle \gamma_t, x_t - x^\star \rangle = 2\eta R \sum_{i\in[n]} w_i \widetilde{\Delta}_i \langle a_i, v \rangle = 2\eta R^2 \sum_{i\in[n]} w_i \widetilde{\Delta}_i \Delta_i \geq 2\eta R^2 C_{\text{prog}}.$$

Hence, following the proof of Lemma 9 (and adjusting for constants), whenever the algorithm does not terminate we make at least a $\frac{50}{T}$ fraction of the progress towards $x^\star$, so in T iterations (assuming no step oracle failed) we will have $\|x_T - x^\star\|_2 \leq \frac{R}{6}$.

Finally, the runtime follows from combining Lemma 21 (with constant failure probability) with a multiplicative overhead of $T \cdot N_{\text{trials}}$ due to the number of calls to the step oracle, contributing one additional logarithmic factor. We adjusted one of the $\log d$ terms to become a $\log \frac{d}{\delta}$ term to account for the runtime of Aggregate (see Claim 1). ■

D.2. Designing a strong step oracle

In this section, we design a strong step oracle $\mathcal{O}_{\text{step}}(\widetilde{\Delta}, \mathbf{A})$ under Assumption 2. As in Section C.2, our oracle iteratively builds a weight vector $\bar{w}$, and sets

$$\gamma_{\bar{w}} := \sum_{i\in[n]} \bar{w}_i \widetilde{\Delta}_i a_i.$$

We will use essentially the same potentials as in (13), defined in the following:

$$\widetilde{\Phi}_2(\bar{w}) := \sum_{i\in[n]} \bar{w}_i \widetilde{\Delta}_i^2, \ \widetilde{\Phi}_{\text{sqmax}}(\bar{w}) := \left(\min_{\|p\|_2 \leq L\|\bar{w}\|_1} \text{sqmax}_\mu(\gamma_{\bar{w}} - p) \right) + \frac{\|\bar{w}\|_1}{4CLs}. \tag{32}$$

Algorithm 5 is essentially identical to Algorithm 2 except for changes in constants. We further have the following which verifies the second property in the definition of strong step oracle.

Fact 2 *The distribution of w returned by Algorithm 5 is stochastically dominated by the distribution*

$$\eta w_\infty^* \text{Multinom}\left(\frac{5Ln}{\eta}, \left(\underbrace{\frac{1}{n}, \ldots, \frac{1}{n}}_{n} \right) \right)$$

Proof Every time we inspect a row, we change the corresponding entry of w by at most ηw_∞^*. The result follows from the number of iterations in the algorithm and uniformity of sampling rows. ■

To analyze Algorithm 5, we provide appropriate analogs of Lemmas 11 and 7. Because Algorithm 5 is very similar to Algorithm 2, we will largely omit the proof of the following statement, which follows essentially identically to the proof of Lemma 11 up to adjusting constants.

Lemma 24 *Assume that the constant K in Assumption 2 is sufficiently large, and that $\widetilde{\Delta} = \mathbf{A}v + \xi$ where v, ξ satisfy the norm conditions in Assumption 2. Then for any $\bar{w} \in \mathbb{R}^n_{\geq 0}$ such that $B(\bar{w}) \leq C^2\mu^2 \log d$, we have*

$$\mathbb{E}_{i\sim_{\text{unif.}}[n]}[B(\bar{w} + \eta w_i^\star)] \leq B(\bar{w}) + \frac{1}{2CLs} \cdot \frac{\eta}{n}.$$

Algorithm 5: StrongStepOracle($\widetilde{\Delta}, \mathbf{A}, \delta$)

Input: $\widetilde{\Delta} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times d}$ satisfying Assumption 1, $\delta \in (0,1)$;

Output: (w, γ) such that $\gamma = \sum_{i \in [n]} w_i \widetilde{\Delta}_i a_i$, and if there is $v \in \mathbb{R}^d$ with $\frac{1}{12} \leq \|v\|_2 \leq 1$ such that $\widetilde{\Delta} = \mathbf{A}v + \xi$ where $\left\|\xi_{(m)}\right\|_2 \leq \frac{\sqrt{m}}{C_\xi}$, with probability $\geq 1 - \delta$, (3), (4) are satisfied with

$$C_{\text{prog}} = 1, \; C_2 = O(1).$$

Furthermore, the second condition in (4) is satisfied with the constant 24 rather than 6, and there is $C_\xi = O(1)$ such that (29) is also satisfied.

