An SDP Relaxation for the Sparse Integer Least Squares Problem

Alberto Del Pia * Dekun Zhou †

July 2, 2025

Abstract

In this paper, we study the *sparse integer least squares problem* (SILS), an NP-hard variant of least squares with sparse $\{0, \pm 1\}$-vectors. We propose an ℓ_1-based SDP relaxation, and a randomized algorithm for SILS, which computes feasible solutions with high probability with an asymptotic approximation ratio $1/T^2$ as long as the sparsity constant $\sigma \ll T$. Our algorithm handles large-scale problems, delivering high-quality approximate solutions for dimensions up to $d = 10,000$. The proposed randomized algorithm applies broadly to binary quadratic programs with a cardinality constraint, even for non-convex objectives. For fixed sparsity, we provide sufficient conditions for our SDP relaxation to solve SILS, meaning that any optimal solution to the SDP relaxation yields an optimal solution to SILS. The class of data input which guarantees that SDP solves SILS is broad enough to cover many cases in real-world applications, such as privacy preserving identification and multiuser detection. We validate these conditions in two application-specific cases: the *feature extraction problem*, where our relaxation solves the problem for sub-Gaussian data with weak covariance conditions, and the *integer sparse recovery problem*, where our relaxation solves the problem in both high and low coherence settings under certain conditions.

Key words: Semidefinite relaxation, Sparsity, Integer least square problem

1 Introduction

In numerous applications, one is interested in solving the *sparse integer least squares* (SILS) problem. SILS is a special class of linear regressions where the solution vectors are both sparse and consist of discrete values, typically in $\{0, \pm 1\}$. Applications can be found in multiuser detection, where only a subset of user terminals transmit binary symbols in a code-division multiple access (CDMA) system [59], in sensor networks, where sensors with low duty cycles are either silent (transmit 0) or active (transmit ± 1) [47], and in privacy preserving identification, where a sparse vector in $\{0, \pm 1\}$ is employed to approximate the "content" of feature data [42].

Formally, in SILS, an instance consists of an $n \times d$ matrix M, a vector $b \in \mathbb{R}^n$, and a positive integer $\sigma \leq d$. Our task is to find a vector x with (at most) σ non-zero entries which solves the optimization problem SILS or its variant SILS', defined as follows:

$$\begin{array}{lll} \min\limits_{x \in \{0, \pm 1\}^d} & \frac{1}{n} \|Mx - b\|_2^2 & \text{(SILS)} \\ \text{s.t.} & \|x\|_0 \leq \sigma, & \end{array} \qquad \begin{array}{lll} \min\limits_{x \in \{0, \pm 1\}^d} & \frac{1}{n} \|Mx - b\|_2^2 & \text{(SILS')} \\ \text{s.t.} & \|x\|_0 = \sigma. & \end{array}$$

Here, $\|x\|_0 := |\{i \in [d] : x_i \neq 0\}|$. One can interpret SILS' as SILS with extra information or belief on the optimal choice of sparsity of the optimal solution. These problems are closely

*Department of Industrial and Systems Engineering & Wisconsin Institute for Discovery, University of Wisconsin-Madison. E-mail: delpia@wisc.edu

†Department of Industrial and Systems Engineering & Wisconsin Institute for Discovery, University of Wisconsin-Madison. E-mail: dzhou44@wisc.edu

related to a class of sparse regression problems, where the goal is to find a sparse solution with continuous variables satisfying a box constraint [6]. In our case, the variables are restricted to discrete values in $\{0, \pm 1\}$, which introduces additional computational challenges.

Our approach. We propose our semidefinite programming (SDP) relaxations of the problems SILS and SILS'. SDP problems, under certain assumptions, can be solved in polynomial time up to an arbitrary accuracy, by means of the ellipsoid algorithm and the interior point methods [50, 32]. Specifically, if there exists a rational point X_0 and positive rational numbers r and R such that $X_0 + B(X_0, r) \subseteq \mathcal{F} \subseteq X_0 + B(X_0, R)$, where $\mathcal{F}$ denotes the feasible region of the SDP problem, then an ϵ-optimal solution to the SDP can be computed in time polynomial in $\log(R/r)$, $\log(1/\epsilon)$, and the encoding size of X_0 and the input data [14, 27]. Define the $n \times (1+d)$ matrix $A := \begin{pmatrix} -b & M \end{pmatrix}$, our relaxations are as follows:

$$\begin{aligned} \min_{W \succeq 0} \quad & \frac{1}{n} \operatorname{tr}(A^\top A W) \\ \text{s.t.} \quad & W_{11} = 1, \\ & \operatorname{tr}(W_x) \leq \sigma, \\ & 1_d^\top |W_x| \, 1_d \leq \sigma^2, \\ & \operatorname{diag}(W_x) \leq 1_d. \end{aligned} \qquad \text{(SILS-SDP)}$$

$$\begin{aligned} \min_{W \succeq 0} \quad & \frac{1}{n} \operatorname{tr}(A^\top A W) \\ \text{s.t.} \quad & W_{11} = 1, \\ & \operatorname{tr}(W_x) = \sigma, \\ & 1_d^\top |W_x| \, 1_d \leq \sigma^2, \\ & \operatorname{diag}(W_x) \leq 1_d. \end{aligned} \qquad \text{(SILS'-SDP)}$$

In these problems, the decision variables are both $(1+d) \times (1+d)$ matrices W. The matrix W_x is the sub-matrix of W obtained by dropping its first row and column. The constraints $1_d^\top |W_x| \, 1_d \leq \sigma^2$ and $\operatorname{tr}(W_x) \leq \sigma$ (or $\operatorname{tr}(W_x) = \sigma$) are relaxations of the original sparsity constraint. Using the almost identical analysis introduced in [13], one can show that SILS-SDP is indeed a relaxation of SILS.

Proposition 1. *Problem SILS-SDP is an SDP relaxation of problem SILS. Specifically,*

(i) Let x be a feasible solution to SILS, let w be obtained from x by adding a new first component equal to one, i.e., $w = \begin{pmatrix} 1 \\ x \end{pmatrix}$, and let $W := ww^\top$. Then, W is feasible to SILS-SDP and has the same cost as x.

(ii) Let W be a feasible solution to SILS-SDP, and let x be obtained from the first column of W by dropping the first entry. If $\operatorname{rank}(W) = 1$ and $x \in \{0, \pm 1\}^d$, then x is feasible to SILS and has the same cost as W.

Note that one can also show that SILS'-SDP is a relaxation of SILS' in a similar way.

Hardness and existing approaches. One can show that SILS and SILS' are NP-hard in their full generality, via a polynomial reduction from *Exact Cover by 3-sets (X3C)*. See [22] for details of X3C. To the best of our knowledge, existing algorithms for solving SILS or SILS' fall into the following categories:

(i) *Exact algorithms*: this category includes Sparse Sphere Decoding Algorithm [4], Sparsity-Exploiting Sphere Decoding-based MUD and Sparsity-Exploiting Decision-Directed MUD [59], and integer quadratic optimization algorithms (see, e.g., [6] and references therein). These algorithms generally require non-polynomial running time. Interestingly, it was shown in [4] that Sparse Sphere Decoding Algorithm has an expected running time polynomial in d in the case where M has i.i.d. standard Gaussian entries and there exists a sparse integer vector $z^* \in \{0, \pm 1\}^d$ such that the residual vector $b - Mz^*$ is comprised of i.i.d. Gaussian entries. However, this algorithm may result in an exponential running time for general input, such as a non-sparse z^*.

(ii) *Convex relaxation methods*: this category includes techniques such as Lasso [49, 59] and Basis Pursuit [11] relax the integer constraints by allowing x to take continuous values and promoting sparsity through ℓ_1-norm regularization. Although these methods are computationally efficient and have approximation guarantees under certain conditions (e.g. restricted isometry property (RIP) [9]), they yield solutions that are not integer-valued in general.

(iii) *Other practical algorithms*: this catrgory includes algorithms such as Adaptive Compressive Sampling Matching Pursuit (Adaptive CoSaMP) [47], Soft-Feedback Orthogonal Matching Pursuit (SF-OMP) [48], and discrete valued sparse ADMM algorithm [46]. These methods do not have approximation guarantees.

Connection to feature extraction and integer sparse recovery. Our work is particularly motivated by two important real-world applications, in which the underlyding data inputs both satisfy the following linear model assumption:

$$b = Mz^* + \epsilon, \tag{LM}$$

for some *ground truth* vector $z^* \in \mathbb{R}^d$ and for some small noise vector $\epsilon \in \mathbb{R}^n$. Note that in this setting, z^* and ϵ are unknown, that is, they are not part of the input of the problem. These two applications are:

(a) Feature extraction: In privacy-preserving data analysis and machine learning, one objective is to extract a subset of features that best represent the data [42, 54]. The integer constraint in SILS are essential for interpretability and compliance with privacy requirements. In this paper, we formally define the *feature extraction problem* as the problem SILS', where (LM) holds (for possibly a general vector z^*).

(b) Integer sparse recovery: In sensor network [47], digital fingerprints [34], array signal processing [55], compressed sensing [30], and multiuser detection [59, 45], recovering a sparse integer signal from noisy measurements is crucial. The challenge lies in accurately reconstructing the original integer signal z^* from observations b contaminated by noise. In this paper, we formally define the *integer sparse recovery problem*, the input satisfies (LM) for some $z^* \in \{0, \pm 1\}^d$ with known cardinality σ, and the goal is to recover z^* correctly.

We note that the integer sparse recovery problem is a special case of the broader *sparse recovery problem*, a fundamental topic across compressed sensing [9, 18], high-dimensional statistics [8, 53], and wavelet denoising [11]. In the sparse recovery problem, the input satisfies (LM) for some (possibly continuous) $z^* \in \mathbb{R}^d$ with support size σ, and our goal is to recover the signed support of z^*. For details on the sparse recovery problem, we refer interested readers to the excellent review by [12]. Observe that, under the assumptions of the integer sparse recovery problem, i.e., $z^* \in \{0, \pm 1\}^d$, determining the signed support of z^* is equivalent to determining z^* itself.

While existing methods provide valuable tools, they have limitations in handling the these two problems effectively. For the feature extraction problem, methods applicable to SILS' [4, 6, 47, 59] can also be applied to solve the feature extraction problem, as previously discussed. However, some of these methods do not have polynomial running time in general, while others offer no approximation guarantees of the solution.

For the integer sparse recovery problem, the approaches above can still be applied, but the same limitations persist. Another way to solve the integer sparse recovery problem is to solve the more general sparse recovery problem, where a large number of algorithms are developed [49, 8, 12, 20, 21]. [21] studies a problem similar to SILS' with $x \in \{0, 1\}^d$ and $z^* \in \{0, 1\}^d$ in (LM), where M and ϵ have i.i.d. Gaussian entries. They demonstrate an "all-or-nothing" phenomenon: if $n > n^*$ for some value n^*, the solution x^* closely approximates z^*; otherwise, it does not. Besides, Lasso [49] and Dantzig Selector [8] are among the most popular and the most useful approaches in solving sparse recovery problem. Theoretical guarantees

for these methods, including conditions such as mutual incoherence [53] and irrepresentable criteria [57], are well-studied. Define the *coherence* of a positive semidefinite matrix Ψ to be

$$\mu(\Psi) := \max_{i \neq j} \frac{|\Psi_{ij}|}{\sqrt{|\Psi_{ii}\Psi_{jj}|}}, \tag{1}$$

where we assume $0/0 = 0$ if necessary. In this paper, we say that an input model has a *high coherence* if we have $\mu(M^\top M) = \omega(1/\sigma)$, while it has a *low coherence* if we have $\mu(M^\top M) = \mathcal{O}(1/\sigma)$. It is shown that Lasso and Dantzig Selector converges to z^* when the coherence of $M^\top M$ is low [33, 35]. However, high coherence models often violate these assumptions, leading to sub-optimal performance of convex relaxation techniques [2, 44, 23]. Although other assumptions are studied, such as the *restricted isometry property (RIP)* and *null space property (NSP)*, they are oftentimes violated in many real-world applications [42]. For detailed discussions on various assumptions, we refer the interested readers to [58] and references therein.

Our contributions. In this paper, we further the understanding of the limits of computations for SILS and SILS', and we make the following key contributions:

1. *Randomized Algorithm for SILS with Approximation Guarantees.* We develop a randomized approximation algorithm for SILS. In fact, the algorithm not only works for SILS, but for any $\{0, \pm 1\}$ quadratic programs with a cardinality constraint, provided that the coefficient matrix of the quadratic function has non-negative diagonal entries. The input of the algorithm consists of an approximate optimal solution to SILS-SDP, and two threshold constants T and C; the output is a feasible solution to SILS with high probability. We show that on average, the expected objective value of such solution is a $1/T^2$ multiple of the optimal value to SILS, after subtracting an additional term that depends on T, C and the input data (M, b, σ). It can be shown that when $\sigma \ll T$, the additional term will diminish as $(\sigma, T) \to \infty$, and hence Algorithm 1 is an asymptotic $1/T^2$-approximation algorithm. To the best of our knowledge, Algorithm 1 is the first known randomized algorithm for SILS that has an approximation guarantee. We also conduct extensive numerical tests, showing that our algorithm is highly practical, as Algorithm 1 requires only an approximate solution to SILS-SDP. It can deliver high-quality solutions to SILS for $d = 2000$ in less than a minute, and for $d = 10000$ in approximately ten minutes.

2. *Sufficient Conditions for Solving SILS'.* We also provide sufficient conditions under which any optimal solution to SILS'-SDP is of rank one and in $\{0, \pm 1\}$, and thus yields an optimal solution to SILS'. To the best of our knowledge, our results are the first ones that study the polynomial solvability of SILS' in its full generality. We then give both theoretical and computational evidence, aiming to explain the flexibility of SILS'-SDP. To be more specific, we tailor our sufficient conditions to the special cases where LM holds, and show that (i) SILS'-SDP can solve the feature extraction problem with high probability in the case where rows of M are i.i.d. standard Gaussian vectors, and where z^* satisfies some mild assumptions; (ii) We show that SILS'-SDP can accurately recover the sparse integer vector z^* in the integer sparse recovery problem under certain assumptions. Notably, the assumptions in (ii) do not depend on the coherence of $M^\top M$, indicating that our method is robust even when the data matrix M exhibits high coherence. We demonstrate this both theoretically and computationally by analyzing a high-coherence data model where traditional ℓ_1-based methods like Lasso and Dantzig Selector often fail, yet our SDP relaxation successfully recovers z^* with high probability. For low coherence scenarios, we specialize our general results to provide conditions under which SILS'-SDP also guarantees exact recovery of z^*. We validate these conditions in a well-studied low-coherence data model with i.i.d. standard Gaussian entries, showing that our method consistently recovers z^* with high probability in this setting as well. This highlights the effectiveness and broad applicability of our approach across different coherence regimes compared to existing sparse recovery techniques.

Related work. A substantial body of literature addresses quadratic programs with sparsity and/or integer constraints, but these problems either differ fundamentally from our focus, both in their problems of interest and in their methodologies, or they do not provide approximation guarantees. For instance, [40] considered quadratic programs with cardinality constraints on continuous variables, instead of discrete variables. Their approach relies on an SDP relaxation derived from the conjugate dual and incorporates a penalty term involving the ℓ_2-norm of the solution, which is absent in our problem. This method is extended to a more general class of penalty functions by [17]. An equivalent SDP relaxation is also proposed by [29] in the setting where the ℓ_2-norm penalty is absent. [38] introduced a "Suggest-and-Improve" framework for solving non-convex quadratically constrained quadratic programming problems, later applied to integer least squares problems using an SDP relaxation in [39]. Their framework addresses integer constraints by random sampling from the SDP solution to find a vector x satisfying $x_i(x_i - 1) \geq 0$ in expectation, followed by solution rounding, and heuristics to improve quality, but does not provide a known approximation guarantee. In contrast, our approach leverages the structure of the lifted solution to construct a high-probability feasible solution in the original discrete space other than simple rounding, and thus provides a known approximation guarantee.

Organization of this paper. In Section 2, we introduce our randomized algorithm for SILS, and develop an approximation gap of this algorithm. In Section 3, we provide our general sufficient conditions for SILS'-SDP to solve SILS'. In Section 4, we apply these sufficient conditions to the scenarios where (LM) holds, and discuss the implications for the feature extraction problem and the integer sparse recovery problem. In Section 5, we present the numerical results. To streamline the presentation, we defer some proofs to Appendices A to F, and we leave detailed and additional empirical results in Appendix G.

Notation. Sets, vectors, and matrices. For any positive integer d, we define $[d] := \{1, 2, \ldots, d\}$. 0_d denotes the d-vector of zeros, and 1_d denotes the d-vector of ones. Let x be a d-vector. The *support* of x is the set $\mathrm{Supp}(x) := \{i \in [d] : x_i \neq 0\}$. For an index set $\mathcal{I} \subseteq [d]$, we denote by $x_{\mathcal{I}}$ the subvector of x whose entries are indexed by $\mathcal{I}$. For $1 \leq p \leq \infty$, we denote the *p-norm* of x by $\|x\|_p$. We say that x is a *unit vector* if $\|x\|_2 = 1$. Given two index sets $\mathcal{I} \subseteq [m]$, $\mathcal{J} \subseteq [n]$, we denote by $M_{\mathcal{I},\mathcal{J}}$ the sub-matrix of M consisting of the entries in rows $\mathcal{I}$ and columns $\mathcal{J}$. We denote by $|M|$ the matrix obtained from M by taking the absolute values of the entries. We denote by $\mathcal{S}^n$ the set of all $n \times n$ symmetric matrices. If $M, N \in \mathcal{S}^n$, we use $M \succeq N$ to denote that $M - N$ is a positive semidefinite matrix. We denote by $M^\dagger$ the Moore-Penrose generalized inverse of M. The *p-to-q norm* of a matrix P, where $1 \leq p, q \leq \infty$, is defined as $\|P\|_{p\to q} := \min_{\|x\|_p=1} \|Px\|_q$. The *2-norm* of a matrix P is defined by $\|P\|_2 = \|P\|_{2\to 2}$. The *infinity norm*, also known as *Chebyshev norm*, of P is defined by $\|P\|_\infty := \max_{i,j} |P_{ij}|$.
Optimality gap. Denote w^* to be the optimal solution to a optimization problem $\mathcal{P}$ with objective function f and input D. We say a randomized algorithm $\mathcal{A}$ is an r-approximation algorithm (or with an approximation ratio r) to the optimization problem, if $\mathcal{A}$ can output a random vector $\bar{w}$ with input D such that $\mathbb{E}f(\bar{w}) \geq 1/r \cdot f(w^*)$ if $\mathcal{P}$ is a maximization problem, and $\mathbb{E}f(\bar{w}) \leq r \cdot f(w^*)$ if $\mathcal{P}$ is a minimization problem.

2 A randomized algorithm for SILS

In this section, we present a novel randomized algorithm for the following binary quadratic optimization problem with sparsity constraint:

$$\begin{array}{ll} \min\limits_{x \in \{0,\pm 1\}^d} & x^\top P x - 2c^\top x \\ \text{s.t.} & \|x\|_0 \leq \sigma, \end{array} \tag{SBQP}$$

where we assume that the input matrix $P \in \mathbb{R}^{d\times d}$ satisfies $P_{ii} \geq 0,\ \forall i \in [d]$, i.e., all its diagonal entries are non-negative, thus the objective function is not necessarily convex. Note that the optimal value of SBQP is non-positive, due to the feasibility of 0_d. Moreover, if one takes $P = M^\top M$ and $c = M^\top b$, then SILS is equivalent to SBQP by ignoring a constant $b^\top b$. To the best of our knowledge, this is the first randomized algorithm for solving a binary quadratic optimization problem with cardinality constraint. Our proposed randomized algorithm is inspired by [10], where the authors presented a $\mathcal{O}(\log d)$-approximation algorithm for maximizing a quadratic function $x^\top Px$ over $\{\pm 1\}^d$. In their setting, the authors assume that $P_{ii} = 0$, as x_i^2 must be one. However, in SBQP, such assumption is not reasonable due to the cardinality constraint. This issue also prevents one from applying their algorithm directly, as one cannot obtain a sparse vector. In fact, [10] introduced a specific random variable that decides whether a chosen entry is ± 1, similar to the ideas presented in [24]. The idea depends on the fact that the u_i's, column vectors in the square root of the (approximated) optimal solution, are unit vectors, which is not true in Algorithm 1. Moreover, we have an additional linear term $-2c^\top x$. In this section, we show that, all these problems can all be solved by choosing a distribution that carefully handles sparsity, at a cost of an additional additive term in the approximation gap.

Let the matrix $Q(c,P) := \begin{pmatrix} 0 & -c^\top \\ -c & P \end{pmatrix}$. Denote $\mathrm{SDP}(c,P)$ to be the optimization problem by replacing the objective function $1/n \cdot \mathrm{tr}(A^\top AW)$ by $\mathrm{tr}(Q(c,P)W)$ in SILS-SDP. Following the proof idea of Proposition 1, it is clear that $\mathrm{SDP}(c,P)$ is indeed a relaxation of SBQP. We define a threshold function $h(x)$ which takes value 1 if $x > 1$, x if $-1 < x < 1$, and -1 if $x < -1$. Now, we present the detailed randomized algorithm in Algorithm 1.