$C \leftarrow 3200$, $\mu \leftarrow \frac{1}{\sqrt{Cs \log d}}$, $\eta \leftarrow \frac{1}{K w_\infty^\star s \rho^2 \log d}$, $N' \leftarrow \lceil \log_2 \frac{2}{\delta} \rceil$;

for $0 \leq k \leq N'$ **do**

  $w_0 \leftarrow 0_n$, $N \leftarrow \lceil \frac{5Ln}{\eta} \rceil$;

  for $0 \leq t \leq N$ **do**

    if $\widetilde{\Phi}_2(w_t) \geq 1$ **then Return:** $\gamma \leftarrow \sum_{i \in [n]} [w_t]_i \widetilde{\Delta}_i a_i$, $w \leftarrow w_t$;

    Sample $i \sim_{\text{unif.}} [n]$;

    Compute (using Lemma 10) $d_t \in [0, \eta w_\infty^\star]$ maximizing to additive $O(\frac{\eta}{n})$

$$\Gamma_t(d) := \widetilde{\Phi}_2(w_t + de_i) - Cs\widetilde{\Phi}_{\text{sqmax}}(w_t + de_i)$$

    $w_{t+1} \leftarrow w_t + d_t e_i$;

  end

end

Return: $\gamma \leftarrow 0_d$, $w \leftarrow 0_n$

Proof The analysis is essentially identical to that of Lemma 11; we discuss only the main difference. To apply the Taylor expansion of the exponential, Lemma 11 required a bound that

$$\frac{1}{\mu}\left|z_j^{(i)}\right| = O\left(\frac{1}{\sqrt{\log d}}\right) \Longleftarrow \left|z_j^{(i)}\right| = O\left(\frac{1}{\sqrt{s}\log d}\right),$$

for all $i \in [n]$ and $j \in [d]$. Note that in the setting of Lemma 11, we took $z_j^{(i)} = \eta w_i^\star(\Delta_i a_{ij} - p_j^\star)$. Here, we will take $z_j^{(i)} = \eta w_i^\star(\widetilde{\Delta}_i a_{ij} - p_j^\star)$; bounds on all of these terms follow identically to in Lemma 11, except that $\widetilde{\Delta}_i = \langle a_i, v\rangle + \xi_i$, so we need to show

$$\eta w_i^\star \rho |\xi_i| = O\left(\frac{1}{\sqrt{s}\log d}\right).$$

This follows since $\eta \leq \frac{1}{\log d}$ and $|\xi_i| = O(\sqrt{m})$ by assumption. Hence, as $w_i^\star \leq \frac{1}{m}$ by definition of m, this is equivalent to $m = \Omega(s\rho^2)$, an explicit assumption we make. ■

We now give a full analysis of Algorithm 5, patterned off of Lemma 7.

Lemma 25 *Suppose* $\mathbf{A}$ *satisfies Assumption 2. Algorithm 5 is a* $(C_{\text{prog}}, C_2, C_\xi, \delta)$ *strong step oracle* StrongStepOracle *for* $(\widetilde{\Delta}, \mathbf{A})$ *with*

$$C_{\text{prog}} = \Omega(1),\ C_2 = O(1),\ C_\xi = O(1),\ \delta = \frac{1}{2}\left(\frac{C_2}{10^5 C_{\text{prog}}}\right)^2,$$

running in time

$$O\left(\left(nd\log^3(nd\rho)\log\frac{1}{\delta}\right)\cdot\left(w_\infty^\star s\rho^2\log^2\frac{d}{\delta}\right)\right).$$

Proof The analysis is essentially identical to that of Algorithm 2 in Lemma 7; we discuss differences here. First, the stochastic domination condition follows from Fact 2 for sufficiently large C_ξ, K.

For the remaining properties, since the algorithm runs $N' \geq \log_2 \frac{2}{\delta}$ times independently, it suffices to show each run meets Definition 19 with probability $\geq \frac{1}{2}$ under the events of Assumption 2, assuming there exists the desired decomposition $\widetilde{\Delta} = \mathbf{A}v + \xi$ in the sense of Assumption 2. Union bounding with the failure probability in Fact 2 yields the overall failure probability.

Correctness. As in Lemma 7, it is straightforward to see that $\widetilde{\Phi}_2$ is 1-Lipschitz, since the value of η is smaller than that used in Algorithm 2. The termination condition in iteration T then again implies $\widetilde{\Phi}_2(w_T) \geq 1$, and $\widetilde{\Phi}_{\text{sqmax}}(w_T) \leq \frac{3}{Cs}$. For $C = 3200$, this implies the short-flat decomposition with stronger parameters required by Definition 19, as well as the $\|w_T\|_1$ bound.

Success probability. As in Lemma 7, the expected growth in Φ_t in any iteration where $\Pr[\widetilde{\Phi}_2(w_t) \geq 1] \leq \frac{1}{2}$ is $\geq \frac{\eta}{4Ln}$. Hence, running for $\geq \frac{5Ln}{\eta}$ iterations and using $\Phi_t - \Phi_0 \leq 2$ yields the claim.

Runtime. This follows identically to the analysis in Lemma 7.

■

D.3. Equivalence between Assumption 2 and RIP

In this section, we prove Lemma 22, restated here for completeness. The proof will build heavily on our previous developments in the noiseless case, as shown in Section C.3.

Lemma 26 *Suppose* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *is* $(\rho, w_\infty^\star)$*-wRIP with a suitable choice of constants in the RIP parameters in Definition 4. Then,* $\mathbf{A}$ *also satisfies Assumption 2.*

Proof The analysis is largely similar to the analysis of Lemma 15; we will now discuss the differences here, which are introduced by the presence of the noise term ξ. There are three components to discuss: the upper and lower bounds in (30), and the decomposition (31).

Regarding the bounds in (5), by changing constants appropriately in Definition 4, we can assume that $\mathbf{A}$ satisfies the second property in Assumption 1 with the parameters $\frac{4}{L}$ and $\frac{L}{4}$. In particular, for $\Delta = \mathbf{A}v$, we then have

$$\frac{4}{L} \leq \sum_{i\in[n]} w_i^\star \Delta_i^2 \leq \frac{L}{4}.$$

Recall that $\widetilde{\Delta} = \Delta + \xi$ for some $\left\|\xi_{(m)}\right\|_2 \leq \frac{\sqrt{m}}{C_\xi}$. Hence,

$$\sum_{i\in[n]} w_i^\star \widetilde{\Delta}_i^2 \leq 2\sum_{i\in[n]} w_i^\star \Delta_i^2 + 2\sum_{i\in[n]} w_i^\star \xi_i^2$$
$$\leq \frac{L}{2} + 2\left(\frac{1}{m}\left\|\xi_{(m)}\right\|_2^2\right) \leq L,$$

for an appropriately large $C_\xi^2 \geq \frac{4}{L}$. Here the first inequality used $(a+b)^2 \leq 2a^2 + 2b^2$, and the second inequality used that the largest $\sum_{i\in[n]} w_i^\star \xi_i^2$ can be subject to $\|w^\star\|_1 = 1$ and $\|w^\star\|_\infty \leq \frac{1}{m}$ is attained by greedily choosing the m largest coordinates of ξ by their magnitude, and setting $w_i^\star = \frac{1}{m}$ for those coordinates. This gives the upper bound in Assumption 2, and the lower bound follows similarly: for appropriately large $C_\xi^2 \geq \frac{L}{2}$,

$$\sum_{i\in[n]} w_i^\star \widetilde{\Delta}_i^2 \geq \frac{1}{2}\sum_{i\in[n]} w_i^\star \Delta_i^2 - \frac{1}{2}\sum_{i\in[n]} w_i^\star \xi_i^2$$
$$\geq \frac{2}{L} - \frac{1}{2}\left(\frac{1}{m}\left\|\xi_{(m)}\right\|_2^2\right) \geq \frac{1}{L}.$$

Lastly, for the decomposition required by (31), we will use the decomposition of Lemma 8 for the component due to $\sum_{i\in[n]} w_i^\star \Delta_i a_i$; in particular, assume by adjusting constants that this component has a $(\frac{L}{2}, \frac{1}{2K\sqrt{s}})$ short-flat decomposition. It remains to show that

$$\sum_{i\in[n]} w_i^\star \xi_i a_i = \mathbf{A}^\top \mathbf{W}^\star \xi.$$

also admits a $(\frac{L}{2}, \frac{1}{2K\sqrt{s}})$ short-flat decomposition, at which point we may conclude by the triangle inequality. Let $u = (\mathbf{W}^\star)^{\frac{1}{2}}\xi$; from earlier, we bounded

$$\|u\|_2^2 \leq \frac{1}{m}\left\|\xi_{(m)}\right\|_2^2 \implies \|u\|_2 \leq \frac{1}{C_\xi}.$$