Algorithm 1 Randomized Algorithm for SBQP

Input: An ϵ-approximated optimal solution $W^* \in \mathbb{R}^{(d+1)\times(d+1)}$ to $\mathrm{SDP}(c,P)$, threshold constants $0 < C \leq 1$ and $T > 0$.

Output: A vector $\bar{x}$ in $\{0, \pm 1\}^d$

$U := (u_0, u_1, \ldots, u_d) \in \mathbb{R}^{(d+1)\times(d+1)} \leftarrow \sqrt{W^*}$
Generate a random vector $g \sim \mathcal{N}(0_{d+1}, I_{d+1})$
$z_0 \leftarrow u_0^\top g$, $y_0 \leftarrow h(z_0/T)$
Sample $x_0 = 1$ with probability $(1+y_0)/2$, $x_0 = -1$ with probability $(1-y_0)/2$
for $k = 1, 2, \ldots, d$ **do**
  $p_k \leftarrow 2/3 \cdot \|u_i\|_2^2$ if $\|u_i\|_2 \geq C$, and $p_k \leftarrow 0$ if otherwise
  Sample $\epsilon_k = 1$ with probability p_k and $\epsilon_k = 0$ with probability $1 - p_k$, independent of k and g
  $\tilde{u}_k \leftarrow \epsilon_k \cdot u_k / p_k$ (where we assume $0/0 = 0$), $z_k \leftarrow \tilde{u}_k^\top g$, $y_k \leftarrow h(z_k/T)$
  Sample $x_k = \mathrm{sign}(y_k)$ with probability $|y_k|$, and $x_k = 0$ with probability $1 - |y_k|$
end for
return $\bar{x} := \mathrm{sign}(x_0) \cdot (x_1, \ldots, x_d)^\top$

In this paper, we abbreviate 'with high probability' with 'w.h.p.', meaning with probability at least $1 - \mathcal{O}(1/d) - \mathcal{O}(\exp(-c\sigma))$ for some absolute constant $c > 0$. An approximation gap of Algorithm 1 is stated as follows, and the proof is left in Appendix A.

Theorem 1. *Assume P is a $d \times d$ symmetric matrix with non-negative diagonal entries, and c is a d-vector. Denote W^* to be an ϵ-optimal solution to $\mathrm{SDP}(c,P)$, x^* to be the optimal solution to SBQP. Let $\bar{x}$ be the output of Algorithm 1, with input W^* and threshold constants $0 < C \leq 1$ and $T > 0$. Define $B := \|Q(c,P)\|_\infty$. Then, we have*

$$\mathbb{E}(\bar{x}^\top P\bar{x} - 2c^\top \bar{x}) - B \cdot \left[f(T,C,\sigma,d) + \frac{1}{T^2}(3\sigma + \sigma^2) + \frac{\sqrt{3}}{\sqrt{2}T} \min\left\{ d, \frac{\sigma}{C^2} \right\} \right]$$

$$\leq \frac{1}{T^2} \cdot \operatorname{tr}(Q(c,P)W^*) \leq \frac{1}{T^2} \cdot \left[(x^*)^\top P x^* - 2c^\top x^* + \epsilon\right]$$

where $f(T,C,\sigma,d) := \mathcal{O}\Big(\sigma e^{-C^2T^2}[\min\{d,\sigma/C^2\}/(CT)+T/C]\Big)$*, and we omit possibly a constant scaling of* T *in the Big-O notation. Furthermore, w.h.p.,* $\bar{x}$ *is feasible to SBQP.*

Note that, in the case where $\sigma \ll T$ and $B, C > 0$ are fixed, the term $g(B,T,C,\sigma,d) := B \cdot \left[f(T,C,\sigma,d) + \frac{1}{T^2}(3\sigma+\sigma^2) + \frac{\sqrt{3}}{\sqrt{2}T}\min\left\{d, \frac{\sigma}{C^2}\right\}\right]$ in Theorem 1 is diminishing as $(\sigma,T) \to \infty$, and thus we can obtain a solution $\bar{x}$ with an expected objective value that is an asymptotically $1/T^2$ multiple of $(x^*)^\top P x^* - 2c^\top x^* + \epsilon$. Formally, we obtain the following corollary:

Corollary 2. *Assume* P *is a* $d \times d$ *symmetric matrix with non-negative diagonal entries,* c *is a* d*-vector, and assume that* $\|Q(c,P)\|_\infty \leq 1$*. Denote* W^* *to be an* ϵ*-optimal solution to* $\mathrm{SDP}(c,P)$*, and* $\bar{x}$ *to be the output of Algorithm 1. Suppose that there exists a threshold value* T*, where* T *is a function value of the input* P *and* c*, such that* $\sigma/T \to 0$ *as* $(\sigma,T) \to \infty$*, then Algorithm 1 with input* W^**, a fixed constant* $0 < C \leq 1$*, and* T*, is an asymptotic* $(1/T^2)$*-approximation algorithm for SBQP, in expectation. Furthermore, with high probability,* $\bar{x}$ *is feasible to SBQP.*

In particular, suppose $\sigma/\sqrt{\log d} \to 0$*, then Algorithm 1 with input tuple* (W^*, C, T) *is an asymptotic* $(1/\log d)$*-approximation algorithm for SBQP in expectation.*

Remark. In [10], the authors take $T = 4\sqrt{\log(d)}$ and obtain a $\mathcal{O}(\log(d))$-approximation algorithm for maximization binary quadratic problems. In Theorem 1, we show that we can obtain a similar result by taking the same value for such T, and if we further fix $0 < C \leq 1$, at the cost of an additional term $g(B,T,C,\sigma,d)$. If we further assume that $\sigma \ll \sqrt{\log(d)}$ and B is fixed, then we obtain an asymptotic $\mathcal{O}(1/\log(d))$-approximation algorithm by Theorem 2. Finally, as suggested by Theorem 1, for different input $Q(c,P)$ and σ, one can accordingly choose different values for T and C to obtain a acceptable trade-off between the term $g(B,T,C,\sigma,d)$ and the multiplicative factor $1/T^2$.

In Section 5.1, we will demonstrate some numerical results of Algorithm 1. We note that, although SDPs can be solved up to an arbitrary accuracy in polynomial time, applying existing SDP solvers to solve SILS-SDP becomes more and more challenging as the dimension of the input increases. However, it is possible to solve SILS-SDP approximately via an approximation algorithm that leverages the unique structures inherent in SILS-SDP, thereby enhancing computational efficiency. Further details regarding the implementation and effectiveness of these approximation methods are discussed in Section 5.1 and Appendix G.1.

3 Sufficient conditions for recovery

In this section, we study SILS'. Note that one can interpret solving SILS' as solving SILS given an optimal choice of σ. For the ease of illustration, starting from this section, we say that *SILS'-SDP recovers* x^*, if $x^* \in \{0, \pm 1\}^d$, and SILS'-SDP admits a unique rank-one optimal solution $W^* := \begin{pmatrix} 1 \\ x^* \end{pmatrix}\begin{pmatrix} 1 \\ x^* \end{pmatrix}^\top$. Due to Proposition 1, the vector x^* is then optimal to SILS', and hence we also say that *SILS'-SDP solves SILS'* if there exists a vector $x^* \in \{0, \pm 1\}^d$ such that SILS'-SDP recovers x^*. We remark that, if SILS'-SDP solves SILS', then SILS' can be indeed solved in polynomial time by solving SILS'-SDP, because we can obtain x^* by checking the first column of W^*.

We present Theorems 3 and 4, which are two of the main results of this section. In both theorems, we provide sufficient conditions for SILS'-SDP to solve SILS', which are primarily focused on the input $A = (M, -b)$ and σ. The statements require the existence of two parameters μ_2^* and δ, and in Theorem 3 we additionally require the existence of a decomposition of a specific

matrix Θ. Therefore, both theorems below can help us identify specific classes of problem SILS' that can be solved by SILS'-SDP. As a corollary to Theorem 4, we then obtain Theorem 5, where we show that in a low coherence model, SILS' can be solved by SILS'-SDP under certain conditions.

It is worth to note that, although the linear model assumption (LM) is often present in the literature in integer least square problems (see, e.g., [4]), in this section we consider the general setting where we do not make this assumption. To help readers understand better the complicated geometry, we will split the section into two parts. In the first part, we discuss KKT conditions, and state Lemma 1 based on KKT conditions, along with a stronger assumption that two specific parameters μ_2^* and δ exist. In the second part, we leave the statements of the two theorems, and discuss the conditions semantically. The proofs can be found in Appendix B.

3.1 KKT conditions

In this section, we study the Karush–Kuhn–Tucker (KKT) conditions [31]. We start by studying the dual of SILS'-SDP, and provide KKT conditions when SILS'-SDP admits an optimal solution W^*. Based on KKT conditions, we then provide a cleaner sufficient conditions for recovering a sparse vector $x^* \in \{0, \pm 1\}^d$ in Lemma 1.

The dual problem of SILS'-SDP is

$$\begin{array}{ll} \max\limits_{Y \succeq 0,\ \mu_1 \in \mathbb{R},\ \mu_2 \geq 0,\ \mu_3 \geq 0} & -\mu_1 - \sigma\mu_2 - \sigma^2\mu_3 - p^\top 1_d \\ \text{s.t.} & \left\| \dfrac{A^\top A}{n} + R(\mu_1, \mu_2, p) - Y \right\|_\infty \leq \mu_3, \end{array} \tag{SILS'-SDP-dual}$$

where $R(\mu_1, \mu_2, p) := \begin{pmatrix} \mu_1 & \\ & \mu_2 I_d + p \end{pmatrix}$. Denote a convex function $f : \mathbb{R}^{(1+d)\times(1+d)} \to \mathbb{R}$ by $f(Z) := (0, 1_d^\top)|Z| \begin{pmatrix} 0 \\ 1_d \end{pmatrix}$. For $Z \in \mathbb{R}^{(1+d)\times(1+d)}$, denote by $\partial f(Z)$ the *sub-differential* of f at Z, i.e., $\partial f(Z) := \{G \in \mathbb{R}^{(1+d)\times(1+d)} : f(Y) \geq f(Z) + \operatorname{tr}(G(Y - Z)),\ \forall Y \in \mathbb{R}^{(1+d)\times(1+d)}\}$. Note that

$$\partial f(Z) = \left\{ U \in \mathbb{R}^{(1+d)\times(1+d)} : U_{ij} = \begin{cases} 0, & \text{if at least one of } i, j \leq 1, \\ \operatorname{sign}(Z_{ij}), & \text{if both of } i, j \geq 2 \text{ and } Z_{ij} \neq 0, \\ \in [-1, 1], & \text{otherwise.} \end{cases} \right\}. \tag{2}$$

Then, KKT conditions state that $W^* = \begin{pmatrix} 1 \\ x^* \end{pmatrix} \begin{pmatrix} 1 \\ x^* \end{pmatrix}^\top$ is optimal to SILS'-SDP if and only if there exist dual variables $Y^* = \begin{pmatrix} Y_{11}^* & (y^*)^\top \\ y^* & Y_x^* \end{pmatrix}$, μ_1^*, p^*, μ_2^*, and μ_3^* feasible to SILS'-SDP-dual such that:

$$O_{d+1} \in \left\{ \frac{1}{n} A^\top A - Y^* + \begin{pmatrix} \mu_1^* & \\ & \operatorname{diag}(p^*) + \mu_2^* I_d \end{pmatrix} \right\} + \mu_3^* \partial f(W^*), \tag{KKT-1}$$

$$Y^* W^* = O_{1+d} \Longleftrightarrow Y^* \begin{pmatrix} 1 \\ x^* \end{pmatrix} = 0_{1+d}, \tag{KKT-2}$$

$$(p^*)^\top (\operatorname{diag}(W_x^*) - 1_d) = 0, \tag{KKT-3}$$

where we apply Minkowski sum in (KKT-1). If we focus our attention on the block matrix that contains $1/n \cdot (M^\top M)_{S,S}$ in (KKT-1), we obtain that

$$-\mu_3^* x_S^* (x_S^*)^\top = \left[\frac{1}{n} M^\top M - Y_x^* + \operatorname{diag}(p^* + \mu_2^* 1_d) \right]_{S,S}. \tag{3}$$

Moreover, insert $(Y_x^*)_{S,S}$ in (3) into (KKT-2), we have that

$$\mathrm{diag}(p_S^*)x_S^* = -\frac{1}{n}(M^\top M)_{S,S}x_S^* - \sigma\mu_3^* x_S^* - y_S^* - \mu_2^* x_S^*. \tag{4}$$

Note that (4) uniquely determines the vector p_S^* if other dual variables are determined. The constraint $p_S^* \geq 0_\sigma$ is then implied by the following two stronger conditions:

$$\mu_2^* \leq -\lambda_{\min}\Big(\frac{1}{n}(M^\top M)_{S,S}\Big) + \delta, \tag{5}$$

$$\mu_3^* := \frac{1}{\sigma}\Big\{\lambda_{\min}\Big(\frac{1}{n}(M^\top M)_{S,S}\Big) - \delta + \min_{i\in S}\Big[-y^* - \frac{1}{n}(M^\top M)_{S,S}x_S^*\Big]_i / x_i^*\Big\}. \tag{6}$$

Here, the minimum eigenvalue of the matrix $1/n \cdot (M^\top M)_{S,S}$ introduced in (5) and (6) helps guarantee that, a block matrix $H_{S,S}$ defined in the statement of Lemma 1, is positive semidefinite, which is a necessary condition for $Y^* \succeq 0$. The details will be made clear in the proof of Lemma 1, in Appendix B.

Together with all these intuitions, we are ready to state Lemma 1, about block structures of dual variables that guarantee recovery of x^*:

Lemma 1. *Let $x^* \in \{0, \pm 1\}^d$, define $S := \mathrm{Supp}(x^*)$, and assume $|S| = \sigma$. Define $y^* := -M^\top b/n$, $Y_{11}^* := -(y_S^*)^\top x_S^*$, and assume $Y_{11}^* > 0$. Let $\delta > 0$, μ_2^* satisfy (5), μ_3^* be defined by (6), $p^* \in \mathbb{R}^d$ be a vector with $p_{S^c}^* := 0_{d-\sigma}$ and p_S^* satisfying (4). Let $Y_x^* \in \mathbb{R}^{d\times d}$ be a matrix that satisfies (3), and let $H := Y_x^* - \frac{1}{Y_{11}^*}y^*(y^*)^\top$. Then we have $p^* \geq 0_d$, $\lambda_2(H_{S,S}) \geq \delta$, and $H_{S,S} \succeq 0$.*

Assume, in addition, that the following conditions are satisfied:

1A. $H_{S^c,S^c} \succeq H_{S^c,S}H_{S,S}^\dagger H_{S^c,S}^\top$;

1B. $H_{S^c,S}x_S^* = 0_{d-\sigma}$;

1C. $\big\|(\frac{1}{n}M^\top M - Y_x^*)_{S^c,S}\big\|_\infty \leq \mu_3^*$;

1D. $\big\|(\frac{1}{n}M^\top M - Y_x^*)_{S^c,S^c} + \mu_2^* I_{d-\sigma}\big\|_\infty \leq \mu_3^*$.

Then $W^* = w^*(w^*)^\top$, where $w^* = \begin{pmatrix}1\\ x^*\end{pmatrix}$, is an optimal solution to SILS'-SDP. Furthermore, if we also assume that $\lambda_2(H) > 0$, then W^* is the unique optimal solution to SILS'-SDP.

Remark. In this remark, we draw attention to the fact that the assumption $Y_{11}^* > 0$ in Lemma 1 is actually natural, given that $\sigma \geq 1$ is the optimal support size of SILS. Indeed, for any optimal solution x^* to SILS', one must have $\|Mx^* - b\|_2^2 = (x^*)^\top M^\top Mx^* - 2b^\top Mx^* + \|b\|_2^2 < \|b\|_2^2$, since otherwise we choose $x^* = 0_d$. This implies $0 \leq \|Mx^*\|_2^2 < 2b^\top Mx^* = n \cdot Y_{11}^*$. Finally, we point out that the optimality of σ in SILS is not necessarily required in Lemma 1 - all that is required are the assumptions made there.

3.2 Main theorems for recovery

In this section, we state the main theorems for recovery. In a nutshell, we take different candidates for $(Y_x^*)_{S^c,S}$ in Lemma 1, and present the corresponding sufficient conditions for recovery. Note that in Lemma 1, our choice of $(Y_x^*)_{S,S}$ is fixed (which is implied by (3)). Thus, it would be well-motivated if we further fixed $(Y_x^*)_{S^c,S}$ to be a specific determined matrix, and then construct $(Y_x^*)_{S^c,S^c}$ accordingly. Particularly, in Theorem 3, we assign $(Y_x^*)_{S^c,S}$ to be the optimal solution to the optimization problem

$$\min \left\|\frac{1}{n}M^\top M - (Y_x)_{S^c,S}\right\|_F \qquad \text{s.t.} \qquad (Y_x)_{S^c,S}x_S^* = -y_S^*, \tag{7}$$

where we relax the max norm of the matrix in **1C** by its Frobenius norm, and enforce **1B** in the constraint set. In fact, 7 admits a closed-form optimal solution. In Theorem 4, we we assign $(Y_x^*)_{S^c,S}$ to be a even simpler matrix - a rank-one matrix $-y^*_{S^c}(x^*_S)^\top/\sigma$.

We note here, although these candidates for $(Y_x^*)_{S^c,S}$ might not make perfect sense for general data inputs (M, b, σ), we found that they fit well in (sub-)Gaussian data matrix M and the linear model assumption (LM). We leave these theorems here as they might still be of interest for some other specific data inputs. Further discussion on (sub-)Gaussianity, (LM), and interpretation of the sufficient conditions tailored in (LM) are presented in Section 4.

We state the first theorem in this section:

Theorem 3. *Let $x^* \in \{0, \pm 1\}^d$, define $S := \mathrm{Supp}(x^*)$, and assume $|S| = \sigma$. Define $y^* := -M^\top b/n$, $Y_{11}^* := -(y_S^*)^\top x_S^*$, and assume $Y_{11}^* > 0$. Then, SILS'-SDP recovers x^*, if there exists a constant $\delta > 0$ such that the following conditions are satisfied:*

A1. *$\left\|\frac{1}{n\sigma}(M^\top M)_{S^c,S}x_S^* + \frac{1}{\sigma}y^*_{S^c}\right\|_\infty \le \mu_3^*$, where μ_3^* is defined by (6)*

A2. *There exists μ_2^* satisfying (5) such that the matrix $\Theta := \frac{1}{n}(M^\top M)_{S^c,S^c} + \mu_2^* I_{d-\sigma} - \frac{1}{Y_{11}^*}y^*_{S^c}(y^*_{S^c})^\top - R\frac{1}{\delta}\big(I_\sigma - \frac{1}{\sigma}x_S^*(x_S^*)^\top\big)R^\top$ can be written as the sum of two matrices $\Theta_1 + \Theta_2$, with $\Theta_1 \succ 0$, $\|\Theta_2\|_\infty \le \mu_3^*$ or $\Theta_1 \succeq 0$, $\|\Theta_2\|_\infty < \mu_3^*$, where $R := \frac{1}{n}(M^\top M)_{S^c,S} - \frac{1}{Y_{11}^*}y^*_{S^c}(y_S^*)^\top$.*

Remark. We first remark that condition **A1** would not be a very restricted assumption, as we are optimizing the relaxed problem (7), and one can choose $\delta > 0$ in (6) wisely according to the optimal value of (7). Plus, condition **A2** in Theorem 3 is not as strong as it might seem. This condition asks for a decomposition of Θ into the sum of a positive definite Θ_1 and another matrix Θ_2 with infinity norm upper bounded by μ_3^*. To construct Θ_1, the following informal idea may be helpful. By Lemma 4 (which can be found in Appendix B), $M^\top M \succeq 0$ implies

$$(M^\top M)_{S^c,S^c} \succeq (M^\top M)_{S^c,S}M^\dagger_{S,S}(M^\top M)^\top_{S^c,S}.$$

Therefore, if $(M^\top M)_{S^c,S^c}$ is large enough and δ is chosen wisely, the matrix

$$\frac{1}{n}(M^\top M)_{S^c,S^c} - \frac{1}{n}(M^\top M)_{S^c,S}\frac{1}{\delta}\big(I_\sigma - \frac{1}{\sigma}x_S^*(x_S^*)^\top\big)\frac{1}{n}(M^\top M)_{S,S^c}$$

is positive semidefinite and can be used to construct the positive semidefinite matrix Θ_1.

Numerically, we found that such decomposition $\Theta = \Theta_1 + \Theta_2$ often exists for several different instances; however, it can be challenging to write it down explicitly. A specific instance is given in the proof of Theorem 9 in Appendix E. In particular, it is an interesting open problem to obtain a simple sufficient condition which guarantees the existence of such decomposition.

In the next theorem, the sufficient conditions are easier to check than those in Theorem 3. This is because the main idea of Theorem 4 depends on a simpler structure of $(Y_x^*)_{S^c,S}$, and hence the theorem statement only requires the existence of two parameters μ_2^* and δ.