Hence, applying Lemma 14 using the RIP matrix $(\mathbf{W}^\star)^{\frac{1}{2}}\mathbf{A}$ with appropriate parameters yields the conclusion, for large enough C_ξ. In particular, the ℓ_2-bounded part of the decomposition follows from Lemma 14, and the proof of the ℓ_∞-bounded part is identical to the proof in Lemma 15. ■

D.4. Putting it all together

We now prove our main result on noisy recovery.

Theorem 16 *Let $\delta \in (0,1)$, and suppose $R_0 \geq \|x^\star\|_2$ for s-sparse $x^\star \in \mathbb{R}^d$. Further, suppose $\mathbf{A} \in \mathbb{R}^{n\times d}$ is $(\rho, w_\infty^\star)$-wRIP and $b = \mathbf{A}x^\star + \xi$, and $R_1 \geq R_\xi := \left\|\xi_{(m)}\right\|_2$. Then with probability at least $1-\delta$, Algorithm 3 using Algorithm 3 as a noisy step oracle computes $\hat{x}$ satisfying*

$$\|\hat{x} - x^\star\|_2 \leq R_{\text{final}} = \Theta(R_\xi),$$

in time

$$O\left(\left(ndw_\infty^\star s\log^4(nd\rho)\log^2\left(\frac{d}{\delta}\cdot\log\left(\frac{R_0}{R_{\text{final}}}\right)\log\left(\frac{R_1}{R_{\text{final}}}\right)\right)\right)\cdot\rho^2\log\left(\frac{R_0}{R_{\text{final}}}\right)\log\left(\frac{R_1}{R_{\text{final}}}\right)\right).$$

Proof Our algorithm will iteratively maintain a guess R_{guess} on the value of $\frac{1}{\sqrt{m}}\left\|\xi_{(m)}\right\|_2$, initialized at $R_{\text{guess}} \leftarrow R_1$. For each value of $R_{\text{guess}} \geq R_\xi$, the hypothesis of Algorithm 3 is satisfied, and hence using a strategy similar to the proof of Theorem 5 (but terminating at accuracy $R = O(R_{\text{guess}})$ where the constant is large enough to satisfy the assumption $R \geq C_\xi R_{\text{guess}}$) results in an estimate at distance R with probability at least $1-\delta$, with runtime

$$O\left(\left(ndw_\infty^\star s\rho^2 \log^4(nd\rho)\log^2\left(\frac{d}{\delta}\cdot\log\left(\frac{R_0}{R_{\text{final}}}\right)\right)\right)\cdot\rho^2\log\left(\frac{R_0}{R_{\text{final}}}\right)\right).$$

The runtime above follows from Lemma 23.

Our overall algorithm repeatedly halves R_{guess}, and outputs the last point returned by a run of the algorithm where it can certify a distance bound to $x^\star$ of $R = C_\xi R_{\text{guess}}$. We use R_{final} to denote $C_\xi R_{\text{guess}}$ on the last run. Clearly for any $R_{\text{guess}} \geq R_\xi$ this certification will succeed, so we at most lose a factor of 2 in the error guarantee as we will have $R_{\text{final}} \leq 2C_\xi R_\xi$. The final runtime follows from adjusting δ by a factor of $O(\log \frac{R_1}{R_{\text{final}}})$ to account for the multiple runs of the algorithm. ■

Appendix E. Greedy and non-convex methods fail in the semi-random setting

In this section, we show how a few standard, commonly-used non-convex or greedy methods can fail (potentially quite drastically) in the semi-random adversary setting. The two algorithms that we examine are Iterative Hard Thresholding and Orthogonal Matching Pursuit [15, 76]. We believe it is likely that similar counterexamples can be constructed for other, more complex algorithms such as CoSaMP [66]. For simplicity in this section, we will only discuss the specific semi-random model introduced in Definition 1, where $\mathbf{A}$ is pRIP, i.e. it contains an unknown RIP matrix $\mathbf{G}$ as a subset of its rows.