Theorem 4. *Let $x^* \in \{0, \pm 1\}^d$, define $S := \mathrm{Supp}(x^*)$, and assume $|S| = \sigma$. Define $y^* := -M^\top b/n$, $Y_{11}^* := -(y_S^*)^\top x_S^*$, and assume $Y_{11}^* > 0$. Denote $\theta := \arccos\left(\frac{(y_S^*)^\top x_S^*}{\sqrt{\sigma}\|y_S^*\|_2}\right)$. Then, SILS'-SDP recovers x^*, if there exists a constant $\delta > 0$ such that the following conditions are satisfied:*

B1. *$\left\|\frac{1}{n}(M^\top M)_{S,S^c} + \frac{1}{\sigma}y^*_{S^c}(x_S^*)^\top\right\|_\infty \le \mu_3^*$, where μ_3^* is defined by (6);*

B2. *There exists μ_2^* satisfying (5) such that $\left\|\frac{1}{n}(M^\top M)_{S^c,S^c} + \mu_2^* I_{d-\sigma}\right\|_\infty + \left\|\frac{1}{Y_{11}^*}y^*_{S^c}(y^*_{S^c})^\top\right\|_\infty + \frac{1-\cos^2(\theta)}{\sigma\delta\cos^2(\theta)}\|y^*_{S^c}\|_\infty^2 < \mu_3^*$.*

Next, we give a corollary to Theorem 4, which shows that the assumptions of Theorem 4 can be fulfilled in models with a low coherence.

Corollary 5. *Let* $x^* \in \{0, \pm 1\}^d$*, define* $S := \mathrm{Supp}(x^*)$*, and assume* $|S| = \sigma$*. Define* $y^* := -M^\top b/n$*,* $Y_{11}^* := -(y_S^*)^\top x_S^*$*, and assume* $Y_{11}^* > 0$*. Denote* $\theta := \arccos\left(\frac{(y_S^*)^\top x_S^*}{\sqrt{\sigma}\|y_S^*\|_2}\right)$*. Let* $\Delta_1 := \min_{i\in S}(-y_i^*/x_i^*) - \|y_{S^c}^*\|_\infty$*,* $\Delta_2 := \min_{i\in S}(-y_i^*/x_i^*) - \sigma\|y_{S^c}^*\|_\infty^2/Y_{11}^* + \frac{1-\cos^2(\theta)}{\delta\cos^2(\theta)}\|y_{S^c}^*\|_\infty^2$*, and assume that the columns of* M *are normalized such that* $\max_{i\in[d]}\|M_i\|_2 \le 1$*. Then, SILS'-SDP recovers* x^**, if there exists a constant* $\delta > 0$ *such that the following conditions are satisfied:*

- ***C1.*** $\lambda_{\min}\big(\frac{1}{n}(M^\top M)_{S,S}\big) - \delta - \big\|\frac{1}{n}(M^\top M)_{S,S}x_S^*\big\|_\infty + \min_{j=1,2}\Delta_j \ge \Delta > 0$ *for some constant* Δ*;*
- ***C2.*** *There exists* μ_2^* *satisfying* (5) *such that* $\big\|\mathrm{diag}\left(M^\top M/n + \mu_2^* I_d\right)_{S^c}\big\|_\infty < \Delta/\sigma$*;*
- ***C3.*** $\mu(M^\top M) < \Delta/\sigma$*, where* $\mu(\cdot)$ *is defined in* (1)*.*

Proof. We define μ_3^* as in (6). From **C3**, we obtain that $\max_{i\neq j}|(M^\top M/n)_{ij}| \le \mu(M^\top M/n) = \mu(M^\top M) \le \frac{\Delta}{\sigma}$. Then, we observe that $\mu_3^* \ge \frac{1}{\sigma}\Big\{\lambda_{\min}\big(\frac{1}{n}(M^\top M)_{S,S}\big) - \delta + \min_{i\in S}(-y_i^*/x_i) - \big\|\frac{1}{n}(M^\top M)_{S,S}x_S^*\big\|_\infty\Big\}$ and $\big\|\frac{1}{n}(M^\top M)_{S,S^c} + \frac{1}{\sigma}y_{S^c}^*(x_S^*)^\top\big\|_\infty \le \big\|\frac{1}{n}(M^\top M)_{S,S^c}\big\|_\infty + \frac{1}{\sigma}\|y_{S^c}^*\|_\infty$. Combining these facts with **C1**, we see that **B1** holds. If, in addition, **C2** holds, we obtain **B2**. □

Remark. Theorem 5 shows that, if the data matrix $M^\top M$ has a low coherence, SILS'-SDP can solve SILS' well under conditions **C1** and **C2**. In this remark, we informally illustrate how these two conditions can be easily fulfilled in certain scenarios. Observe that **C1** and **C2** hold if $\min_{j=1,2}\Delta_j$ is sufficiently large, and it is indeed possible to obtain a large $\min_{j=1,2}\Delta_j$. Intuitively, a large Δ_1 can be obtained if, for example, there is a set S with cardinality σ such that $\min_{i\in S}|y_i^*| - \|y_{S^c}^*\|_\infty$ is large, and $x_S^* = \mathrm{sign}(y_S^*)$. In addition, the requirement that Δ_2 is large is not as restrictive as it might seem. In particular, if $\cos(\theta)$ is close to one, we easily obtain a large Δ_2 if we secure a large Δ_1. Indeed, since $\sigma\|y_{S^c}^*\|_\infty^2/Y_{11}^* = \sigma\|y_{S^c}^*\|_\infty^2/(-\sum_{i\in S}y_i^*x_i^*)$, each term in the summation on the denominator is always greater than $\|y_{S^c}^*\|_\infty$ if Δ_1 is large. Thus, this term is in fact upper bounded by $\|y_{S^c}^*\|_\infty$. As another term $[1-\cos^2(\theta)]/\cos^2(\theta)\cdot\|y_{S^c}^*\|_\infty^2$ vanishes given that $\cos(\theta)$ is close to one, we thus obtain that $\Delta_2 \approx \Delta_1$, and so Δ_2 is also large.

While the above ideas on how **C1** and **C2** can be satisfied are not very precise, they can be further formalized and used in proofs for some concrete data models, including those given in the next section.

4 Consequences for linear data models

In this section, we showcase the power of Theorems 3 and 4, by presenting some of their implications for the feature extraction problem and the integer sparse recovery problem, as defined in Section 1. First, note that we can directly employ these two theorems and Theorem 5 in the specific settings of the two problems, in order to obtain corresponding sufficient conditions for SILS'-SDP to solve these problems. To avoid repetition, we do not present these specialized sufficient conditions, and we leave their derivation to the interested reader. Instead, we focus on the consequences of Theorems 3 and 4 for these two problems, that we believe are the most significant. In Section 4.1, we consider the feature extraction problem, where M and ϵ have sub-Gaussian entries. We specialize Theorem 4 to this setting, and thereby obtain Theorem 6, where we give user-friendly sufficient conditions based on second moment information. In Section 4.1.1, we then give a concrete data model for the feature extraction problem. In particular, the feature extraction problem under this data model can be solved by SILS'-SDP due to Theorem 6. Next,

in Section 4.2, we consider the integer sparse recovery problem. We present Theorem 8, which is obtained by specializing Theorem 3 to this problem. We then consider two concrete data models for the integer sparse recovery problem, which can be solved by SILS'-SDP. The first model, presented in Section 4.2.1, has a high coherence, while the second model, in Section 4.2.2, has a low coherence.

We note that, we will prove that SILS'-SDP works well for several probabilistic models, by showing that if the number of data points n is large enough, SILS'-SDP recovers a specific x^* with high probability. However, discussion on sample complexity is not the main focus of this paper. All these illustrations are intended to showcase the power and flexibility of SILS'-SDP solving SILS'.

Before introducing the results, we first give notation of probability that we will use in the remainder of the paper. A random vector $X \in \mathbb{R}^d$ is *centered* if $\mathbb{E}(X) = 0_d$. We denote the Gaussian distribution with mean θ and covariance Σ by $\mathcal{N}(\theta, \Sigma)$. We say a random variable $X \in \mathbb{R}$ is *sub-Gaussian* with parameter L if $\mathbb{E}\exp\{t(X - \mathbb{E}X)\} \leq \exp\left(t^2L^2/2\right)$, for every $t \in \mathbb{R}$, and we write $X \sim \mathcal{SG}(L^2)$. We say a centered random vector $X \in \mathbb{R}^d$ is *sub-Gaussian* with parameter L if $\mathbb{E}\exp\left(tX^\top x\right) \leq \exp\left(t^2L^2/2\right)$, for every $t \in R$ and for every x such that $\|x\|_2 = 1$. With a little abuse of notation, we also write $X \sim \mathcal{SG}(L^2)$. For more details, and for properties of sub-Gaussian random variables (or vectors), we refer readers to [52].

4.1 Feature extraction problem with sub-Gaussian data

In this section, we consider the feature extraction problem, and we assume that M and ϵ have sub-Gaussian entries. Recall that the feature extraction problem is Problem SILS', where (LM) holds (for a general vector z^*).

We now present our sufficient conditions for solving the feature extraction problem with sub-Gaussian data. We note that, to the best of our knowledge, Theorem 6 provides the first known sample complexity bound for solving feature extraction problem in polynomial time.

Theorem 6. *Let $x^* \in \{0, \pm 1\}^d$, define $S := \mathrm{Supp}(x^*)$, and assume $|S| = \sigma$. Assume* (LM) *holds. In addition, suppose that M consists of centered row vectors $m_i \overset{i.i.d.}{\sim} \mathcal{SG}(L^2)$ for some $L > 0$ and $i \in [n]$, and we denote the covariance matrix of m_i by Σ. Assume the noise vector ϵ is a centered sub-Gaussian random vector independent of M, with each $\epsilon_i \overset{i.i.d.}{\sim} \mathcal{SG}(\varrho^2)$ for $i \in [n]$. Let the constants c_1, B, B_1, B_2 be the same as in Lemma 6. Define $\hat{y}^* := -\Sigma z^*$, $\hat{Y}^*_{11} := -(\hat{y}^*_S)^\top x^*_S$, $\hat{\theta} := \arccos\left(\frac{(\hat{y}^*_S)^\top x^*_S}{\sqrt{\sigma}\|y^*_S\|_2}\right)$, and assume $\hat{Y}^*_{11} > 0$ and $\frac{1}{\sigma}\hat{Y}^*_{11} = \Omega(1)$. Suppose there exist $\delta > 0$ such that the following conditions are satisfied:*

- ***D1.*** *The function $f_n(x) := \sqrt{\frac{\|x\|_2^2}{(x^\top x^*_S)^2} - \frac{1}{\sigma}}$ is $\frac{\ell_n}{\sqrt{\sigma}}$-Lipschitz continuous at the point $\hat{y}^*_S$ for some constant ℓ_n;*
- ***D2.*** *$\left\|\Sigma_{S,S^c} + \frac{1}{\sigma}\hat{y}^*_{S^c}(x^*_S)^\top\right\|_\infty + BL^2\sqrt{\log(d)/n} + \frac{1}{\sigma}\lambda_n \leq \hat{\mu}^*_3$ holds, where $\lambda_n := B_2L\sqrt{(\varrho^2 + L^2\|z^*\|_2^2)\log(d)/n}$ and $\hat{\mu}^*_3 := \frac{1}{\sigma}\left\{\lambda_{\min}\left(\Sigma_{S,S}\right) - \delta + \min_{i\in S}\frac{-\hat{y}^*_i - (\Sigma x^*)_i}{x^*_i} - \lambda_n - B_1L^2\sqrt{\frac{\sigma\log(d)}{n}} - c_1L\sqrt{\frac{\sigma}{n}}\right\}$;*
- ***D3.*** *There exists $\hat{\mu}^*_2 \in (-\infty, -\lambda_{\min}\left(\Sigma_{S,S}\right) - c_1L\sqrt{\frac{\sigma}{n}} + \delta]$ such that the inequality $\|\Sigma_{S^c,S^c} + \hat{\mu}^*_2 I_{d-\sigma}\|_\infty + BL^2\sqrt{\frac{\log(d)}{n}} + \frac{\left(\|\hat{y}^*_{S^c}\|_\infty + \lambda_n\right)^2}{\hat{Y}^*_{11} - \sigma\lambda_n} + \gamma_n/\delta \leq \hat{\mu}^*_3$ holds, where $\gamma_n := (f_n(\hat{y}^*_S) + \ell_n\lambda_n)^2\left(\|\hat{y}^*_{S^c}\|_\infty + \lambda_n\right)^2/\delta$.*

Then, there exists a constant $C = C(\Sigma, z^*, x^*, \sigma)$ such that when $n \geq CL^2(\varrho^2 + L^2\|z^*\|_2^2 + \sigma)\log(d)$, SILS'-SDP recovers x^* w.h.p. as $(n, \sigma, d) \to \infty$.

Remark. Condition **D1** guarantees that, the function $\frac{1-\cos^2(\theta)}{\sigma\cos^2(\theta)}$ in **B2** in Theorem 4, is sufficiently smooth, and it is not a very restrictive assumption. In fact, in some cases, it can be easily fulfilled. For example, in the case where $x_S^* = \mathrm{sign}(\hat{y}_S^*)$, the assumption $\frac{1}{\sigma}\hat{Y}_{11}^* = \frac{1}{\sigma}(-\hat{y}_S^*)^\top x_S^* = \Omega(1)$ in Theorem 6 guarantees condition **D1**. Indeed, we see

$$\nabla_i f(x) = \frac{1}{2\sqrt{\frac{\|x\|_2^2}{[x^\top x_S^*]^2} - \frac{1}{\sigma}}} \cdot \frac{2x_i[x^\top x_S^*]^2 - 2x_i^*[x^\top x_S^*]\,\|x\|_2^2}{[x^\top x_S^*]^4} = \frac{x_i[x^\top x_S^*] - x_i^*\,\|x\|_2^2}{[x^\top x_S^*]^3\sqrt{\frac{\|x\|_2^2}{[x^\top x_S^*]^2} - \frac{1}{\sigma}}},$$

and hence $\|\nabla f_n(x)\|_2 = \frac{\sqrt{\sigma}\|x\|_2}{[x^\top x_S^*]^2}$. Using Taylor's expansion, there exists some $\eta \in [0,1]$ such that $|f_n(\hat{y}_S^*) - f_n(\hat{y}_S^*)| \le \|\nabla f_n(\hat{y}_S^* + \eta(\hat{y}_S^* - y_S^*))\|_2\,\|\hat{y}_S^* - y_S^*\|_2$. As long as $\|y_S^* - \hat{y}_S^*\|_\infty$ is sufficiently small such that $\mathrm{sign}(y_S^*) = \mathrm{sign}(\hat{y}_S^*)$ and $\frac{1}{\sigma}[\hat{y}_S^* + \eta(\hat{y}_S^* - y_S^*)]^\top x_S^* = \Omega(1)$, we have

$$\|\nabla f_n(\hat{y}_S^* + \eta(\hat{y}_S^* - y_S^*))\|_2 = \frac{\sqrt{\sigma}}{|[\hat{y}_S^* + \eta(\hat{y}_S^* - y_S^*)]^\top x_S^*|} \cdot \frac{\|\hat{y}_S^* + \eta(\hat{y}_S^* - y_S^*)\|_2}{\|\hat{y}_S^* + \eta(\hat{y}_S^* - y_S^*)\|_1} = \mathcal{O}(\frac{1}{\sqrt{\sigma}}),$$

and hence we obtain **D1**.

In the opposite case, where $x_S^* \neq \mathrm{sign}(\hat{y}_S^*)$, some additional but realistic conditions can be assumed to guarantee **D1**. A possible case is that the function $g(x) := \frac{\|x\|_2}{|x^\top x_S^*|}$ is upper bounded by some absolute constant $c > 0$ at $x = \hat{y}_S^*$, and $\min_{i\in S}|\hat{y}_i^*| = \Omega(1)$. Intuitively, the first assumption is equivalent to saying that the unit direction vector of $\hat{y}_S^*$ is not nearly orthogonal to x_S^*, and the second assumption is equivalent to saying that the vector $\Sigma_{S,[d]}z^*$ is bounded away from zero. Since $\min_{i\in S}|\hat{y}_i^*| = \Omega(1)$, when $\|y_S^* - \hat{y}_S^*\|_\infty$ is sufficiently small, then $[\hat{y}_S^* + \eta(\hat{y}_S^* - y_S^*)]^\top x_S^* \ge \frac{1}{2}|(\hat{y}_S^*)^\top x_S^*|$ and $\|\hat{y}_S^* + \eta(\hat{y}_S^* - y_S^*)\|_2 \le 2\,\|\hat{y}_S^*\|_2$ hold. Combining the assumption $\frac{1}{\sigma}\hat{Y}_{11}^* = \Omega(1)$, we obtain **D1** from the fact

$$\begin{aligned}\|\nabla f_n(\hat{y}_S^* + \eta(\hat{y}_S^* - y_S^*))\|_2 &= \frac{\sqrt{\sigma}}{|[\hat{y}_S^* + \eta(\hat{y}_S^* - y_S^*)]^\top x_S^*|} \cdot g(\hat{y}_S^* + \eta(\hat{y}_S^* - y_S^*)) \\ &\le \frac{2\sqrt{\sigma}}{|(\hat{y}_S^*)^\top x_S^*|} \cdot 4\frac{\|\hat{y}_S^*\|_2}{|(\hat{y}_S^*)^\top x_S^*|} \le \frac{8c}{\Omega(\sqrt{\sigma})}.\end{aligned}$$

4.1.1 A data model for the feature extraction problem.

In this section, we study a concrete data model for the feature extraction problem and we show that it can be solved by SILS'-SDP with high probability, due to Theorem 6. We now define our first data model, in which the m_i's are standard Gaussian vectors.

Model 1. Assume that (LM) holds, where the input matrix M consists of i.i.d. centered random entries drawn from $\mathcal{SG}(1)$, and where the noise vector ϵ is centered and is sub-Gaussian independent of M, with $\epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{SG}(\varrho^2)$. We assume the ground truth vector z^* satisfies $\|z^*\|_\infty \le u$ for some absolute constant $u > 0$. We additionally assume $|z_1^*| \ge |z_2^*| \ge \cdots \ge |z_d^*|$, and that $|z_\sigma^*| \ge 1 + g$, and $|z_{\sigma+1}^*| < 1$ for some absolute constants $g > 0$. Finally, we assume z^* satisfies

$$\sigma\sum_{i=1}^{\sigma}|z_i^*|^2 \le \left(\frac{g^2}{2(g+1)} + 1\right)\left(\sum_{i=1}^{\sigma}|z_i^*|\right)^2 \tag{8}$$

Model 1 can be viewed as follows: M is a normalized real-world sub-Gaussian data matrix (for each entry of the real-world data matrix, we subtract the column mean and then divide by the column standard deviation) with independent columns, and z^* is a feature vector, with the σ most significant features having "feature significance" that is at least $g > 0$ more than those $d - \sigma$ less significant features. Lastly, (8) can be seen as a reversed Cauchy-Schwarz inequality, which guarantees that the most significant σ components do not "spread" too far away from

each other. One can see that (8) holds if g is sufficiently large. In computer vision, we can view a Gaussian M as an image, which is a simplified yet natural assumption [41], and we view the vector z^* as the relationship among the center pixel and the pixels around [54]. It is worthy pointing out that, existing algorithms generally take an exponential running time [4, 59] due to the fact that z^* is not sparse.

Note that, in Model 1, it is not realistic to assume that the largest components of z^* are all in the first σ components. Rather, we should consider the more general model where the components of z^* are arbitrarily permuted. However, this assumption on z^* in the model can be made without loss of generality. In fact, SILS'-SDP can solve Model 1 if and only if it can solve the more general model. This is because both SILS'-SDP and the model are invariant under permutation of variables. A similar note applies to Models 2 and 3 that will be considered later. In addition, the assumption that all less significant features are less than or equal to one can is true, if one scale properly the input (M, b), at the cost of a scaling of noise variance ρ.

In our next theorem, we present that SILS'-SDP solves SILS' with high probability provided that n is sufficiently large. The numerical performance of SILS'-SDP under Model 1 will be demonstrated and discussed in Appendix G.2.1, and the proof of Theorem 7 is left in Appendix D.

Theorem 7. *Consider the feature extraction problem under Model 1. Then, there exists an absolute constant C such that when $n \geq C\big(\sigma^2+d+\varrho^2\big)\log(d)$, SILS'-SDP solves SILS' w.h.p. as $(n,d) \to \infty$.*

In the proof of Theorem 7, we actually showed that, if $n \geq C\big(\sigma^2 + d + \varrho^2\big)\log(d)$, then SILS'-SDP solves SILS' by recovering a special x^*, which is supported on $[\sigma]$. As we will see in Appendix G.2.1, we observe from numerical tests that SILS'-SDP solves SILS' even for smaller values of n, and the recovered sparse integer vector is not necessarily supported on $[\sigma]$. A possible explanation of this phenomenon is that, the upper bounds used in the proof for random variables can be large when n is not sufficiently large. The terms related to n in conditions **D2** - **D3** in Theorem 6 will no longer vanish and may become the dominating terms, causing the support set S of the optimal solution to possibly change.

4.2 Integer sparse recovery problem

In the realm of communications and signal processing, reconstruction of sparse signals has become a prominent and essential subject of study. In this section, we aim to solve the integer sparse recovery problem. Recall that, in this problem, our input M, b, σ satisfies (LM), for some $z^* \in \{0, \pm 1\}^d$ with cardinality σ, and our goal is to recover z^* correctly. As mentioned in Section 1, assuming $z^* \in \{0, \pm 1\}^d$, solving the integer sparse recovery problem is equivalent to solving the well-known sparse recovery problem.

We first give sufficient conditions for SILS'-SDP to recover z^*. For brevity, we denote by $H^0 := I_\sigma - z^*_S(z^*_S)^\top/\sigma$ and define

$$
\begin{aligned}
\Theta :=\ & \frac{1}{n}(M^\top M)_{S^c,S^c} - \frac{(M^\top\epsilon)_{S^c}(y^*_S)^\top}{\delta n Y^*_{11}}H^0\big(I_\sigma + \frac{y^*_S(z^*_S)^\top}{Y^*_{11}}\big)\frac{1}{n}(M^\top M)_{S,S^c} - \frac{1}{Y^*_{11}}(\frac{1}{n}M^\top\epsilon)_{S^c}(\frac{1}{n}M^\top\epsilon)^\top_{S^c} \\
& - \frac{1}{n}(M^\top M)_{S^c,S}\big(I_\sigma + \frac{z^*_S(y^*_S)^\top}{Y^*_{11}}\big)H^0\frac{y^*_S(M^\top\epsilon)^\top_{S^c}}{\delta n Y^*_{11}} - \frac{1}{\delta(nY^*_{11})^2}(M^\top\epsilon)_{S^c}(y^*_S)^\top H^0 y^*_S(M^\top\epsilon)^\top_{S^c} \\
& - \frac{1}{Y^*_{11}}(\frac{1}{n}M^\top\epsilon)_{S^c}\big(\frac{1}{n}(M^\top M)_{S^c,S}z^*_S\big)^\top - \frac{1}{Y^*_{11}}\big(\frac{1}{n}(M^\top M)_{S^c,S}z^*_S\big)(\frac{1}{n}M^\top\epsilon)^\top_{S^c} \\
& - \frac{1}{n^2}(M^\top M)_{S^c,S}\Big([I_\sigma + \frac{1}{Y^*_{11}}z^*_S(y^*_S)^\top]\frac{1}{\delta}H^0[I_\sigma + \frac{1}{Y^*_{11}}y^*_S(z^*_S)^\top] + \frac{z^*_S(z^*_S)^\top}{Y^*_{11}}\Big)(M^\top M)_{S,S^c} + \mu^*_2 I_{d-\sigma}.
\end{aligned}
\tag{9}
$$

In light of Theorem 3 and the linear model assumption (LM), we are able to derive the following sufficient conditions for recovering z^*.

Theorem 8. *Consider the integer sparse recovery problem. We denote* $S := \mathrm{Supp}(z^*)$, $y^* := -M^\top b/n$, $Y_{11}^* := -(y_S^*)^\top z_S^*$, *and assume* $Y_{11}^* > 0$. *Then SILS'-SDP recovers* z^*, *if there exists a constant* $\delta > 0$ *such that the following conditions are satisfied:*

E1. $\frac{1}{n\sigma}\left\|(M^\top \epsilon)_{S^c}\right\|_\infty \le \mu_3^* := \frac{1}{\sigma}\{\lambda_{\min}\big(\frac{1}{n}(M^\top M)_{S,S}\big) - \delta + \min_{i\in S}(\frac{1}{n}M^\top\epsilon)_i/z_i^*\}$;

E2. *There exists* $\mu_2^* \in (-\infty, -\lambda_{\min}\big(\frac{1}{n}(M^\top M)_{S,S}\big) + \delta]$ *such that the matrix* Θ *defined in* (9) *can be written as the sum of two matrices* $\Theta_1 + \Theta_2$, *with* $\Theta_1 \succ 0$, $\|\Theta_2\|_\infty \le \mu_3^*$ *or* $\Theta_1 \succeq 0$, $\|\Theta_2\|_\infty < \mu_3^*$.

Proof. We intend to use Theorem 3 with $x^* = z^*$, hence we need to prove that conditions **E1** - **E2** imply **A1** - **A2**. Recall that we have $b = Mz^* + \epsilon$, and $|S| = |\,\mathrm{Supp}(z^*)| = \sigma$. To show **A1**, we only need to observe that $-y^* - \frac{1}{n}(M^\top M)_{S,S}x_S^* = \frac{1}{n}M^\top(Mx_S^* + \epsilon) - \frac{1}{n}(M^\top M)_{S,S}x_S^* = \frac{1}{n}M^\top\epsilon$, so **A1** coincides with **E1** in this setting. Then, a direct calculation shows that Θ in this theorem coincides with the one in Theorem 3 by expanding $y_{S^c}^*$. □

We observe that the assumptions in Theorem 8 do not imply that $M^\top M$ has a low coherence, the RIP, the NSP, or any other property which guarantees that Lasso or Dantzig Selector solve the sparse recovery problem. On the contrary, our proposed SDP relaxation SILS'-SDP is capable of solving instances with high coherence, whereas not possible via Lasso or Dantzig Selector. This will be evident from our computational results in Section 5.2.

Remark. The assumptions of Theorem 8 can be easily fulfilled in some scenarios. We start by claiming that **E1** is essentially weak and natural. It is met in the case where ϵ is a random noise vector independent of M when n is large, and $\lambda_{\min}((M^\top M/n)_{S,S})$ is lower bounded by some positive constant. In addition, **E1** is quite similar to the constraint in the definition of Dantzig Selector DS, but here we only require this type of constraint for the S^c block of $M^\top\epsilon/n$. Next, Condition **E2** asks to construct Θ_1 in a way such that $\|\Theta_2\|_\infty$ is small. Note that, although **E2** is complicated and sometimes it can be challenging to give such decomposition of Θ, this assumption holds in an ideal scenario, where $(M^\top M)_{S^c,S^c}$ is large enough such that $\lambda_{\min}(\Theta) \ge 0$.

4.2.1 A data model with a high coherence for the integer sparse recovery problem

In this section, we introduce a data model for the integer sparse recovery problem that admits high coherence. The reason why we look into data models with high coherence is straightforward: by Theorem 5 and Section 3.2, SILS'-SDP is not expected to misidentify a certain active user with a silent user in the case where they both have low correlation, i.e., in the low coherence case. Hence, one may ask whether SILS'-SDP tend to make mistake when data coherence becomes higher. We will present that our SDP relaxation SILS'-SDP can solve the integer sparse recovery problem under a simple yet fundamental high coherence model with high probability, as a consequence of Theorem 8. To be concrete, we study the following data model.

Model 2. Assume that (LM) holds, where the rows $m_1, m_2, \cdots, m_n$ of the input matrix M are random vectors drawn from i.i.d. $\mathcal{N}(0_d, \Sigma)$, with

$$\Sigma := \begin{pmatrix} cI_\sigma & 1_\sigma 1_{d-\sigma}^\top \\ 1_{d-\sigma}1_\sigma^\top & c'\sigma 1_{d-\sigma}1_{d-\sigma}^\top \end{pmatrix} + \begin{pmatrix} O_\sigma & \\ & c''I_{d-\sigma} \end{pmatrix} := \Sigma_1 + \Sigma_2$$

for $c > 1$, $c' > 1$ and $c'' > 0$. The ground truth vector is $z^* = \begin{pmatrix} a \\ 0_{d-\sigma} \end{pmatrix}$, with $a \in \{\pm 1\}^\sigma$, and the noise vector ϵ is centered and is sub-Gaussian independent of M, with $\epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{SG}(\varrho^2)$.

We can interpret Model 2 as follows: the first σ independent variables (active users) send out signal a, while the remaining variables (silent users) do nothing. Those $d - \sigma$ silent users have high correlations with the active ones, and even higher correlations among themselves.

The part explained by Σ_2 states that the silent users are not the same, so the model does not reduce to a trivial model in which repeated users are involved in the data set.

Though it might be a bit simplified and restrictive, Model 2 is in fact a baseline model for us to understand how algorithms perform under a data model with a high coherence. A perceptual reasoning is that, one can always split a set of variables into two groups having the following property: group 1 has variables with a covariance matrix that admits a low coherence; and once any one of variables in group 2 is added to group 1, the corresponding covariance matrix of group 1 will admit a high coherence. In Model 2, we can assign the first σ active users to group 1, and assign the remaining $(d-\sigma)$ highly correlated silent users to group 2. In particular, we study the simplest case, where correlations among two different users in the same group are exactly the same, and where correlations among two users in different groups are also exactly the same. We further limit our focus to the case when users in group 1 are independent, i.e., two different users in group 1 have correlation zero, in order to quickly verify that the proposed model is valid, i.e., the covariance matrix Σ is positive semidefinite. Indeed, Lemma 4 in Appendix B and the fact that $\Sigma_{S^c,S^c} \succeq (\sigma/c)1_\sigma 1_{d-\sigma}^\top$ together imply that $\Sigma \succeq 0$.

As Model 2 is a model with highly correlated users, and $\mu(M^\top M) = \Omega(1)$ when n is sufficiently large, we see that Model 2 does not have a low coherence. Moreover, Model 2 does not satisfy the mutual incoherence property, since $\|(M^\top M)_{S^c,S}(M^\top M)_{S,S}^{-1}\|_{\infty\to\infty} = \Omega(\sigma) > 1$ when n is sufficiently large. The above two facts follow from **6A** and **6B** in Lemma 6. The aforementioned properties are known to be crucial for ℓ_1-based convex relaxation algorithms like Dantzig Selector and Lasso to recover z^*. Though the intuition behind Model 2 may seem naive, we find that numerically, these two algorithms indeed give a high prediction error in this model, as we will discuss in Section 5.2. However, the following theorem shows that our semidefinite relaxation SILS'-SDP can recover z^* with high probability.

Theorem 9. *Consider the integer sparse recovery problem under Model 2. Then, there exists a constant $C = C(c, c', c'')$ such that when $n \geq C\sigma^2\varrho^2\log(d)$, SILS-SDP recovers z^* w.h.p. as $(n, \sigma, d) \to \infty$.*

The proof of Theorem 9 is given in Appendix E and the numerical performance of SILS-SDP under Model 2 is presented in Section 5.2.

To the best of our knowledge, the optimal sample complexity for solving the integer sparse recovery problem under Model 2 in polynomial time remains unexplored, and we introduce the first bound. The only known theoretical results on sparse recovery problem with models with a high coherence are presented in [16]. The authors proposed an algorithm known as *Structured Iterative Hard Thresholding (IHT) algorithm* for general sparse recovery problems where additional structures of sparsity are recognized. This includes a division of the index set $[d]$ into partitions $S_1, S_2, \ldots, S_p$ and a corresponding partition of the sparsity level σ into p positive integers $\sigma_1, \sigma_2, \ldots, \sigma_p$, such that $\sigma = \sum_{i=1}^p \sigma_i$. Their results demonstrate that if $\mu(M_{[d],S_i}^\top M_{[d],S_i}) \leq 1/(3\sigma_i)$, the Structured IHT Algorithm achieves linear convergence. Additionally, the solution approximates z^*, apart from some residual additive error, as detailed in Theorem 3.3 and Corollary 3.6 in their paper.

It should be noted that even when one assumes that the addictive error is small in Model 2, exact recovery of z^* through this algorithm remains theoretically unknown unless specific assignments are made, such as setting some S_i to be $[\sigma]$ and $\sigma_i = \sigma$, as having an index $j > \sigma$ in S_i immediately results in $\mu(M_{[d],S_i}^\top M_{[d],S_i}) = \Omega(1)$. This implies that their algorithm obtains a recovery of z^* only under the very strong assumption that one has the information of $\mathrm{Supp}(z^*)$. We also note that [1] proposes a similar approach for compressed sensing, and the setups of problems are not the same. To be more specific, users are allowed to obtain samples as any linear measurements on Uz^* given a basis matrix V, where U and V are all orthogonal bases of $\mathbb{C}^d$, requiring extra structural properties on inputs M and b that are not satisfied by Model 2. For further details, we refer the interested readers to the paper.

4.2.2 A data model with a low coherence for the integer sparse recovery problem

In this section, we show that SILS'-SDP can solve the integer sparse recovery problem also under some low coherence data models. Here, we focus on the following data model, which is a generalized version of the model studied in [43].

Model 3. Assume that (LM) holds, where the input matrix M consist of i.i.d. random entries drawn from $\mathcal{SG}(1)$, the ground truth vector is $z^* = \begin{pmatrix} a \\ 0_{d-\sigma} \end{pmatrix}$, with $a \in \{\pm 1\}^\sigma$, and the noise vector ϵ is centered and is sub-Gaussian independent of M, with $\epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{SG}(\varrho^2)$.

From **6A** and **6B** in Lemma 6, we can see that when $n = \Omega(\sigma^2 \log(d))$, the mutual incoherence property holds in Model 3, i.e., $\|(M^\top M)_{S^c,S}(M^\top M)_{S,S}^{-1}\|_{\infty\to\infty} < 1$. At the same time, Model 3 admits a low coherence, so it is known that algorithms like Lasso and Dantzig Selector can recover z^* efficiently [53, 33]. As a similar result, we show in the next theorem that SILS'-SDP can recover z^* when $n = \Omega((\sigma^2 + \varrho^2)\log(d))$. The proof of the theorem is left in Appendix F. While this result can be proven using Theorem 5 or Theorem 8, in our proof we use Theorem 6 instead. This is because, although Theorem 6 is tailored to the feature extraction problem, it leads to a cleaner proof. In Appendix G.2.3, we will demonstrate the numerical performance of SILS'-SDP under Model 3 and we will compare it with Lasso and DS.

Theorem 10. *Consider the integer sparse recovery problem under Model 3. There exists an absolute constant C such that when $n \geq C\big(\sigma^2 + \varrho^2\big)\log(d)$, SILS'-SDP recovers z^* w.h.p. as $(n, d) \to \infty$.*

To the best of our knowledge, the best sample complexity for recovering z^* in polynomial time is proposed by [37]. The authors show that it is possible to recover z^* efficiently when the entries of M are i.i.d. standard Gaussian random variables with sample complexity $n = \Omega(\sigma \log(ed/\sigma) + \varrho^2 \log(d))$. In Theorem 10, we show that we need $n = \Omega((\sigma^2 + \varrho^2)\log(d))$ many samples. The differences between these results are that: (1) we recover the integer vector z^* exactly, while [37] recovers an estimator of z^*; (2) our method is more general, since theirs may not extend to the sub-Gaussian setting. We view the difference in sample complexity as a trade-off to obtain integrality in a more general setting.

5 Numerical tests

In this section, we discuss the numerical performance of our SDP relaxations SILS-SDP and SILS'-SDP. We first report the algorithmic performance of Algorithm 1, given an (approximate) optimal solution to SILS-SDP, under some synthetic datasets including Models 2 and 3, and some large real-world datasets introduced in [15, 19]. We compare the performance of Algorithm 1 with SBQP and the MIO formulation (defined in MIO) in [6].

Then, we report the statistical performance of SILS'-SDP under Model 2, and we defer detailed statistical performance under Models 1 and 3 to Appendix G.2. For comparisons made in these statistical models, we do not include comparisons with SBQP and MIO for two principal reasons: First, SILS'-SDP is a relaxation specifically designed for both SBQP and MIO, and they all obtain the same solution in our instances since SILS'-SDP admits an integer optimal solution. This makes a comparison with methods that yield identical outcomes redundant. Second, our focus is to assess the statistical performance of other existing polynomial time algorithms that incorporate ℓ_1 constraints, exploring the class of inputs that lead to significant differences. This approach addresses key questions in the fields of sparse recovery and compressed sensing. Therefore, we report the numerical performance of SILS'-SDP under the data models which are studied in Section 4, and compare the statistical performance of SILS'-SDP with other known convex relaxation algorithms.

Unless specified, solutions to convex programs are obtained via CVX v2.2, a package for solving convex optimization problems [26] implemented in Matlab, with Mosek 9.2 [3] as its solver. All Mixed Integer quadratic programs are solved via Gurobi 10.0 [28] with its Matlab interface. We conducted all tests on a computing cluster equipped with 36 Cores (2x 3.1G Xeon Gold 6254 CPUs) and 768 GB of memory. We leave the computational details and additional empirical results in Appendix G.

5.1 Algorithmic performance

In this section, we present the algorithmic performance of SILS-SDP by summarizing the numerical test results under various datasets of Algorithm 1 - our proposed randomized algorithm in Section 2. Since Mosek faces scalability issue with these datasets, we instead obtain an approximate optimal solution to SILS-SDP with the Conditional Gradient Augmented Lagrangian (CGAL) framework, as proposed by [56]. Additionally, we evaluate against SBQP, and the Mixed-Integer Optimization (MIO) formulation from [6]. MIO has demonstrated empirical success in general least squares problems with sparsity constraints. We solve SBQP and MIO by the following quadratic integer programs:

$$\begin{aligned} \min \quad & x^\top M^\top M x - 2b^\top M x \\ \text{s.t.} \quad & |x_i| \le z_i, \quad i \in [d], \\ & \sum_{i=1}^{d} z_i \le \sigma, \\ & x \in \{0, \pm 1\}^d, \\ & z \in \{0,1\}^d. \end{aligned} \tag{SBQP}$$

$$\begin{aligned} \min \quad & x^\top M^\top M x - 2b^\top M x \\ \text{s.t.} \quad & |x_i| \le z_i, \quad i \in [d], \\ & \sum_{i=1}^{d} z_i \le \sigma, \\ & x \in \mathbb{R}^d, \\ & z \in \{0,1\}^d. \end{aligned} \tag{MIO}$$

Note that MIO is equivalent to equations (2.4) and (2.5) in [6].

We employ Algorithm 1 from [6] with a $\mathcal{N}(0, I_d)$ vector as a warm-start for MIO and randomly generate a $\{0, \pm 1\}^d$ with support size σ as a warm-start of SBQP.

We generate the data input as follows: for a data matrix $M \in \mathbb{R}^{n \times d}$, we randomly generate a sparse vector $z^* \in \{0, \pm 1\}^d$ with $\|z^*\|_0 = k$, and set $b := Mz^* + \epsilon$ for some noise vector $\epsilon \in \mathbb{R}^n$. We leave the detailed specifications for the datasets in Appendix G.1.2.

We report the *relative gaps* of each method, with SBQP serving as the baseline. Specifically, let $\text{obj}_{z^*} := (z^*)^\top M^\top M z^* - 2b^\top M z^*$, and define the relative gap of an algorithm as

$$\text{relative gap} := \frac{\text{obj}_{\text{Alg}} - \text{obj}_{z^*} + 1}{\text{obj}_{SBQP} - \text{obj}_{z^*} + 1}.$$

Here, $\text{obj}_{\text{Alg}} := x^\top M^\top M x - 2b^\top M x$, where x is a solution obtained by a specific algorithm. We use obj_{z^*} as a reference, since in practice we observe that these three algorithms often fail to find solutions with objective values strictly less than obj_{z^*}. Therefore, $\text{obj}_{\text{Alg}} - \text{obj}_{z^*}$ serves as a lower bound for the optimality gap across all three algorithms. Finally, we add one to both the enumerator and the denominator to avoid division by zero.

In Table 1, we summarize the computational results by providing the average relative gap and runtime across various datasets. We run Algorithm 1 with $T = \sqrt{\log d}$ and $C = 0.05$ for 1000 iterations, followed by a simple greedy algorithm to improve the performance of Algorithm 1. This greedy algorithm first finds the set of indices $S \subseteq [d]$ corresponding to the indices of the largest σ p_i's in Algorithm 1, and then finds a solution x by assigning $x_i = \text{sign}((w_x)_i)$ if $i \in S$, and 0 otherwise. The motivation behind the greedy algorithm is to find a feasible heuristic solution with cardinality σ such that it maximizes the probabilistic "likelihood" that

Algorithm 1 would pick. We report the best relative gap obtained by these 1001 solutions in the column "CGAL + Algorithm 1". It is clear that CGAL + Algorithm 1 oftentimes obtains better solutions in less computational time, showcasing its efficacy not only in statistical models but also in real-world datasets.

Table 1: Average Relative Gaps and Running Times. Time limits for SIQP and MIO are set to 1000 seconds.

Dataset	SIQP		MIO		CGAL + Algorithm 1	
	Rel. Gap	Time (s)	Rel. Gap	Time (s)	Rel. Gap	Time (s)
Example 1 in [6]	1	658.8	0.4508	649.9	**0.4288**	348.7
Model 2	1	904.1	1.9664	1003.7	**0.3021**	378.7
Model 3	1	878.6	0.4107	983.8	**0.2549**	293.1
Diabete	**1**	0.0733	**1**	0.0957	**1**	0.1467
Leukemia	1	379.75	0.6411	387.0	**0.5513**	111.75
Prostate	1	1002.75	11.7628	1002.0	**0.5791**	318.75

We leave the computational details in Appendix G.1, and performance of Algorithm 1 for non-convex objectives in Appendix G.3.