E.1. Iterative hard thresholding

The iterative hard thresholding algorithm [15] involves initializing $x_0 = 0$ and taking

$$x_{t+1} = H_s\left(x_t - \frac{1}{n}\mathbf{A}^\top(b - \mathbf{A}x_t)\right)$$

where H_s zeroes out all but the s largest entries in magnitude (ties broken lexicographically). We can break this algorithm in the semi-random setting by simply duplicating one row many times.

Hard semi-random adversary. Let $n = Cm$ for some sufficiently large constant C. The first m rows of $\mathbf{A}$ are drawn independently from $\mathcal{N}(0, \mathbf{I})$. Now draw $v \sim \mathcal{N}(0, \mathbf{I})$, except set the first entry of v to 1. We set the last $(C-1)m$ rows of $\mathbf{A}$ all equal to v. We will set the sparsity parameter $s = 1$ and let $x^\star = (1, 0, \ldots, 0)$. We let $b = \mathbf{A}x^\star$.

Proposition 27 *With $\mathbf{A}, b$ generated as above, with high probability, iterative hard thresholding does not converge.*

Proof With high probability, some coordinate of v is $\Omega(\sqrt{\log d})$. We then have that some entry of $\mathbf{A}^\top b$ has magnitude at least $\Omega(n\sqrt{\log d})$ with high probability. Thus, the next iterate x_1 must have exactly one nonzero entry that has magnitude at least $\Omega(\sqrt{\log d})$ and furthermore, this entry must

correspond to some coordinate of v that has magnitude at least $\Omega(\sqrt{\log d})$. However, this means that the residuals in all of the rows that are copies of v are at least $\Omega(\log d)$. In the next step, by the same argument, we get that the residuals blow up even more and clearly this algorithm will never converge. In fact, x_t will never have the right support because its support will always be on one of the entries where v is large. ■

E.2. Orthogonal matching pursuit

The orthogonal matching pursuit algorithm [76] involves initializing $x_0 = 0$ and keeping track of a set S (that corresponds to our guess of the support of $x^\star$). Each iteration, we choose a column c_j of $\mathbf{A}$ that maximizes $\frac{|\langle c_j, r_t\rangle|}{\|c_j\|_2^2}$ and then add j to S (where $r_t = \mathbf{A}x_t - b$ is the residual). We then add j to S and project the residual onto the orthogonal complement of all coordinates in S. We show that we can again very easily break this algorithm in the semi-random setting.

Hard semi-random adversary. Let $n = 3m$. First, we draw all rows of $\mathbf{A}$ independently from $\mathcal{N}(0, \mathbf{I})$. Next, we modify some of the entries in the last $2m$ rows. Let s be the sparsity parameter. Let $x^\star = (s^{-\frac{1}{2}}, \ldots, s^{-\frac{1}{2}}, 0, \ldots, 0)$ be supported on the first s coordinates and set $b = \mathbf{A}x^\star$. Now we modify the columns of $\mathbf{A}$ (aside from the first s so $\mathbf{A}x^\star$ is not affected). We set the last $2m$ entries of one of these columns c_j to match those of b.

Proposition 28 *With $\mathbf{A}, b$ generated as above, with high probability, orthogonal matching pursuit does not recover $x^\star$.*

Proof With high probability (as long as $s \geq 10$), the column c_j is the one that maximizes $\frac{|\langle c_j, b\rangle|}{\|c_j\|_2^2}$ because its last $2m$ entries exactly match those of b. However, j is not in the support of $x^\star$ so the algorithm has already failed. ■

We further make the following observation.

Remark 29 *By modifying other columns of $\mathbf{A}$ as well, the semi-random adversary can actually make the algorithm pick all of the wrong columns in the support.*

E.3. Convex methods

Now we briefly comment on how convex methods are robust, in the sense that they can still be used in the semi-random setting (but may have substantially slower rates than their fast counterparts). In the noiseless observations case, this is clear because the additional rows of $\mathbf{A}$ are simply additional constraints that are added to the standard ℓ_1 minimization convex program.