5.2 Statistical performance

In this part, we show how SILS'-SDP performs numerically in the integer sparse recovery problem under Model 2, as studied in Section 4.2.1. We compare the statistical performance of SILS'-SDP with Lasso and Dantzig selector, which are defined by

$$z^{Lasso} := \arg\min \frac{1}{2n}\|Mz-b\|^2 + \lambda\|z\|_1, \quad \text{(Lasso)} \qquad z^{DS} := \underset{\|M^\top(Mz-b)\|_\infty \le \eta}{\arg\min} \|z\|_1, \quad \text{(DS)}$$

where λ and η are user-specified parameters.

In Model 2, we take $c = 1.2$, $c' = 1.05$, and $c'' = 1$ in the covariance matrix Σ, and we take $\epsilon \sim \mathcal{N}(0_d, \varrho^2 I_d)$. We restrict ourselves to the setting where $z^* = \begin{pmatrix} 1_\sigma \\ 0_{d-\sigma} \end{pmatrix}$, and we compare the performance of SILS'-SDP, Lasso, and DS. We are particularly interested in this setting as it is explicitly shown in [53] that Lasso is not guaranteed to perform well. This is still a high coherence model and no guarantee on the performance of Dantzig Selector is known for this model. The parameters λ in Lasso and η in DS are determined via a 10-fold cross-validation on a held out validation set, as suggested in [6]. We report three significant quantities for sparse recovery problems, which evaluate the quality of the solution vector z returned by the algorithm. For SILS'-SDP, the vector z that we evaluate is the vector w^* obtained from the first column of the optimal solution W^* to SILS'-SDP, by deleting its first entry equal to one. The first quantity that we report is the *number of nonzeros*, which is $|\operatorname{Supp}(z)|$ and measures how sparse a solution is. The second quantity that we report is the *true positive rate*, defined as

$$\text{true positive rate}(z) := \frac{|\operatorname{Supp}(z^*) \cap \operatorname{Supp}(z)|}{|\operatorname{Supp}(z^*)|}.$$

This quantity measures how well z recovers the ground truth sparse vector z^* by evaluating how much their support sets overlap. The last quantity that we report, which is suggested in [6], is known as *prediction error*, which is defined as

$$\text{prediction error}(z) := \frac{\|M(z-z^*)\|_2^2}{\|Mz^*\|_2^2}.$$

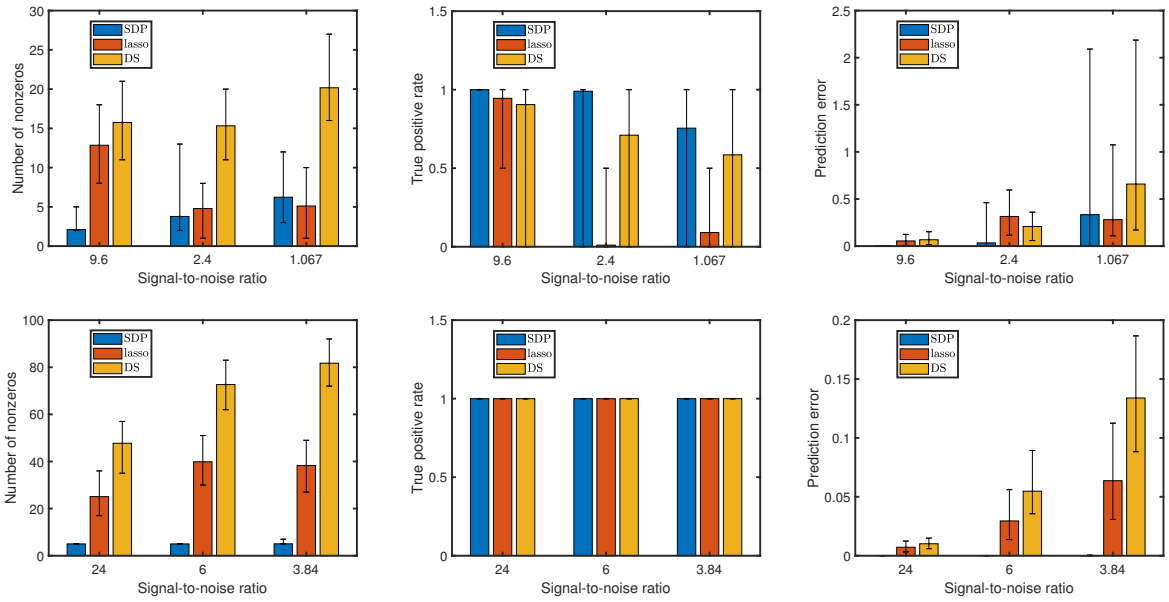

Figure 1: Performance of SILS'-SDP, Lasso, DS under Model 2, with $d = 40$, $\sigma = 2$, $n = \lceil 2\sigma^2 \log(d) \rceil = 30$ in the first row, and with $d = 100$, $\sigma = 5$, $n = \lceil 2\sigma^2 \log(d) \rceil = 231$ in the second row. 100 instances are considered with $\varrho \in \{0.5, 1, 1.5\}$. The average is reported in the histogram, and the minimum and maximum in the box plot.

As discussed in [6], the prediction error takes into account the correlation of features and is a meaningful measure of error for algorithms that do not have performance guarantee. We report these three quantities under different *signal-to-noise ratios*, i.e.,

$$\text{signal-to-noise ratio} := \frac{\text{Var}(m_i^\top z^*)}{\varrho^2} = \frac{\left\| \Sigma_{[d],S}^{\frac{1}{2}} z_S^* \right\|_2^2}{\varrho^2}.$$

In Figure 1, we study two sets of (d, σ), namely, $(d, \sigma) \in \{(100, 5), (40, 2)\}$, with $\varrho \in \{0.5, 1, 1.5\}$, and we fix our choice of n to be $\lceil 2\sigma^2 \log(d) \rceil$.

In an underdetermined system ($d > n$), plotted in the first row of Figure 1, we conclude that the probability that Lasso and Dantzig Selector recover the true support $[\sigma]$ of z^* is low, while SILS'-SDP nearly always recovers the true support, even when signal-to-noise ratio is low. In an overdetermined system ($d < n$), plotted in the second row of Figure 1, the true positive rates of Lasso and Dantzig Selector dramatically improve, however they are still inferior to SILS'-SDP in terms of number of nonzeros and prediction error.

We remark that, Model 2 is just one example of a high coherence model for the sparse recovery problem under which SILS'-SDP works better than Lasso and DS. For instance, we observe the same behavior in a model introduced in [6] (see Example 1 therein for details). For this model, several methods including Lasso, tend to give a solution with an excessively large support set, and cannot provide a satisfactory prediction error (see Fig. 4. therein for details). On the other hand, for SILS'-SDP, as n grows, the empirical probability of recovery of z^* tends to one, and the conditions in Theorem 8 can be satisfied. We omit the discussion on statistical performance in this model as it shares similar observations as these already made in [6].

We leave the discussions of support recovery of z^*, and detailed computational results in other statistical models in Appendix G.2.

6 Acknowledgements

A. Del Pia and D. Zhou are partially funded by AFOSR grant FA9550-23-1-0433. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Air Force Office of Scientific Research. The authors would love to thank the Associate Editor and anonymous referees for their constructive feedback.

References

[1] Ben Adcock, Anders C Hansen, Clarice Poon, and Bogdan Roman. Breaking the coherence barrier: A new theory for compressed sensing. In *Forum of mathematics, sigma*, volume 5, page e4. Cambridge University Press, 2017.

[2] Arash A Amini and Martin J Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. In *IEEE International Symposium on Information Theory*, pages 2454–2458, 2008.

[3] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.2.*, 2020.

[4] Somsubhra Barik and Haris Vikalo. Sparsity-aware sphere decoding: Algorithms and complexity analysis. *IEEE Transactions on Signal Processing*, 62(9):2212–2225, 2014.

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

[6] Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The annals of statistics*, 44(2):813–852, 2016.

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

[8] Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *Annals of statistics*, 35(6):2313–2351, 2007.

[9] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[10] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: Extending grothendieck's inequality. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 54–60. IEEE, 2004.

[11] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[12] Elaine Crespo Marques, Nilson Maciel, Lírida Naviner, Hao Cai, and Jun Yang. A review of sparse recovery algorithms. *IEEE Access*, 7:1300–1322, 2019.

[13] Alexandre d'Aspremont, Laurent Ghaoui, Michael Jordan, and Gert Lanckriet. A direct formulation for sparse pca using semidefinite programming. *Advances in neural information processing systems*, 17, 2004.

[14] Etienne de Klerk and Frank Vallentin. On the turing model complexity of interior point methods for semidefinite programming. *SIAM Journal on Optimization*, 26(3):1944–1961, 2016.

[15] Marcel Dettling. Bagboosting for tumor classification with gene expression data. *Bioinformatics*, 20(18):3583–3593, 2004.

[16] Joseph S Donato and Howard W Levinson. Structured iterative hard thresholding with on-and off-grid applications. *Linear Algebra and its Applications*, 638:46–79, 2022.

[17] Hongbo Dong, Kun Chen, and Jeff Linderoth. Regularization vs. relaxation: A conic optimization perspective of statistical variable selection. *arXiv preprint arXiv:1510.06083*, 2015.

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

[19] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. 2004.

[20] Axel Flinth and Gitta Kutyniok. Promp: A sparse recovery approach to lattice-valued signals. *Applied and Computational Harmonic Analysis*, 45(3):668–708, 2018.

[21] David Gamarnik and Ilias Zadik. Sparse high-dimensional linear regression. estimating squared error and a phase transition. *The Annals of Statistics*, 50(2):880–903, 2022.

[22] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1990.

[23] Huanmin Ge and Peng Li. The dantzig selector: recovery of signal via ℓ_1 - $\alpha\ell_2$ minimization. *Inverse Problems*, 38(1):015006, 2021.

[24] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[25] Gene H Golub. Some modified matrix eigenvalue problems. *Siam Review*, 15(2):318–334, 1973.

[26] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1, March 2014.

[27] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[28] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022.

[29] Shaoning Han, Andrés Gómez, and Alper Atamtürk. The equivalence of optimal perspective formulation and shor's sdp for quadratic programs with indicator variables. *Operations Research Letters*, 50(2):195–198, 2022.

[30] Sandra Keiper, Gitta Kutyniok, Dae Gwan Lee, and Götz E Pfander. Compressed sensing for finite-valued signals. *Linear Algebra and its Applications*, 532:570–613, 2017.

[31] Harold W Kuhn and Albert W Tucker. Nonlinear programming. In *Traces and emergence of nonlinear programming*, pages 247–258. Springer, 2014.

[32] Monique Laurent and Franz Rendl. Semidefinite programming and integer programming. In K. Aardal, G. Nemhauser, and R. Weismantel, editors, *Handbook on Discrete Optimization*, pages 393–514. Elsevier B.V., December 2005.

[33] Peng Li and Wengu Chen. Signal recovery under mutual incoherence property and oracle inequalities. *Frontiers of Mathematics in China*, 13(6):1369–1396, 2018.

[34] Zang Li and W. Trappe. Collusion-resistant fingerprints from wbe sequence sets. pages 1336 – 1340 Vol. 2, 06 2005.

[35] Karim Lounici. Sup-norm convergence rate and sign concentration property of lasso and dantzig estimators. *Electronic Journal of statistics*, 2:90–102, 2008.

[36] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[37] Mohamed Ndaoud and Alexandre B Tsybakov. Optimal variable selection and adaptive noisy compressed sensing. *IEEE Transactions on Information Theory*, 66(4):2517–2532, 2020.

[38] Jaehyun Park and Stephen Boyd. General heuristics for nonconvex quadratically constrained quadratic programming. *arXiv preprint arXiv:1703.07870*, 2017.

[39] Jaehyun Park and Stephen Boyd. A semidefinite programming method for integer convex quadratic minimization. *Optimization Letters*, 12:499–518, 2018.

[40] Mert Pilanci, Martin J Wainwright, and Laurent El Ghaoui. Sparse learning via boolean relaxations. *Mathematical Programming*, 151(1):63–87, 2015.

[41] Simon J. D. Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, USA, 1st edition, 2012.

[42] Behrooz Razeghi, Slava Voloshynovskiy, Dimche Kostadinov, and Olga Taran. Privacy preserving identification using sparse approximation with ambiguization. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2017.

[43] Galen Reeves, Jiaming Xu, and Ilias Zadik. The all-or-nothing phenomenon in sparse linear regression. In *Conference on Learning Theory*, pages 2652–2663. PMLR, 2019.

[44] M Ross Kunz and Yiyuan She. Multivariate calibration maintenance and transfer through robust fused lasso. *Journal of Chemometrics*, 27(9):233–242, 2013.

[45] Hampei Sasahara, Kazunori Hayashi, and Masaaki Nagahara. Multiuser detection based on map estimation with sum-of-absolute-values relaxation. *IEEE Transactions on Signal Processing*, 65(21):5621–5634, 2017.

[46] Nuno MB Souto and Hugo André Lopes. Efficient recovery algorithm for discrete valued sparse signals using an admm approach. *IEEE Access*, 5:19562–19569, 2017.

[47] Susanne Sparrer and Robert FH Fischer. Adapting compressed sensing algorithms to discrete sparse signals. In *WSA 2014; 18th International ITG Workshop on Smart Antennas*, pages 1–8. VDE, 2014.

[48] Susanne Sparrer and Robert FH Fischer. Soft-feedback omp for the recovery of discrete-valued sparse signals. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 1461–1465. IEEE, 2015.

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

[50] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[51] Roman Vershynin. How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25, 04 2010.

[52] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[53] Martin J Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1 -constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, 2009.

[54] Zuodong Yang, Yong Wu, Wenteng Zhao, Yicong Zhou, Zongqing Lu, Weifeng Li, and Qingmin Liao. A novel illumination-robust local descriptor based on sparse linear regression. *Digital Signal Processing*, 48:269–275, 2016.

[55] Tarik Yardibi, Jian Li, Peter Stoica, and Louis N. Cattafesta III. Sparse representations and sphere decoding for array signal processing. *Digital Signal Processing*, 22(2):253–262, 2012.

[56] Alp Yurtsever, Olivier Fercoq, and Volkan Cevher. A conditional-gradient-based augmented lagrangian framework. In *International Conference on Machine Learning*, pages 7272–7281. PMLR, 2019.

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

[58] Yun-Bin Zhao and Duan Li. A theoretical analysis of sparse recovery stability of dantzig selector and lasso. *arXiv preprint arXiv:1711.03783*, 2017.

[59] Hao Zhu and Georgios B. Giannakis. Exploiting sparse user activity in multiuser detection. *IEEE Transactions on Communications*, 59(2):454–465, 2011.

Appendices

A Proof of Theorem 1

In this section, we prove Theorem 1. To keep aligned with the notations in Section 2, throughout this section, we will keep using the same notations introduced in Algorithm 1 and Theorem 1. Moreover, we will assume the matrix $Q(c,P) = \begin{pmatrix} 0 & -c^\top \\ -c & P \end{pmatrix}$ is 0-indexed, and denote its (i,j)-th entry by q_{ij}, $0 \le i,j \le d$. As we will see later in the proofs, u_0 is in fact a special vector, so it is worthy to distinguish it from $u_1, u_2, \ldots, u_d$, with index zero. In other sections, we will continue to assume all matrices are 1-indexed.

Recall that the problem SDP(c,P) is defined by replacing the objective function $1/n \cdot \mathrm{tr}(A^\top AW)$ by $\mathrm{tr}(Q(c,P)W)$ in SILS-SDP. We first show a nice property about the first column of any feasible solution to SDP(c,P):

Proposition 2. *Consider any feasible solution W to* SDP(c,P). *Let the first column of W be $(1, w_x^\top)^\top$, where $w_x \in \mathbb{R}^d$. Then, $\|w_x\|_1 \le \sigma$.*

Proof. Denote $\mathcal{F}$ to be the feasible region of SDP(c,P), we show that the optimal value of the optimization problem $\max_{W\in\mathcal{F}} \|w_x\|_1$ is exactly σ. By symmetry of $\mathcal{F}$, the problem is equivalent to $\max_{W\in\mathcal{F}} 1_d^\top w_x$. It is clear that by taking $W^* := uu^\top$ with $u := (1, \sigma/d, \sigma/d, \cdots, \sigma/d) \in \mathbb{R}^{1+d}$, we attain a cost of σ in this problem.

We conclude the proof by showing that σ can be attained by its dual. Denote $P_0 := \begin{pmatrix} 0 & 1_d^\top/2 \\ 1_d^\top/2 & O_d \end{pmatrix}$, the primal problem is then equivalent to $\max_{W\in\mathcal{F}} \mathrm{tr}(P_0 W)$, and the dual problem is

$$\min_{\substack{Y\succeq 0, \mu_1\in\mathbb{R}, \mu_2\ge 0, \mu_3 \ge 0 \\ \|P_0 - R(\mu_1,\mu_2,p) + Y\|_\infty \le \mu_3}} \mu_1 + \sigma\mu_2 + \sigma^2\mu_3 + p^\top 1_d,$$

where $R(\mu_1,\mu_2,p) := \begin{pmatrix} \mu_1 & \\ & \mu_2 I_d + p \end{pmatrix}$. It can be checked that the set of dual variables $\mu_1^* := \sigma/2$, $\mu_2^* := 0$, $\mu_3^* := 1/(2\sigma)$, $p^* := 0_d$, and $Y^* := vv^\top$ with $v := \sqrt{\sigma/2}(1, 1/\sigma, 1/\sigma, \cdots, 1/\sigma)^\top \in \mathbb{R}^{1+d}$ is indeed feasible to the dual problem with cost σ. □

The following lemma states some properties regarding some random variables that we introduce in Algorithm 1:

Lemma 2. *Consider Algorithm 1 and the variables therein. Denote $p_0 := 1$, then:*

2A. $\mathbb{E}z_i z_j = u_i^\top u_j$ *for those $0 \le i < j \le d$ such that $p_i, p_j > 0$.*

2B. $\mathbb{E}z_i^2 = \|u_i\|_2^2 / p_i$ *for those $1 \le i \le d$ such that $p_i > 0$.*

2C. $\mathbb{E}x_i^2 = \mathbb{E}|y_i|$ *for $1 \le i \le d$.*

2D. $\mathbb{E}x_i x_j = \mathbb{E}y_i y_j$, *for any $0 \le i < j \le d$.*

2E. *Define $P := \{i \in [d] : p_i > 0\}$, then $|P| \le \min\{d, \sigma/C^2\}$.*

2F. *For those $i, j \in P$, we have that $\mathbb{E}\left[\left|y_i y_j - z_i z_j / T^2\right|\right]$ is upper bounded by*

$$e^{-\frac{2C^2T^2}{9}} \left\{ \frac{2\|u_i\|_2^2 + 2\|u_j\|_2^2}{\sqrt{2\pi}CT} + \|u_i\|_2 \|u_j\|_2 \left[\frac{4}{\sqrt{2\pi}} \cdot \left(\frac{2T}{3} + \frac{3}{2CT} \right) + \frac{4}{\pi} \right] \right\}$$

2G. *For those $j \in P$, we have that $\mathbb{E}\left[\left|y_0 y_j - z_0 z_j / T^2\right|\right]$ is upper bounded by*

$$e^{-\frac{2C^2T^2}{9}} \left\{ \frac{2\left\|u_j\right\|_2^2}{\sqrt{2\pi}CT} + \left\|u_j\right\|_2 \left[\frac{2}{\sqrt{2\pi}} \cdot \left(\frac{2T}{3} + \frac{3}{2CT} \right) + \frac{2}{\pi} \right] \right\}$$
$$+ e^{-\frac{T^2}{2}} \left\{ \frac{2}{\sqrt{2\pi}} \frac{1}{T} + \left\|u_j\right\|_2 \cdot \left[\frac{2}{\sqrt{2\pi}} \left(T + \frac{2}{T} \right) + \frac{1}{\pi} \right] \right\}$$

Proof. **2A**, **2B**, and **2C** follow from direct calculation. We start with **2D**. We first consider the case $i = 0$. We observe that the conditional probability $\mathbb{E}[x_0 x_j | y_0, y_j]$ is exactly $y_0 y_j$. Indeed,

$$\mathbb{E}[x_0 x_j | y_0, y_j] = \left(1 \cdot \frac{1+y_0}{2}\right) \cdot \operatorname{sign}(y_j)|y_j| + \left(-1 \cdot \frac{1-y_0}{2}\right) \cdot \operatorname{sign}(y_j)|y_j| = y_0 y_j,$$

and then by law of total expectation we are done. Then, we assume that $i \geq 1$, and we see that $\mathbb{E}[x_i x_j | y_i, y_j] = (\operatorname{sign}(y_i) \cdot |y_i|) \cdot (\operatorname{sign}(y_j) \cdot |y_j|) = y_i y_j$. By law of total expectation, we again obtain the desired result.

To show **2E**, one only need to observe that $\sigma \geq \sum_{i=1}^d \left\|u_i\right\|_2^2 \geq \sum_{i \in P} \left\|u_i\right\|_2^2 \geq C^2 |P|$.

Finally, **2F** and **2G** follow similarly from the proof of Lemma 2 in [10], and hence we skip the proof here. For a high level idea, we evaluate the expectation of $\left|y_i y_j - z_i z_j / T^2\right|$ conditioned on $\tilde{u}_i$'s similar to [10], and use the facts that $p_i = 2/3 \cdot \left\|u_i\right\|_2^2$, $\left\|u_i\right\|_2 \geq C$, and $\int_t^{+\infty} e^{-x^2/2} \mathrm{d}x < 1/t \cdot e^{-t^2/2}$ in the upper bounds. □

We are now ready to prove Theorem 1:

Proof of Theorem 1. We first show the approximation gap. The second inequality is due to relaxation and ϵ-optimality, and we only need to show the first. We denote $U := (u_0, u_1, \ldots, u_d) = \sqrt{W^*}$, as in Algorithm 1. We observe that $\bar{x}^\top P \bar{x} - 2c^\top \bar{x} = \sum_{i,j=0}^d q_{ij} \bar{x}_i \bar{x}_j$, and $\operatorname{tr}(Q(c,P)W^*) = \sum_{i,j=0}^d q_{ij} u_i^\top u_j$. We will split the proof into two parts:

(i) (Non-diagonal entries, i.e., $i < j$) We first assume $p_i, p_j > 0$, where p_i's are defined in Algorithm 1. By **2A**, **2D**, and the fact that $\bar{x}$ differs x only by possibly flipping a sign in Algorithm 1, we observe that

$$\frac{1}{T^2} \cdot q_{ij} u_i^\top u_j = q_{ij} \mathbb{E} y_i y_j + q_{ij} \left(\frac{1}{T^2} \mathbb{E} z_i z_j - \mathbb{E} y_i y_j \right)$$
$$\geq q_{ij} \mathbb{E} \bar{x}_i \bar{x}_j - |q_{ij}| \cdot \mathbb{E}\left[\left| y_i y_j - z_i z_j \cdot \frac{1}{T^2} \right| \right].$$

For the case where, WLOG, $p_i = 0$. By the definition of p_i in Algorithm 1, it must be the case $\left\|u_i\right\|_2 \leq C$. Therefore, we obtain a trivial bound (note that $\mathbb{E}\bar{x}_i \bar{x}_j = 0$)

$$\frac{1}{T^2} \cdot q_{ij} u_i^\top u_j \geq q_{ij} \mathbb{E} \bar{x}_i \bar{x}_j - \frac{1}{T^2} \cdot |q_{ij}| \cdot |u_i^\top u_j|.$$

(ii) (Diagonal entries, i.e., $i = j$) We first study the case $p_i > 0$ ($i \geq 1$). By **2B**, **2C**, and the facts that $q_{ii} \geq 0$, $\left\|u_i\right\|_2 \geq C$, and $p_i = 2/3 \cdot \left\|u_i\right\|_2^2$, we see that

$$q_{ii} \mathbb{E} \bar{x}_i^2 = q_{ii} \mathbb{E}|y_i| \leq q_{ii} \sqrt{\mathbb{E}|y_i|^2} \leq q_{ii} \sqrt{\mathbb{E} \frac{1}{T^2} |z_i|^2} = \frac{q_{ii}}{T\sqrt{p_i}} \left\|u_i\right\|_2 = \frac{\sqrt{3} q_{ii}}{\sqrt{2} T}$$
$$= \frac{q_{ii}}{T^2} u_i^\top u_i + q_{ii} \left(\frac{\sqrt{3}}{\sqrt{2}T} - \frac{1}{T^2} u_i^\top u_i \right)$$

For the case $p_i = 0$, we again use the trivial inequality

$$\frac{1}{T^2} \cdot q_{ii} u_i^\top u_i \geq q_{ii} \mathbb{E} \bar{x}_i^2 - \frac{1}{T^2} \cdot q_{ii} \cdot u_i^\top u_i.$$

Denote the set $P := \{i \in [d] : p_i > 0\}$ the same as in **2E**, and define $g(C,T) := 1/\sqrt{2\pi} \cdot (2T/3 + 3/(2T)) + 1/\pi$. Putting (i), (ii), **2F**, and **2G** together, we see that $\mathrm{tr}(Q(c,P)W^*)/T^2$ is lower bounded by

$$
\begin{aligned}
&\sum_{i,j=0}^{d} q_{ij}\mathbb{E}\bar{x}_i\bar{x}_j - \sum_{\substack{i\neq j,\\ i,j\in P}} |q_{ij}| e^{-\frac{2C^2T^2}{9}} \left\{ \frac{2\left\|u_i\right\|_2^2 + 2\left\|u_j\right\|_2^2}{\sqrt{2\pi}CT} + 4g(C,T)\left\|u_i\right\|_2\left\|u_j\right\|_2 \right\} \\
&- 2\sum_{j\in P} |q_{0j}| \Bigg\{ e^{-\frac{2C^2T^2}{9}} \left\{ \frac{2\left\|u_j\right\|_2^2}{\sqrt{2\pi}CT} + 2g(C,T)\left\|u_j\right\|_2 \right\} \\
&+ e^{-\frac{T^2}{2}} \left\{ \frac{2}{\sqrt{2\pi}}\frac{1}{T} + \left\|u_j\right\|_2 \cdot \left[\frac{2}{\sqrt{2\pi}} \left(T + \frac{2}{T} \right) + \frac{1}{\pi} \right] \right\} \Bigg\} \\
&- \frac{1}{T^2} \sum_{\substack{(i,j)\notin P\times P,\\ 0\le i,j\le d, (i,j)\neq(0,0)}} |q_{ij}| \cdot |u_i^\top u_j| - \sum_{i\in P} q_{ii} \left| \frac{\sqrt{3}}{\sqrt{2}T} - \frac{1}{T^2} u_i^\top u_i \right| \\
\geq\ & \sum_{i,j=0}^{d} q_{ij}\mathbb{E}\bar{x}_i\bar{x}_j - B\cdot e^{-\frac{2C^2T^2}{9}} \left\{ \frac{4\sigma\min\{d,\sigma/C^2\}}{\sqrt{2\pi}CT} + \frac{4\sigma}{\sqrt{2\pi}CT} + 4g(C,T)(\sigma+1) \right\} \\
&- B\cdot e^{-\frac{T^2}{2}} \left\{ \frac{2|P|}{\sqrt{2\pi}T} + \frac{\sigma}{C} \cdot \left[\frac{2}{\sqrt{2\pi}} \left(T + \frac{2}{T} \right) + \frac{1}{\pi} \right] \right\} - \frac{1}{T^2} B(3\sigma + \sigma^2) - \frac{\sqrt{3}B}{\sqrt{2}T}|P|,
\end{aligned}
$$

where we use Hölder's inequality with $(\infty, 1)$-norm, together with the following facts:

- $\sum_{i=1}^{d} \left\|u_i\right\|_2^2 = \mathrm{tr}(W_x^*) \le \sigma$, $\left\|u_0\right\|_2^2 = W_{11}^* = 1$,
- $\sum_{i\in P} \left\|u_i\right\|_2 \le \sum_{i\in P} \left\|u_i\right\|_2^2 / C \le \sigma/C$,
- $\sum_{0\le i,j\le d} \left\|u_i\right\|_2 \left\|u_j\right\|_2 \le \sum_{i=0}^{d} \left\|u_i\right\|_2^2 \le \sigma + 1$,
- $\sum_{0\le i,j\le d,(i,j)\neq(0,0)} |u_i^\top u_j| \le 2\sum_{i=1}^{d} |u_0^\top u_i| + \sum_{i,j=1}^{d} |u_i^\top u_j| \le 2\sigma + \sigma^2$, where we use Proposition 2 and $1_d^\top |W_x^*| 1_d \le \sigma^2$ in the last inequality.

Lastly, by **2E** we obtain our desired inequality.

To conclude the proof, we remains to show that $\bar{x}$ is feasible to SBQP with high probability. we only need to show that $\left\|\bar{x}\right\|_0 \le \sigma$ holds with probability at least $1 - \exp\{-c\sigma\}$ for some (absolute) constant $c > 0$. Since $\mathbb{E}\left\|\bar{x}\right\|_0 \le \sum_{i=1}^{d} p_i \le 2/3 \cdot \sigma$, by multiplicative Chernoff bound equipped with an upper bound for the expectation (see, e.g., Theorem 4.4 and the remark after Corollary 4.6 in [36]), we have

$$
\mathbb{P}\left(\left\|\bar{x}\right\|_0 \ge \sigma \right) \le \mathbb{P}\left(\left\|\bar{x}\right\|_0 \ge \left(1 + \frac{1}{2} \right) \cdot \frac{2}{3}\sigma \right) \le e^{-\frac{\sigma}{18}}.
$$

□

B Proofs of Theorems 3 and 4

In this section, we first prove Lemma 1, and then we use it to prove prove Theorem 3 in Appendix B.1 and Theorem 4 in Appendix B.2. To show Lemma 1, we need two lemmas.

Lemma 3 ([25], Section 5)**.** *Let $D = \mathrm{diag}(d_i)$ be a diagonal matrix of order n, and let $C = D + auu^\top$ with $a < 0$ and u being an n-vector. Denote the eigenvalues of C by $\lambda_1, \lambda_2, \cdots, \lambda_n$ and assume $\lambda_i \le \lambda_{i+1}$, $d_i \le d_{i+1}$. We have $d_1 + a\left\|u\right\|_2^2 \le \lambda_1 \le d_1$, and $d_{i-1} \le \lambda_i \le d_i$ for $i \ge 2$.*

Lemma 4 ([7], Appendix A.5.5)**.** *Let P be a symmetric matrix written as a 2×2 block matrix $P = \begin{pmatrix} P_{11} & P_{12} \\ P_{12}^\top & P_{22} \end{pmatrix}$. The following are equivalent:*

(1) $P \succeq 0$.

(2) $P_{11} \succeq 0$, $(I - P_{11}P_{11}^\dagger)P_{12} = O$, *and* $P_{22} \succeq P_{12}^\top P_{11}^\dagger P_{12}$.

We are now ready to prove Lemma 1.

Proof of Lemma 1. We divide the proof into three steps. In Step A, we show $p^* \geq 0_d$, $\lambda_2(H_{S,S}) \geq \delta$, and $H_{S,S} \succeq 0$. In Step B, we show that if in addition, **1A** - **1D** hold, then W^* is optimal to SILS'-SDP. In Step C, we show that if furthermore $\lambda_2(H) > 0$ holds, then W^* is the unique optimal solution to SILS'-SDP.

Step A. By (4), (5), and (6), we can show that $\min_{i \in S} p_i^* = -\lambda_{\min}\big(\frac{1}{n}(M^\top M)_{S,S}\big) + \delta - \mu_2^* \geq 0$. Combining the fact $p_{S^c}^* = 0_{d-\sigma}$, we conclude that $p^* \geq 0_d$.

Next, we show $\lambda_2(H_{S,S}) \geq \delta$. To see this, (3) gives

$$H_{S,S} = \frac{1}{n}(M^\top M)_{S,S} + \mu_3^* x_S^* (x_S^*)^\top + \mathrm{diag}(p_S^* + \mu_2^* 1_\sigma) - \frac{1}{Y_{11}^*} y_S^* (y_S^*)^\top.$$

By (4), x_S^* is an eigenvector of $H_{S,S}$ corresponding to the zero eigenvalue. Therefore, to show $\lambda_2(H_{S,S}) \geq \delta$, it is sufficient to show that for any unit vector $a \in \mathrm{Span}(\{x_S^*\})^\perp$, we have $a^\top H_{S,S} a \geq \delta$. We obtain

$$a^\top H_{S,S} a = a^\top \Big(\frac{1}{n}(M^\top M)_{S,S} + \mathrm{diag}(p_S^* + \mu_2^* 1_\sigma) - \frac{1}{Y_{11}^*} y_S^* (y_S^*)^\top \Big) a.$$

We then define the following two auxiliary matrices:

$$R := \frac{1}{n}(M^\top M)_{S,S} + \mu_2^* I_\sigma + \mathrm{diag}(p_S^*) - \frac{1}{Y_{11}^*} y_S^* (y_S^*)^\top, \qquad P := R + \frac{1}{Y_{11}^*} y_S^* (y_S^*)^\top.$$

To prove $a^\top H_{S,S} a \geq \delta$, it is sufficient to show $\lambda_{\min}(P) \geq \delta$. Indeed, by Lemma 3, we see $\lambda_2(R) \geq \lambda_{\min}(P) \geq \delta$. From (4), x_S^* is an eigenvector of R corresponding to eigenvalue $-\sigma\mu_3^* \leq 0$, so it is an eigenvector corresponding to the smallest eigenvalue of R, which then implies $a^\top H_{S,S} a = a^\top R a \geq \delta$. We now check $\lambda_{\min}(P) \geq \delta$. Recall again $\min_{i \in S} p_i^* = -\lambda_{\min}\big(\frac{1}{n}(M^\top M)_{S,S}\big) + \delta - \mu_2^*$. We have

$$P = \frac{1}{n}(M^\top M)_{S,S} + \mu_2^* I_\sigma + \mathrm{diag}(p_S^*) \succeq \Big(\lambda_{\min}\big(\frac{1}{n}(M^\top M)_{S,S}\big) + \mu_2^* + \min_{i \in S} p_i^* \Big) I_\sigma = \delta I_\sigma.$$

This concludes the proof that $\lambda_{\min}(P) \geq \delta$, and therefore $\lambda_2(H_{S,S}) \geq \delta$.

Finally, $H_{S,S} \succeq 0$ follows easily if one observes that $\lambda_{\min}(H_{S,S}) = 0$. Indeed, direct calculation and (4) gives $H_{S,S} x_S^* = 0_\sigma$, which gives our desired property.

Step B. In this part, we show W^* is optimal by checking (KKT-1) - (KKT-3). We first show that $H \succeq 0$. From Lemma 4, it suffices to show the following three facts: (i) $H_{S,S} \succeq 0$, (ii) $(I_\sigma - H_{S,S} H_{S,S}^\dagger) H_{S,S^c} = O_{\sigma \times (d-\sigma)}$, and (iii) $H_{S^c,S^c} \succeq H_{S^c,S} H_{S,S}^\dagger H_{S^c,S}^\top$. Note that (i) holds by part (a) and (iii) holds by **1A**, so it remains to show (ii). By **1B**, we see that $H_{S^c,S} x_S^* = 0_{d-\sigma}$. Since $\lambda_2(H_{S,S}) \geq \delta > 0$, we conclude that that (ii) indeed holds.

We define $Y^* := \begin{pmatrix} Y_{11}^* & (y^*)^\top \\ y^* & Y_x^* \end{pmatrix}$ and $\mu_1^* := Y_{11}^* - 1/n \cdot b^\top b$. Observe that $Y^* \succeq 0$ again by Lemma 4, due to the facts $H \succeq 0$ and $Y_{11}^* > 0$.

Step C. Finally, we show that W^* is the unique optimal solution if we additionally assume $\lambda_2(H) > 0$. First, note that $\lambda_2(H) > 0$ implies $\lambda_2(Y^*) > 0$ due to the fact that $Y^* = \begin{pmatrix} 1 & \\ \frac{1}{Y_{11}^*}y^* & I_d \end{pmatrix} \begin{pmatrix} Y_{11}^* & \\ & Y_x^* - \frac{1}{Y_{11}^*}y^*(y^*)^\top \end{pmatrix} \begin{pmatrix} 1 & \\ \frac{1}{Y_{11}^*}y^* & I_d \end{pmatrix}^\top$.

We define the Lagrangian function $\mathcal{L} : \mathbb{R}^{(1+d)\times(1+d)} \to \mathbb{R}$ as follows: $\mathcal{L}(W) := \frac{1}{n}\operatorname{tr}(A^\top AW) - \operatorname{tr}(Y^*W) + \mu_1^*(W_{11} - 1) + \mu_2^*(\operatorname{tr}(W_x) - \sigma) + \mu_3^*(1_d^\top |W_x| 1_d - \sigma^2) + \operatorname{tr}(\operatorname{diag}(p^*)(W_x - I))$. Then, for any optimal solution W_0 to SILS'-SDP, we show $W^* = W_0$. It is clear

$$\frac{1}{n}\operatorname{tr}(A^\top AW_0) \geq \mathcal{L}(W_0) \geq \mathcal{L}(W^*) = \frac{1}{n}\operatorname{tr}(A^\top AW^*),$$

where the second inequality is due to (KKT-1), which states that O_{1+d} lies in the sub-differential of $\mathcal{L}(W^*)$. By the optimality of W_0, it is clear that, from the second term $-\operatorname{tr}(Y^*W_0)$ to the last term $\operatorname{tr}(\operatorname{diag}(p^*)((W_0)_x - I))$ in $\mathcal{L}(W_0)$, are all zero, as they are always non-positive. In particular, $0 = \operatorname{tr}(Y^*W^*) = \operatorname{tr}(Y^*W_0)$ holds. This implies that W_0 must be a scaling of W^* since $\lambda_2(Y^*) > 0$. Again by optimality of W_0, we see $W_0 = W^*$. □

In the remainder of the section we prove Theorems 3 and 4. We start with a useful lemma, which introduces the Schur complement of a positive semidefinite matrix. This result follows from Lemma 4.

Lemma 5. *For a positive semidefinite matrix $H \in \mathbb{R}^{d\times d}$, and a set of indices $S \subseteq [d]$. Denote $P_1 := \begin{pmatrix} I_\sigma & \\ H_{S,S^c}^\top H_{S,S}^\dagger & I_{d-\sigma} \end{pmatrix}$, we have*

$$\begin{pmatrix} H_{S,S} & H_{S,S^c} \\ H_{S,S^c}^\top & H_{S^c,S^c} \end{pmatrix} = P_1 \cdot \begin{pmatrix} H_{S,S} & \\ & H_{S^c,S^c} - H_{S,S^c}^\top H_{S,S}^\dagger H_{S,S^c} \end{pmatrix} \cdot P_1^\top.$$

B.1 Proof of Theorem 3

In this proof we intend to use Lemma 1, thus we check that all assumptions in Lemma 1 are satisfied. In particular, we take $(Y_x^*)_{S,S}$ as per (3), p_S^* as per (4), and $p_{S^c}^* = 0_{d-\sigma}$, as in the statement of Lemma 1. Note that since Y_x^* is not completely determined, we also need to define its missing parts, i.e., its (S^c, S) and (S^c, S^c) blocks. For brevity, we denote $H^0 := I_\sigma - x_S^*(x_S^*)^\top/\sigma$ and $P := (M^\top M)_{S,S^c}/n - y_S^*(y_{S^c}^*)^\top/Y_{11}^*$. We take

$$(Y_x^*)_{S^c,S} := \frac{1}{n}(M^\top M)_{S^c,S} - \left[\frac{1}{n\sigma}(M^\top M)_{S^c,S}x_S^* - \frac{1}{Y_{11}^*\sigma}y_{S^c}^*(y_S^*)^\top x_S^*\right](x_S^*)^\top, \tag{10}$$

$$(Y_x^*)_{S^c,S^c} := \Theta_1 + \nu I_{d-\sigma} + \frac{1}{Y_{11}^*}y_{S^c}^*(y_{S^c}^*)^\top + \frac{1}{\delta}P^\top H^0 P, \tag{11}$$

where we set $\nu := \mu_3^* - \|\Theta_2\|_\infty \geq 0$. As in Lemma 1 we define $H := Y_x^* - y^*(y^*)^\top/Y_{11}^*$.

Next, we show that **1A** - **1D** are implied due to our choice of p^* and Y_x^*, and conditions **A1** - **A2**. This will show that $W^* := \begin{pmatrix} 1 \\ x^* \end{pmatrix}\begin{pmatrix} 1 \\ x^* \end{pmatrix}^\top$ is optimal to SILS'-SDP. After that, we show that **A2** automatically implies $\lambda_2(H) > 0$, which additionally guarantees the uniqueness of W^*, and we conclude that SILS-SDP recovers x^*.

We now check that **1A** holds. By direct calculation,

$$H_{S^c,S^c} \succeq \frac{1}{\delta}P^\top H^0 P \succeq H_{S^c,S}H_{S,S}^\dagger H_{S^c,S}^\top,$$

where the last inequality is due to the facts that $P = H^0 H_{S^c,S}^\top$, $(H^0)^2 = H^0$, and $H_{S,S} \succeq \delta\big(I_\sigma - x_S^*(x_S^*)^\top/\sigma\big)$. The last fact is due to $\lambda_2(H_{S,S}) \geq \delta$ and $H_{S,S}x_S^* = 0_\sigma$.

Next, we prove that **1B** is satisfied. From (10), we obtain

$$H_{S^c,S}x_S^* = [\frac{1}{n}(M^\top M)_{S^c,S} - \frac{1}{Y_{11}^*}y_{S^c}^*(y_S^*)^\top][I_\sigma - \frac{1}{\sigma}x_S^*(x_S^*)^\top]x_S^* = 0_{d-\sigma}.$$

1C is automatically true due to (10) and **A1**, and **1D** is directly implied by (11), triangle inequality, and **A2**.