In the noisy case, let the target error be $\theta = \left\|\xi_{(m)}\right\|_2$. We then solve the modified problem

$$\begin{aligned} &\min \|x\|_1 \\ &\text{subject to } \left\|[\mathbf{A}x - b]_{(m)}\right\|_2 \leq \theta. \end{aligned}$$

Note that the above is a convex program and thus can be solved in polynomial time by e.g. cutting plane methods [46]. Also, note that $x^\star$ is indeed feasible for the second constraint. Now for the solution $\widehat{x}$ that we obtain, we must have $\|\widehat{x}\|_1 \leq \|x^\star\|_1$ and

$$\left\|[\mathbf{A}(x^\star - \widehat{x})]_{(m)}\right\|_2 \leq 2\theta.$$

Let $\mathbf{G}$ be the set of m randomly generated rows of $\mathbf{A}$ under our semi-random adversarial model. The previous two conditions imply

- $\|\widehat{x} - x^\star\|_1 \leq 2\sqrt{s}\,\|x^\star - \widehat{x}\|_2$
- $\|\mathbf{G}(x^\star - \widehat{x})\|_2 \leq 2\theta$

which now by restricted strong convexity of $\mathbf{G}$ (see [1]) implies that $\|x^\star - \widehat{x}\|_2 = O(\frac{\theta}{\sqrt{m}})$. We can furthermore round $\widehat{x}$ to s-sparse to obtain the sparse vector x', and the above bound only worsens by a factor of 2 for x' (see Lemma 9 for this argument).

Appendix F. Deferred proofs

Lemma 30 *Let $\delta > 0$ and $\theta \geq 0$. For any vector $\gamma \in \mathbb{R}^d$, we can solve the optimization problem*

$$\min_{\|p\|_2 \leq \theta} \text{sqmax}_\mu(\gamma - p)$$

to additive accuracy δ in time

$$O\left(d \log^2\left(\frac{\|\gamma\|_2^2}{\mu\sqrt{\delta}}\right)\right).$$

Proof Let $\mathcal{P} \subset \mathbb{R}^d$ be the set of p such that p has the same sign as γ entrywise and $|p_j| \leq |\gamma_j|$ for all $j \in [d]$. By symmetry of the sqmax and the ℓ_2 norm under negation, the optimal p lies in $\mathcal{P}$.

Next we claim that the function $\text{smax}_\mu(\gamma - p)$ is $2\|\gamma\|_2$-Lipschitz in the ℓ_2 norm as a function of p, over $\mathcal{P}$. To see this, the gradient is directly computable as

$$2(p - \gamma) \circ x \text{ where } x \in \Delta^d \text{ with } x_i = \frac{\exp([\gamma_i - p_i]^2/\mu^2)}{\sum_{j\in[n]} \exp([\gamma_j - p_j]^2/\mu^2)} \text{ for all } i \in [n]$$

where $\circ$ denotes entrywise multiplication. Thus, the ℓ_2 norm of the derivative is bounded by $2\|\gamma\|_2$ over $\mathcal{P}$. In the remainder of the proof, we show how to find $p \in \mathcal{P}$ which has ℓ_2 error $\frac{\delta}{2\|\gamma\|_2}$ to the optimal, which implies by Lipschitzness that the function value is within additive δ of optimal.

Next, since $0 \in \mathcal{P}$, we may assume without loss of generality that

$$\theta > \frac{\delta}{2\|\gamma\|_2}. \tag{33}$$

else we may just output 0, which achieves optimality gap at most $2\|\gamma\|_2\,\theta$.

Now, by monotonicity of $\ln$ it suffices to approximately minimize

$$\sum_{j\in[d]} \exp\left(\frac{[\gamma - p]_j^2}{\mu^2}\right).$$

The sum above is always at least d. First we check if $\|\gamma\|_2 \leq \theta + \sqrt{\delta}$. If this is true then clearly we can set p so that all entries of $\gamma - p$ have magnitude at most $\sqrt{\delta}$. This gives a solution such that

$$\text{sqmax}_\mu(\gamma - p) \leq \mu^2 \log\left(d \exp\left(\frac{\delta}{\mu^2}\right)\right) = \mu^2 \log d + \delta$$

and since the value of sqmax is always at least $\mu^2 \log d$, this solution is optimal up to additive error δ. Thus, we can assume $\|\gamma\|_2 \geq \theta + \sqrt{\delta}$ in the remainder of the proof. We also assume all entries of γ are nonzero since if an entry of γ is 0 then the corresponding entry of p should also be 0. Finally by symmetry of the problem under negation we will assume all entries of γ are positive in the remainder of the proof, such that each entry of p is also positive.

By monotonicity of sqmax in each coordinate (as long as signs are preserved) and the assumption that $\|\gamma\|_2 \geq \theta + \sqrt{\delta}$, the optimal solution must have $\|p\|_2 = \theta$. By using Lagrange multipliers, for some scalar ζ and all j,

$$p_j = \exp(\zeta) \cdot [\gamma - p]_j \exp\left(\frac{[\gamma - p]_j^2}{\mu^2}\right). \tag{34}$$

For the optimal ζ by taking ℓ_2 norms of the quantity above, we have

$$\theta = \|p\|_2 = \zeta \|\gamma - p\|_2 \cdot C \text{ for some } C \in \left[0, \exp\left(\frac{\|\gamma\|_2^2}{\mu^2}\right)\right].$$

Hence taking logarithms of both sides and using both the bounds (33) and $\|\gamma - p\|_2 \geq \sqrt{\delta}$ at the optimum, which follows from the previous discussion, we obtain

$$\log \frac{\theta}{\|\gamma - p\|_2} - \zeta \in \left[0, \frac{\|\gamma\|_2^2}{\mu^2}\right] \implies \zeta \in \left[-\frac{\|\gamma\|_2^2}{\mu^2} - \log\left(\frac{2\|\gamma\|_2^2}{\delta}\right), \log\left(\frac{\|\gamma\|_2}{\sqrt{\delta}}\right)\right].$$

We next show how to compute p to high accuracy given a guess on ζ. Observe that if $\gamma_j > 0$, then the right-hand side of (34) is decreasing in p_j and hence by the intermediate value theorem, there is a unique solution strictly between 0 and γ_j for any ζ. Also, note that the location of this solution increases with ζ. Let $p(\zeta)$ be the solution obtained by exactly solving (34) for some given ζ. We have shown for all ζ that $0 \leq [p(\zeta)]_j \leq \gamma_j$ entrywise and hence $\|p(\zeta)\|_2 \leq \|\gamma\|_2$ for all ζ.

For a fixed ζ, we claim we can estimate $p(\zeta)$ to ℓ_2 error β in time $O(d \log \frac{\|\gamma\|_2}{\beta})$. To see this, fix some ζ, μ, and γ_j, and consider solving (34) for the fixed point p_j. We can discretize $[0, \gamma_j]$ into intervals of length $\frac{\gamma_j \beta}{\|\gamma\|_2}$ and perform a binary search. The right-hand side is decreasing in p_j and the left-hand side is increasing so the binary search yields some interval of length $\frac{\gamma_j \beta}{\|\gamma\|_2}$ containing the fixed point p_j via the intermediate value theorem. The resulting ℓ_2 error along all coordinates is then β. We also round this approximate $p(\zeta)$ entrywise down in the above search to form a vector $\tilde{p}(\zeta, \beta)$ such that $\tilde{p}(\zeta, \beta) \leq p(\zeta)$ entrywise and $\|\tilde{p}(\zeta, \beta) - p(\zeta)\|_2 \leq \beta$. We use this notation and it is well-defined as the search is deterministic.

In the remainder of the proof we choose the constants

$$\alpha := \frac{\delta^2}{192 \|\gamma\|_2^4}, \ \beta := \min\left(\frac{\delta^2}{192 \|\gamma\|_2^3}, \frac{\delta}{4 \|\gamma\|_2}\right).$$

We define $\tilde{p}(\zeta) := \tilde{p}(\zeta, \beta)$ for short as β will be fixed. Discretize the range $[-\frac{\|\gamma\|_2^2}{\mu^2} - \log \frac{2\|\gamma\|_2^2}{\delta}, \log \frac{\|\gamma\|_2}{\sqrt{\delta}}]$ into a grid of uniform intervals of length α. Consider the ζ such that $\zeta \leq \zeta^\star < \zeta + \alpha$. Because