Finally, we will show **A2** implies $\lambda_2(H) > 0$. Lemma 5 shows that $\lambda_2(H) > 0$ is equivalent to $H_{S^c,S^c} - H_{S,S^c}^\top H_{S,S}^\dagger H_{S,S^c} \succ 0$, due to the facts $\lambda_{\min}(H_{S,S}) = 0$ and $\lambda_2(H_{S,S}) \geq \delta > 0$. Finally, we observe that $H_{S^c,S^c} - H_{S,S^c}^\top H_{S,S}^\dagger H_{S,S^c} \succeq H_{S^c,S^c} - H_{S,S^c}^\top (1/\delta)\cdot\big(I_\sigma - x_S^*(x_S^*)^\top/\sigma\big)H_{S,S^c} = \Theta_1 + \nu I_{d-\sigma} \succ 0$ as desired. □

B.2 Proof of Theorem 4

In this proof we use Lemma 1, thus we check that all assumptions in Lemma 1 are satisfied. We fix $(Y_x^*)_{S,S}$ as per (3), p_S^* as per (4), and $p_{S^c}^* = 0_{d-\sigma}$. Note that we still need to define the missing parts of Y_x^*, namely, its (S^c, S) and (S^c, S^c) blocks. We take

$$(Y_x^*)_{S^c,S} := -\frac{1}{\sigma}y_{S^c}^*(x_S^*)^\top, \tag{12}$$

$$(Y_x^*)_{S^c,S^c} := \nu I_{d-\sigma} + \frac{1}{Y_{11}^*}y_{S^c}^*(y_{S^c}^*)^\top + H_{S^c,S}H_{S,S}^\dagger H_{S^c,S}^\top. \tag{13}$$

With a little abuse of notation, we denote by $\nu > 0$ the slack in the inequality introduced in **B2**. As in Lemma 1 we define $H := Y_x^* - y^*(y^*)^\top/Y_{11}^*$.

Next, we check **1A** - **1D**, and $\lambda_2(H) > 0$. Similarly to the proof of Theorem 3, we show that **1A** - **1D** are implied by our choice of p^* and Y_x^*, and conditions **B1** - **B2**. This will show that $W := \begin{pmatrix}1\\ x^*\end{pmatrix}\begin{pmatrix}1\\ x^*\end{pmatrix}^\top$ is optimal to SILS'-SDP. After that, we show that **B2** implies $\lambda_2(H) > 0$, which additionally guarantees the uniqueness of W^*, and we conclude that SILS'-SDP recovers x^*.

It is clear that **1A** holds by (13). Next, we prove that **1B** is satisfied. From (12) and the definition $Y_{11}^* = -(y_S^*)^\top x_S^*$, we obtain that $H_{S^c,S}x_S^* = 0_{d-\sigma}$. **1C** is true due to (12) and **B1**. For **1D**, by (13), and triangle inequality, $\big\|(\frac{1}{n}M^\top M - Y_x^*)_{S^c,S^c} + \mu_2^* I_{d-\sigma}\big\|_\infty$ is then upper bounded by

$$\begin{aligned}
&\left\|\frac{1}{n}(M^\top M)_{S^c,S^c} + \mu_2^* I_{d-\sigma}\right\|_\infty + \left\|H_{S^c,S}H_{S,S}^\dagger H_{S^c,S}^\top\right\|_\infty + \left\|\frac{1}{Y_{11}^*}y_{S^c}^*(y_{S^c}^*)^\top\right\|_\infty + \nu\\
\leq &\left\|\frac{1}{n}(M^\top M)_{S^c,S^c} + \mu_2^* I_{d-\sigma}\right\|_\infty + \left\|\frac{1}{Y_{11}^*}y_{S^c}^*(y_{S^c}^*)^\top\right\|_\infty + \nu + \frac{1}{\delta}\left\|\frac{1}{\sigma}x_S^* + \frac{1}{Y_{11}^*}y_S^*\right\|_2^2 \|y_{S^c}^*\|_\infty^2\\
= &\left\|\frac{1}{n}(M^\top M)_{S^c,S^c} + \mu_2^* I_{d-\sigma}\right\|_\infty + \left\|\frac{1}{Y_{11}^*}y_{S^c}^*(y_{S^c}^*)^\top\right\|_\infty + \nu + \frac{1-\cos^2(\theta)}{\delta\sigma\cos^2(\theta)}\|y_{S^c}^*\|_\infty^2 \overset{\mathbf{B2}}{=} \mu_3^*,
\end{aligned}$$

where we used the fact that $\left\|H_{S,S}^\dagger\right\|_2 \leq 1/\delta$ in the first inequality, and the fact that

$$\left\|-\frac{1}{\sigma}x_S^* - \frac{1}{Y_{11}^*}y_S^*\right\|_2^2 = \frac{1}{\sigma} + \frac{2(x_S^*)^\top y_S^*}{Y_{11}^*\sigma} + \left\|\frac{1}{Y_{11}^*}y_S^*\right\|_2^2 = -\frac{1}{\sigma} + \left\|\frac{1}{Y_{11}^*}y_S^*\right\|_2^2 = \frac{1-\cos^2(\theta)}{\sigma\cos^2(\theta)}$$

in the penultimate equality.

Finally, we show that **B2** implies $\lambda_2(H) > 0$. From Lemma 5, it suffices to show $\lambda_{\min}(H_{S^c,S^c} - H_{S,S^c}^\top H_{S,S}^\dagger H_{S,S^c})$ is positive. By definition of H_{S^c,S^c}, we obtain that $H_{S^c,S^c} - H_{S,S^c}^\top H_{S,S}^\dagger H_{S,S^c} = \nu I_{d-\sigma} \succ 0$. □

C Proof of Theorem 6

In this section, we prove Theorem 6. We first give a technical lemma, which gives high-probability upper bounds for metrics between some random variables and their means. This lemma is due to known results in probability and statistics.

Lemma 6. *Suppose that M consists of centered row vectors $m_i \overset{i.i.d.}{\sim} \mathcal{SG}(L^2)$ for some $L > 0$ and $i \in [n]$, and denote the covariance matrix of m_i by Σ. Assume the noise vector ϵ is a centered sub-Gaussian random vector independent of M, with $\epsilon_i \overset{i.i.d.}{\sim} \mathcal{SG}(\varrho^2)$ for $i \in [n]$. Then, the following statements hold:*

- ***6A.*** *Suppose $\sigma/n \to 0$. Then, there exists an absolute constant $c_1 > 0$ such that $\left\| \frac{1}{n}(M^\top M)_{S,S} - \Sigma_{S,S} \right\|_2 \le c_1 L\sqrt{\sigma/n}$ holds w.h.p. as $(n, \sigma) \to \infty$;*
- ***6B.*** *Suppose $\log(d)/n \to 0$ and let $F := \frac{1}{n}M^\top M - \Sigma$. Then, there exists an absolute constant B such that $\|F\|_\infty \le BL^2\sqrt{\log(d)/n}$ holds w.h.p. as $(n, d) \to \infty$;*
- ***6C.*** *Suppose $\log(d)/n \to 0$ and let $F := \frac{1}{n}M^\top M - \Sigma$. Let $x^* \in \{0, \pm 1\}^d$, define $S := \mathrm{Supp}(x^*)$, and assume $|S| = \sigma$. Then, there exists an absolute constant B_1 such that $\|Fx^*\|_\infty = \|F_{S,S}x^*_S\|_\infty \le B_1 L^2\sqrt{\sigma \log(d)/n}$ holds w.h.p. as $(n, d) \to \infty$;*
- ***6D.*** *Suppose $\log(d)/n \to 0$ and let $F := \frac{1}{n}M^\top M - \Sigma$. Let $z^* \in \mathbb{R}^d$. Then, there exists an absolute constant B_2 such that $\left\|Fz^* + \frac{1}{n}M^\top \epsilon\right\|_\infty < B_2 L\sqrt{(\varrho^2 + L^2\|z^*\|_2^2)\log(d)/n}$ holds w.h.p. as $(n, d) \to \infty$.*

Proof. **6A** follows from Proposition 2.1 in [51]. **6B**, **6C**, and **6D** follow from Berstein inequality (see, e.g., Theorem 2.8.1 in [52]), and an argument of union bound. □

Then, we prove Theorem 6 by utilizing Theorem 4. In order to maintain the conditions in Theorem 4, we use the concentration bounds introduced in Lemma 6, substitute the random variables in Theorem 4 by their means, and then add or subtract upper bounds of metrics between the random variables and their means, as proposed in Lemma 6. □

D Proof of Theorem 7

In this section, we prove Theorem 7. Let $x^*_i = \begin{cases} \mathrm{sign}(z^*_i),\ i \le \sigma, \\ 0,\ \text{otherwise}, \end{cases}$ and $S := [\sigma]$. In this proof, we employ Theorem 6 to prove that SILS recovers x^* when n is large enough, by checking all the assumptions therein. We observe that $L = 1$ when $\Sigma = I_d$. We also have $\hat{y}^*_S = -z^*_S$ and $\hat{Y}^*_{11}/\sigma = (x^*_S)^\top I_d z^*_S/\sigma \ge g + 1 = \Omega(1)$. Throughout the proof, we take $n \ge C(\|z^*\|_2^2 + \sigma^2 + \varrho^2)\log(d)$ for some absolute constant $C > 0$. For brevity, we say that n is sufficiently large if we take a sufficiently large C. For **D1**, we first show that $l_n = \mathcal{O}(1/\sqrt{\sigma})$ if n is large enough. By Section 4.1, we see that for some $\eta \in [0, 1]$,

$$l_n \le \frac{\sigma}{|[\hat{y}^*_S + \eta(\hat{y}^*_S - y^*_S)]^\top x^*_S|} \cdot \frac{\|\hat{y}^*_S + \eta(\hat{y}^*_S - y^*_S)\|_2}{\left\|\hat{y}^*_S + \eta(\hat{y}^*_S - y^*_S)\right\|_1}.$$

For ease of notation, we denote $\lambda_n := B_2\sqrt{(\varrho^2 + \|z^*\|_2^2)\log(d)/n}$. From **6D**,

$$|[\hat{y}^*_S + \eta(\hat{y}^*_S - y^*_S)]^\top x^*_S| \ge |(\hat{y}^*_S)^\top z^*_S| - |(\hat{y}^*_S - y^*_S)^\top x^*_S| \ge 2(1+g)\sigma - \sigma\|\hat{y}^*_S - y^*_S\|_\infty \ge \sigma\left(2(1+g) - \lambda_n\right).$$

Using **6D** again, we have

$$\frac{\|\hat{y}^*_S + \eta(\hat{y}^*_S - y^*_S)\|_2}{\left\|\hat{y}^*_S + \eta(\hat{y}^*_S - y^*_S)\right\|_1} \le \frac{\|\hat{y}^*_S\|_2 + \|\hat{y}^*_S - y^*_S\|_2}{\left\|\hat{y}^*_S\right\|_1 - \left\|\hat{y}^*_S - y^*_S\right\|_1} \le \frac{\sqrt{2u\sigma} + \sqrt{\sigma}\|\hat{y}^*_S - y^*_S\|_\infty}{2(1+g)\sigma - \sigma\left\|\hat{y}^*_S - y^*_S\right\|_\infty} \le \frac{1}{\sqrt{\sigma}} \cdot \frac{\sqrt{2u} + \lambda_n}{2(1+g) - \lambda_n}.$$

Combining the above two inequalities, we see $l_n = \mathcal{O}(1/\sqrt{\sigma})$ when n is sufficiently large.

For **D2**, we set $\delta = g/2$. We obtain that

$$\hat{\mu}_3^* \geq \frac{1}{\sigma}\left(1 - \frac{g}{2} + g - \lambda_n - B_1\sqrt{\frac{\sigma\log(d)}{n}} - c_1\sqrt{\frac{\sigma}{n}}\right) > \frac{1}{\sigma}\left(1 + \frac{g}{4}\right)$$

if n is sufficiently large. Since we have $|\hat{y}_{S^c}^*| \leq 1_{d-\sigma}$ and $\Sigma_{S,S^c} = O_{\sigma\times(d-\sigma)}$, we see that **D2** is true for a sufficiently large n.

To show **D3**, we set $\hat{\mu}_2^* = -1$ and we see that $\hat{\mu}_2^* = -1 \leq -1 + \delta - c_1\sqrt{\sigma/n}$ holds for large n. Therefore, $\Sigma_{S^c,S^c} + \hat{\mu}_2^* I_{d-\sigma} = O_{(d-\sigma)\times(d-\sigma)}$. Moreover, (8) implies $f_n(\hat{y}_S^*)^2 = \frac{\|\hat{y}_S^*\|_2^2}{[(\hat{y}_S^*)^\top x_S^*]^2} - \frac{1}{\sigma} \leq \frac{g^2}{2\sigma(g+1)}$, and hence

$$\gamma_n = (f_n(\hat{y}_S^*) + l_n\lambda_n)^2\left(\|\hat{y}_{S^c}\|_\infty + \lambda_n\right)^2 \cdot \frac{1}{\delta} \leq \frac{g^2}{2\sigma(g+1)} \cdot 1 \cdot \frac{2}{g} = \frac{g}{\sigma(g+1)},$$

for sufficiently large n, where we absorb the diminishing term brought by $l_n\lambda_n$ into the term $(\|\hat{y}_{S^c}\|_\infty + \lambda_n)^2$, as $\|\hat{y}_{S^c}\|_\infty = \|z_{S^c}^*\|_\infty < 1$. It remains to check $B\sqrt{\frac{\log(d)}{n}} + \frac{(\|\hat{y}_{S^c}\|_\infty + \lambda_n)^2}{\hat{Y}_{11}^* - \sigma\lambda_n} + \frac{g}{\sigma(g+1)} \leq \hat{\mu}_3^*$. By absorbing the diminishing term brought by λ_n into $\|\hat{y}_{S^c}\|_\infty < 1$, we obtain that

$$B\sqrt{\frac{\log(d)}{n}} + \frac{1}{\sigma(g+1)} + \frac{g}{\sigma(g+1)} = B\sqrt{\frac{\log(d)}{n}} + \frac{1}{\sigma} \cdot 1 < \frac{1}{\sigma}\left(1 + \frac{g}{4}\right) < \mu_3^*$$

for a sufficiently large n. Finally, we observe that $\|z^*\|_2^2 \leq d + \sigma u^2$, which concludes the proof. □

E Proof of Theorem 9

Before proving Theorem 9, we need some detailed analysis of our covariance matrix Σ and some useful probabilistic inequalities. We will use them to evaluate norms of some matrices, which are used for the construction of the decomposition $\Theta = \Theta_1 + \Theta_2$ in Theorem 8.

Throughout the section, we use the same definitions as in the statement of Theorem 8, i.e., $S := \mathrm{Supp}(z^*)$, $y^* := -M^\top b/n$, $Y_{11}^* := -(y_S^*)^\top z_S^*$, and $\mu_3^* = 1/\sigma \cdot \{\lambda_{\min}\big((M^\top M/n)_{S,S}\big) - \delta + \min_{i\in S}[M^\top\epsilon]_i/(nx_i^*)\}$. Furthermore, we use the notation introduced in Model 2 and we introduce some additional notation that is specific for it. Let $y_i', y_i'' \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0_d, I_d)$. We observe that m_i has the same distribution as another random vector $\Sigma_1^{\frac{1}{2}} y_i' + \Sigma_2^{\frac{1}{2}} y_i''$. For the ease of notation, we write $M_1^\top := \Sigma_1^{\frac{1}{2}}(y_1', \cdots, y_n')$ and $M_2^\top := \Sigma_2^{\frac{1}{2}}(y_1'', \cdots, y_n'')$. Hence we assume $M = M_1 + M_2$. Observe that $\Sigma_2^{\frac{1}{2}} = \begin{pmatrix} O_\sigma & \\ & \sqrt{c''} I_{d-\sigma} \end{pmatrix}$, so M_2 is an $n \times d$ matrix with the first σ columns being zero.

In Lemma 7 below, we show that $\Sigma_1^{\frac{1}{2}}$ has a simple structure. The proof can be easily done via an analysis of singular value decomposition of Σ, and we omit it here.

Lemma 7. *In Model 2, we have* $\Sigma_1^{\frac{1}{2}} = \begin{pmatrix} A_{11} & a1_\sigma 1_{d-\sigma}^\top \\ a1_{d-\sigma}1_\sigma^\top & b1_{d-\sigma}1_{d-\sigma}^\top \end{pmatrix}$ *for some matrix* $A_{11} \in \mathbb{R}^{\sigma\times\sigma}$ *and* $a, b \in \mathbb{R}$.

By Lemma 7, we observe that $(M_1^\top M_1)_{S^c,S}$, $(M_1^\top M_2)_{S^c,S^c}$, and $(M_1^\top M_1)_{S^c,S^c}$ are rank-one matrices. In fact, there exist vectors $u \in \mathbb{R}^\sigma$, $v \in \mathbb{R}^{d-\sigma}$, and a scalar c_1 such that $(M_1^\top M_1)_{S^c,S}/n = 1_{d-\sigma}u^\top$, $(M_1^\top M_2)_{S^c,S^c}/n = 1_{d-\sigma}v^\top$, and $(M_1^\top M_1)_{S^c,S^c}/n = c_1 1_{d-\sigma}1_{d-\sigma}^\top$. In the next lemma, we provide some probabilistic upper bounds. The proofs can be obtained by applying Bernstein inequalities to different sub-exponential variables introduced in the lemma below.

Lemma 8. *Consider Model 2 and suppose* $\log(d)/n \to 0$ *and* $(n,d,\sigma) \to \infty$. *Let* u, v, c_1 *be as defined above. Then, the following properties hold with probability at least* $1 - \mathcal{O}(1/d)$*:*

8A. $\exists$ *constant* $C_1 = C_1(c,c'')$ *such that* $\big\|(M_2^\top M_1/n)_{S^c,S}\big\|_\infty \le C_1\sqrt{\log(d)/n}$;

8B. $\exists$ *constant* $C_2 = C_2(c,c'')$ *such that* $\big\|(M_2^\top M_1)_{[d],S}z_S^*/n\big\|_\infty \le C_2\sqrt{\sigma\log(d)/n}$;

8C. $\exists$ *constant* $C_3 = C_3(c,c',c'')$ *such that* $\big\|(M^\top\epsilon/n)_{S^c}\big\|_\infty \le C_3\sqrt{\varrho^2\sigma\log(d)/n}$;

8D. $\exists$ *constant* $C_4 = C_4(c)$ *such that* $\big\|(M^\top\epsilon/n)_{S}\big\|_\infty \le C_4\sqrt{\varrho^2\log(d)/n}$;

8E. $\exists$ *constant* $C_5 = C_5(c',c'')$ *such that* $\|v\|_\infty \le C_5\sqrt{\sigma\log(d)/n}$

8F. $\exists$ *constant* $C_6 = C_6(c,c'')$ *such that* $\|(M_2^\top M_1/n)_{S^c,S}\|_{2\to\infty} \le C_6(\sqrt{\log(d)}+\sqrt{\sigma})/\sqrt{n}$;

8G. $\exists$ *constant* $C_7 = C_7(c,c')$ *such that* $\|u - 1_\sigma\|_\infty \le C_7\sqrt{\sigma\log(d)/n}$;

8H. $\exists$ *constant* $C_8 = C_8(c')$ *such that* $|c_1 - c'\sigma| \le C_8\sigma\sqrt{\log(d)/n}$.