$p(\zeta^\star)$ is entrywise larger than $p(\zeta)$ and hence the logarithmic term on the right-hand side of (34) is smaller for $p(\zeta^\star)$ than $p(\zeta)$, we have

$$[p(\zeta)]_j \le [p(\zeta^\star)]_j \le \exp(\alpha)\,[p(\zeta)]_j\,.$$

Moreover the optimal $p(\zeta^\star)$ has ℓ_2 norm θ, so $|\zeta - \zeta^\star| \le \alpha$ and $\exp(\alpha) - 1 \le 2\alpha$ imply

$$\|p(\zeta) - p(\zeta^\star)\|_2 \le 2\alpha\,\|p(\zeta^\star)\|_2 \le 2\alpha\,\|\gamma\|_2 \le \Delta := \frac{\delta^2}{96\,\|\gamma\|_2^3}.$$

Consider the algorithm which returns the ζ_{alg} on the search grid which minimizes $|\|\tilde{p}(\zeta_{\text{alg}})\|_2 - \theta|$ (we will discuss computational issues at the end of the proof). As we have argued above, there is a choice which yields $\|p(\zeta)\|_2 \in [\theta - \Delta, \theta + \Delta]$ and hence

$$\left\|\tilde{p}(\zeta_{\text{alg}})\right\|_2 \in [\theta - \Delta - \beta, \theta + \Delta + \beta]\,. \tag{35}$$

We next claim that

$$\left\|p(\zeta_{\text{alg}}) - p(\zeta^\star)\right\|_2 \le \frac{\delta}{4\,\|\gamma\|_2}. \tag{36}$$

Suppose (36) is false and $\zeta_{\text{alg}} > \zeta^\star$. Then letting $u := p(\zeta_{\text{alg}})$ and $v := p(\zeta^\star)$, note that u, v, and $u - v$ are all entrywise nonnegative and hence

$$\|u\|_2^2 \ge \|v\|_2^2 + \sum_{i\in[n]} 2u_i(u_i - v_i) + (u_i - v_i)^2 > \|v\|_2^2 + \left(\frac{\delta}{4\,\|\gamma\|_2}\right)^2.$$

Hence, we have by $\sqrt{x^2 + y^2} \ge x + \frac{y^2}{3x}$ for $0 \le y \le x$, (33), and $\theta \le \|\gamma\|_2$,

$$\left\|p(\zeta_{\text{alg}})\right\|_2 = \|u\|_2 > \|v\|_2 + \frac{\left(\frac{\delta}{4\|\gamma\|_2}\right)^2}{3\,\|v\|_2} \ge \theta + \frac{\delta^2}{48\,\|\gamma\|_2^3} \ge \theta + \Delta + 2\beta.$$

So, by triangle inequality $\left\|\tilde{p}(\zeta_{\text{alg}})\right\|_2 > \theta + \Delta + \beta$ and hence we reach a contradiction with (35).

Similarly, suppose (36) is false and $\zeta_{\text{alg}} < \zeta^\star$. Then for the same definitions of u, v, and using the inequality $\sqrt{x^2 - y^2} \le x - \frac{y^2}{3x}$ for $0 \le y \le x$, we conclude

$$\|v\|_2^2 > \|u\|_2^2 + \left(\frac{\delta}{4\,\|\gamma\|_2}\right)^2 \implies \|u\|_2 \le \sqrt{\|v\|_2^2 - \left(\frac{\delta}{4\,\|\gamma\|_2}\right)^2} < \theta - \Delta - 2\beta.$$

So we reach a contradiction with (35) in this case as well.

In conclusion, (36) is true and we obtain by triangle inequality the desired

$$\left\|\tilde{p}(\zeta_{\text{alg}}) - p(\zeta^\star)\right\|_2 \le \frac{\delta}{4\,\|\gamma\|_2} + \beta\sqrt{d} \le \frac{\delta}{2\,\|\gamma\|_2}.$$

The complexity of the algorithm is bottlenecked by the cost of finding $\tilde{p}(\zeta_{\text{alg}})$. For each ζ on the grid the cost of evaluating $\tilde{p}(\zeta)$ induces a multiplicative $d\log(\frac{\|\gamma\|_2^2}{\delta})$ overhead. The cost of performing the binary search on the ζ grid is a multiplicative $\log(\frac{\|\gamma\|_2^2}{\mu\sqrt{\delta}})$ overhead; note that a binary search suffices because $\left\|\tilde{p}(\zeta_{\text{alg}})\right\|_2$ is monotonic by our consistent choice of rounding down, and hence $|\left\|\tilde{p}(\zeta_{\text{alg}})\right\|_2 - \theta|$ is unimodal. ■