In the following, we define some matrices that will be used in the proof of Theorem 9 for the construction of Θ_1 and Θ_2 in Theorem 8. Recall that $H^0 = I_\sigma - z_S^*(z_S^*)^\top/\sigma$. For simplicity, we denote $B := [I_\sigma + z_S^*(y_S^*)^\top/Y_{11}^*](1/\delta)H^0[I_\sigma + y_S^*(z_S^*)^\top/Y_{11}^*] + z_S^*(z_S^*)^\top/Y_{11}^*$, and we define

$$
\begin{aligned}
\Theta_2^A &:= -\frac{1}{Y_{11}^*}(\frac{1}{n}M^\top\epsilon)_{S^c}(\frac{1}{n}M^\top\epsilon)_{S^c}^\top,\\
\Theta_1^B &:= \Big(\sqrt{\tilde c}1_{d-\sigma} + \frac{u^\top x_S^*}{Y_{11}^*\sqrt{\tilde c}}(\frac{1}{n}M^\top\epsilon)_{S^c}\Big)\Big(\sqrt{\tilde c}1_{d-\sigma} + \frac{u^\top x_S^*}{Y_{11}^*\sqrt{\tilde c}}(\frac{1}{n}M^\top\epsilon)_{S^c}\Big)^\top,\\
\Theta_2^B &:= -\frac{1}{Y_{11}^*}(\frac{1}{n}M^\top\epsilon)_{S^c}\big(\frac{1}{n}(M_2^\top M_1)_{S^c,S}z_S^*\big)^\top - \frac{1}{Y_{11}^*}\big(\frac{1}{n}(M_2^\top M_1)_{S^c,S}z_S^*\big)(\frac{1}{n}M^\top\epsilon)_{S^c}^\top\\
&\quad - \frac{(u^\top z_S^*)^2}{(Y_{11}^*)^2\tilde c}(\frac{1}{n}M^\top\epsilon)_{S^c}(\frac{1}{n}M^\top\epsilon)_{S^c}^\top,\\
\Theta_2^C &:= -\frac{1}{\delta n^2 Y_{11}^*}\Bigg[(M^\top M)_{S^c,S}\big(I_\sigma + \frac{z_S^*(y_S^*)^\top}{Y_{11}^*}\big)H^0y_S^*(M^\top\epsilon)_{S^c}^\top\\
&\quad - (M^\top\epsilon)_{S^c}(y_S^*)^\top H^0\big(I_\sigma + \frac{y_S^*(z_S^*)^\top}{Y_{11}^*}\big)(M^\top M)_{S,S^c}\Bigg],\\
\Theta_2^D &:= \frac{1}{\delta(nY_{11}^*)^2}(M^\top\epsilon)_{S^c}(y_S^*)^\top H^0y_S^*(M^\top\epsilon)_{S^c}^\top,\\
\Theta_1^E &:= \hat c 1_{d-\sigma}1_{d-\sigma} - \frac{1}{n}(M_1^\top M_1)_{S^c,S}B\frac{1}{n}(M_1^\top M_1)_{S,S^c}\\
&\quad + \Big(\sqrt{\bar c}1_{d-\sigma} - \frac{1}{\sqrt{\bar c}}\frac{1}{n}(M_2^\top M_1)_{S^c,S}Bu\Big)\Big(\sqrt{\bar c}1_{d-\sigma} - \frac{1}{\sqrt{\bar c}}\frac{1}{n}(M_2^\top M_1)_{S^c,S}Bu\Big)^\top,\\
\Theta_2^E &:= -\frac{1}{\bar c}\frac{1}{n}(M_2^\top M_1)_{S^c,S}Au\Big(\frac{1}{n}(M_2^\top M_1)_{S^c,S}Bu\Big)^\top - \frac{1}{n}(M_2^\top M_1)_{S^c,S}B\frac{1}{n}(M_2^\top M_1)_{S,S^c},\\
\Theta_1^F &:= \Big(\frac{1}{n}M_1^\top M_1\Big)_{S^c,S^c} - (\bar c + \hat c + \tilde c + \check c)1_{d-\sigma}1_{d-\sigma}^\top + \check c\Big(1_{d-\sigma} + \frac{1}{\check c}v\Big)\Big(1_{d-\sigma} + \frac{1}{\check c}v\Big)^\top,
\end{aligned}
$$

$$\Theta_2^F := -\frac{1}{\check{c}} vv^\top + \Big(\frac{1}{n} M_2^\top M_2\Big)_{S^c,S^c} + \mu_2^* I_{d-\sigma},$$

for some proper positive constants $\bar{c}$, $\hat{c}$, $\tilde{c}$ and $\check{c}$ such that Θ_1^B, Θ_1^E and Θ_1^F are positive semidefinite matrices. The high-level idea in the proof of Theorem 9 is to take $\Theta_1 = \Theta_1^B + \Theta_1^E + \Theta_1^F$ and $\Theta_2 = \Theta_2^A + \Theta_2^B + \Theta_2^C + \Theta_2^D + \Theta_2^E + \Theta_2^F$, and to directly check that such Θ_1 and Θ_2 add up to Θ in Theorem 8. Before proving Theorem 9, we need two lemmas: Lemma 9 gives some useful results that will be used repeatedly in the proofs of Lemma 10 and Theorem 9, and Lemma 10 gives upper bounds on the infinity norms of the matrices defined above that contribute to Θ_2.

Lemma 9. *There exists a constant $C = C(c, c', c'') > 0$ such that when $n \geq C\varrho^2\sigma^2 \log(d)$, the following properties hold w.h.p. as $(n, \sigma, d) \to \infty$:*

9A. $Y_{11}^* \geq \sigma/2$;

9B. $\|-y_S^* - z_S^*\|_2 \leq 1/2$;

9C. $\big\|u^\top (I_\sigma + y_S^*(z_S^*)^\top / Y_{11}^*)\big\|_2 \leq 6\sqrt{\sigma}$;

9D. $\big\|H^0 y_S^*\big\|_2 \leq 1/2$.

Proof. For brevity, in this proof, we say that n is sufficiently large if we take a sufficiently large C.

For **9A**, observe $Y_{11}^* = -(z_S^*)^\top y_S^* = (z_S^*)^\top (M^\top M/n)_{S,S} z_S^* - (z_S^*)^\top (M^\top \epsilon/n)_S$, and hence from **6A** and **8D**, $Y_{11}^* \geq \sigma - c_1\sigma\sqrt{\sigma/n} - (z_S^*)^\top (M^\top\epsilon/n)_S \geq \sigma(1 - c_1\sqrt{\sigma/n} - C_4\sqrt{\varrho^2\log(d)/n}) \geq \sigma/2$, for sufficiently large n.

For **9B**, observe that

$$\|-y_S^* - z_S^*\|_2 = \Big\|((M^\top M)_{S,S}/n - I_\sigma) z_S^* + (M^\top\epsilon/n)_S\Big\|_2 \leq \Big\|(M^\top M)_{S,S}/n - I_\sigma\Big\|_2 \cdot \|z_S^*\|_2 + \Big\|(M^\top\epsilon/n)_S\Big\|_2 .$$

From **6A** and **8D**, we see that this quantity is upper bounded by $c_1\sqrt{\sigma^2/n} + \sqrt{\sigma} C_4 \sqrt{\varrho^2\log(d)/n}$, which is less than $1/2$, for sufficiently large n.

For **9C**, we have that

$$\Big\|u^\top (I_\sigma + y_S^*(z_S^*)^\top/Y_{11}^*)\Big\|_2 \leq \Big\|I_\sigma + z_S^*(y_S^*)^\top/Y_{11}^*\Big\|_2 \|u\|_2 \leq (1 + \|z_S^*\|_2 \|y_S^*\|_2 / Y_{11}^*)\|u\|_2 ,$$

and hence by **9A**, **8G**, and **9B**, we obtain that it is upper bounded by $[1 + 2\cdot(1/2 + 1/(2\sqrt{\sigma}))]\cdot \sqrt{\sigma}(1 + C_7\sqrt{\sigma\log(d)/n}) \leq 6\sqrt{\sigma}$ for sufficiently large n.

Finally, for **9D**, we observe that $H^0 z_S^* = (I_\sigma - z_S^*(z_S^*)^\top/\sigma) z_S^* = 0_\sigma$, thus $\big\|H^0 y_S^*\big\|_2 = \big\|H^0(y_S^* - z_S^*)\big\|_2 \leq \big\|H^0\big\|_2 \|y_S^* - z_S^*\|_2 \leq 1/2$ by the fact that $\big\|H^0\big\|_2 = 1$ and **9B**. □

Lemma 10. *There exists a constant $C = C(c, c', c'') > 0$ such that when $n \geq C\varrho^2\sigma^2\log(d)$, the following properties hold w.h.p. as $(n, \sigma, d) \to \infty$:*

10A. $\big\|\Theta_2^A\big\|_\infty = \mathcal{O}\Big(\varrho^2\log(d)/n\Big)$;

10B. $\big\|\Theta_2^B\big\|_\infty = \mathcal{O}\Big(\varrho\sigma\log(d)/n + \varrho^2\sigma\log(d)/(\check{c}n)\Big)$;

10C. $\big\|\Theta_2^C\big\|_\infty = \mathcal{O}\Big(\sqrt{\varrho^2\log(d)/n}/\delta + \sqrt{\sigma}\varrho\log(d)/(\delta n)\Big)$;

10D. $\big\|\Theta_2^D\big\|_\infty = \mathcal{O}\Big(\varrho^2\log(d)/(n\sigma\delta)\Big)$;

10E. $\big\|\Theta_2^E\big\|_\infty = \mathcal{O}\Big((\sqrt{\sigma\log(d)} + \sigma)^2/(\bar{c}\delta^2 n) + \sigma\log(d)/(\delta n)\Big)$.

Proof. For brevity, in this proof, we say that n is sufficiently large if we take a sufficiently large C. In the proof, we will repeatedly use the fact that for a rank-one matrix $P = ab^\top$, $\|P\|_\infty = \|a\|_\infty \|b\|_\infty$.

(**10A**). The statement simply follows from **9A** and **8C**.

(**10B**). observe that, among the three terms in Θ_2^B, the first term is the transpose of the second term, so it is sufficient to upper bound the infinity norm of the first term, since the same bound holds for the second. From **8B**, **8C**, and **9A**, we have that

$$\left\| 1/Y_{11}^* \cdot (M^\top \epsilon/n)_{S^c} ((M_2^\top M_1/n)_{S^c,S} z_S^*)^\top \right\|_\infty$$

is upper bounded by $2C_2C_3\varrho\sigma\log(d)/n$. Then, from **8C**, **8G**, and **9A**, we conclude to the fact that $\left\|(u^\top z_S^*)^2/[(Y_{11}^*)^2\tilde{c}] \cdot (M^\top\epsilon/n)_{S^c}(M^\top\epsilon/n)_{S^c}^\top\right\|_\infty \le 4C_3^2\varrho^2\sigma\log(d)/(\tilde{c}n)$.

(**10C**). Note that the first term in the definition of Θ_2^C is the transpose of the second term, thus it is sufficient to upper bound the infinity norm of the first term. We write

$$\begin{aligned}
&(M^\top M/n)_{S^c,S}(I_\sigma + z_S^*(y_S^*)^\top/Y_{11}^*)H^0 y_S^*(M^\top\epsilon/n)_{S^c}^\top \\
&= 1_{d-\sigma} u^\top (I_\sigma + z_S^*(y_S^*)^\top/Y_{11}^*)H^0 y_S^*(M^\top\epsilon/n)_{S^c}^\top + (M_2^\top M_1)_{S^c,S}(I_\sigma + z_S^*(y_S^*)^\top/Y_{11}^*)H^0 y_S^*(M^\top\epsilon/n)_{S^c}^\top \\
&:= P_1 + P_2,
\end{aligned}$$

since $(M_1^\top M_2)_{S^c,S} = (M_2^\top M_2)_{S^c,S} = O_{(d-\sigma)\times\sigma}$. It is clear that

$$\|P_1\|_\infty = |u^\top (I_\sigma + y_S^*(z_S^*)^\top/Y_{11}^*)H^0 y_S^*| \left\|(M^\top\epsilon/n)_{S^c}\right\|_\infty,$$

by **9C**, **9D**, and **8C**, we see $\|P_1\|_\infty \le 3C_3\sqrt{\varrho^2\sigma^2\log(d)/n}$.

Next, $\|P_2\|_\infty = \left\|(M_2^\top M_1)_{S^c,S}(I_\sigma + z_S^*(y_S^*)^\top/Y_{11}^*)H^0 y_S^*\right\|_\infty \left\|(M^\top\epsilon/n)_{S^c}\right\|_\infty$, thus from **8F** and **8C**, we obtain that

$$\|P_2\|_\infty \le C_3C_6\sqrt{\varrho^2\sigma\log(d)/n}(\sqrt{\log(d)} + \sqrt{\sigma})/\sqrt{n} \cdot \left\|(I_\sigma + z_S^*(y_S^*)^\top/Y_{11}^*)H^0 y_S^*\right\|_2.$$

By **9D**, **9B**, and **9A**, we obtain that

$$\left\|(I_\sigma + z_S^*(y_S^*)^\top/Y_{11}^*)H^0 y_S^*\right\|_2 \le \left\|H^0 y_S^*\right\|_2 + \|x_S^*\|_2 \|y_S^*\|_2 \left\|H^0 y_S^*\right\|_2/Y_{11}^* \le 2.$$

Hence, we see $\|P_2\|_\infty \le 2C_3C_6\sqrt{\varrho^2\sigma\log(d)/n}(\sqrt{\log(d)} + \sqrt{\sigma})/\sqrt{n}$.

Finally, from **9A**, we obtain $\left\|\Theta_2^C\right\|_\infty = \mathcal{O}\Big(\sqrt{\varrho^2\log(d)/n}/\delta + \sqrt{\sigma}\varrho\log(d)/(\delta n)\Big)$.

(**10D**). Since $H^0 z_S^* = 0_\sigma$, we obtain that $(y_S^*)^\top H^0 y_S^* = (y_S^* - z_S^*)^\top H^0 (y_S^* - z_S^*)$. By **9B** and $\left\|H^0\right\|_2 = 1$, we see $(y_S^*)^\top H^0 y_S^* \le 1/4$. We are done by combining the above conclusion, **8C**, and **9A**.

(**10E**). We start by estimating the infinity norm of the first term in the definition of Θ_2^E. To do so, we first provide an upper bound on $\|B\|_2$. Write

$$B = [H^0/\delta + z_S^*(z_S^*)^\top/Y_{11}^* + (y_S^*)^\top H^0 y_S^* z_S^*(z_S^*)^\top/(Y_{11}^*)^2] + [z_S^*(y_S^*)^\top H^0 + H^0 y_S^*(x_S^*)^\top]/(\delta Y_{11}^*) := B_1 + B_2,$$

and we will upper bound $\|B_1\|_2$ and $\|B_2\|_2$. For B_1, recall that $H^0 = I_\sigma - z_S^*(z_S^*)^\top/\sigma$, thus $B_1 = (1/\delta)I_\sigma + [1/Y_{11}^* + (y_S^*)^\top H^0 y_S^*/(Y_{11}^*)^2 - 1/(\sigma\delta)]x_S^*(x_S^*)^\top$. From $H^0 = (H^0)^2$, **9A**, and **9D**, we see $(y_S^*)^\top H^0 y_S^*/(Y_{11}^*)^2 \le 1/\sigma^2$, and thus $\|B_1\|_2 \le 1/\delta + 2 + 1/\sigma + 1/\delta \le 3 + 2/\delta$. For B_2, we only need to upper bound $z_S^*(y_S^*)^\top H^0/(\delta Y_{11}^*)$, since the other term is symmetric. From **9A** and **9D**, $\left\|z_S^*(y_S^*)^\top H^0/(\delta Y_{11}^*)\right\|_2 \le 2\|x_S^*\|_2 \left\|H^0 y_S^*\right\|_2/(\delta\sigma) \le 1/\delta$. Thus, $\|B\|_2 \le 3 + 3/\delta$. Combining this and **8F**, **8G**, we obtain $\left\|(M_2^\top M_1/n)_{S^c,S} Bu\right\|_\infty = \mathcal{O}\Big((\sqrt{\sigma\log(d)} + \sigma)/(\delta\sqrt{n})\Big)$.

For the second term in the definition of Θ_2^E, we write $B := H^0/\delta + B_3$, and we give upper bounds on the infinity norms of $(M_2^\top M_1/n)_{S^c,S} H^0 (M_2^\top M_1/n)_{S,S^c}$ and $(M_2^\top M_1/n)_{S^c,S} B_3 (M_2^\top M_1/n)_{S,S^c}$. We know that the diagonal entries of H^0 are $1 - 1/\sigma$, and the off-diagonal entries have an absolute value of $1/\sigma$, thus, along with **8A** we see $\left\|(M_2^\top M_1/n)_{S^c,S} H^0 (M_2^\top M_1/n)_{S,S^c}\right\|_\infty \le \left\|(M_2^\top M_1/n)_{S^c,S}\right\|_\infty^2 [\sigma \cdot (1-1/\sigma) + \sigma(\sigma-1)\cdot 1/\sigma] = \mathcal{O}(\sigma \log(d)/n)$. Next, by **9A** and **9D**, each entry in B_3 is upper bounded by $\mathcal{O}(1/\sigma + 1/(\delta\sigma))$. Together with **8A**, we obtain that

$$\left\|(M_2^\top M_1/n)_{S^c,S} B_3 (M_2^\top M_1/n)_{S,S^c}\right\|_\infty \le \sigma^2 \left\|(M_2^\top M_1/n)_{S^c,S}\right\|_\infty^2 \cdot \mathcal{O}(1/\sigma + 1/(\delta\sigma)) = \mathcal{O}(\sigma \log(d)/(\delta n)).$$

Using the triangle inequality, the second term in Θ_2^E has infinity norm upper bounded by $\mathcal{O}(\sigma \log(d)/(\delta n))$. □

We are now ready to prove Theorem 9 using Theorem 8.

Proof of Theorem 9. We use Theorem 8 to prove this proposition. In the proof, We take $n \ge C\varrho^2\sigma^2 \log(d)$ for some constant $C = C(c, c', c'') > 0$. For brevity, we say n is sufficiently large if we take a sufficiently large C. Recall that we take $\mu_3^* = 1/\sigma \cdot \{\lambda_{\min}\big((M^\top M/n)_{S,S}\big) - \delta + \min_{i\in S}[M^\top \epsilon]_i/(n x_i^*)\}$. We now check the remaining conditions required in Theorem 8. Note that the assumption $Y_{11}^* > 0$ is automatically true by **9A**. Next, we take $\delta := 1 + \max\{\lambda_{\min}\big((M^\top M/n)_{S,S}\big) - 1 - c'', 0\} \ge 1$. μ_3^* is indeed nonnegative due to **8D** and **6A** with $L^2 = c$, because $\mu_3^* \ge (c-1)/(2\sigma) > 0$ (if $\delta = 1$) or $\mu_3^* \ge c''/(2\sigma) > 0$ (if $\delta > 1$) for sufficiently large n. From **8C**, **E1** is true for sufficiently large n. Next, we focus on **E2**. We first take $\mu_2^* := -c''$, and now we show that it is a valid choice by checking $\mu_2^* \in (-\infty, -\lambda_{\min}\big((M^\top M/n)_{S,S}\big) + \delta]$. Note that if $\delta = 1$, we have $\lambda_{\min}\big((M^\top M/n)_{S,S}\big) - 1 - c'' \le 0$, and therefore $-\lambda_{\min}\big((M^\top M/n)_{S,S}\big) + \delta \ge -c''$; on the contrary, if $\delta > 1$, we have $-\lambda_{\min}\big((M^\top M/n)_{S,S}\big) + \delta = -c''$. This implies that we can take $\mu_2^* = -c''$ in both cases.

Next, we construct Θ_1 and Θ_2 as required in **E2**. We take $\Theta_1 = \Theta_1^B + \Theta_1^E + \Theta_1^F$ and $\Theta_2 = \Theta_2^A + \Theta_2^B + \Theta_2^C + \Theta_2^D + \Theta_2^E + \Theta_2^F$. It still remains to (a) give valid choices for the constants $\bar{c}$, $\hat{c}$, $\tilde{c}$, and $\check{c}$ in Θ_1^B, Θ_1^E and Θ_1^F such that these three matrices are positive semidefinite; (b) show that $\Theta = \Theta_1 + \Theta_2$; and (c) prove that $\|\Theta_2\|_\infty < \mu_3^*$.

For (a), it suffices to show that we can take $\bar{c}$, $\hat{c}$, $\tilde{c}$, and $\check{c}$ in a way such that the first two terms in the definition of Θ_1^E sum up to a positive semidefinite matrix, and the first two terms in the definition of Θ_1^F sum up to a positive semidefinite matrix. From **8H**, we obtain $(M_1^\top M_1/n)_{S^c,S^c} \succeq (c'\sigma - C_8\sigma\sqrt{\log(d)/n}) 1_{d-\sigma} 1_{d-\sigma}^\top$, so it suffices to give some choices of these constants such that $\hat{c} 1_{d-\sigma} 1_{d-\sigma} - 1_{d-\sigma} u^\top B u 1_{d-\sigma}^\top \succeq 0$ and $c'\sigma - C_8\sigma\sqrt{\log(d)/n} - (\bar{c} + \hat{c} + \tilde{c} + \check{c}) \ge 0$. We first take $\hat{c} = u^\top B u$, where the definition of B can be found after the proof of Lemma 8. We then validate the choice by showing $u^\top B u = \sigma + \mathcal{O}(\sqrt{\sigma})$. Indeed, since $c' > 1$, this shows $u^\top B u < c'\sigma$ for some moderately large σ, making it possible to attain a nonnegative $c'\sigma - Bc'\sigma\sqrt{\log(d)/n} - (\bar{c} + \hat{c} + \tilde{c} + \check{c})$, for sufficiently large n. Observe that

$$u^\top B u = u^\top \Big(H^0 + \frac{1}{Y_{11}^*} x_S^* (x_S^*)^\top\Big) u + 2\frac{u^\top x_S^*}{Y_{11}^*} (y_S^*)^\top H^0 u + \frac{(u^\top x_S^*)^2}{(Y_{11}^*)^2} (y_S^*)^\top H^0 y_S^*.$$

By **6A** and **8D**, we have $Y_{11}^* \ge \sigma\Big(1 - c_1\sqrt{\sigma/n} - C_4\sqrt{\varrho^2 \log(d)/n}\Big)$. Thus, when n is large enough, we obtain $1/Y_{11}^* \cdot \le 1/\sigma(1 + 2c_1\sqrt{\sigma/n} + 2C_4\sqrt{\varrho^2\log(d)/n})$. Recall that $H^0 = I_\sigma - x_S^*(x_S^*)^\top/\sigma$. We then see that by **8G**, $u^\top\Big(H^0 + \frac{1}{Y_{11}^*} x_S^*(x_S^*)^\top\Big)u$ is upper bounded by $\sigma\Big(1 + C_7\sqrt{\frac{\sigma\log(d)}{n}}\Big)^2 + \sigma \cdot \Big(2c_1\sqrt{\frac{\sigma}{n}} + 2C_4\sqrt{c\frac{\varrho^2\log(d)}{n}}\Big) = \sigma + \mathcal{O}(\sqrt{\sigma})$, when n is sufficiently large. Implied by **8G**, we see that $\Big|\frac{u^\top x_S^*}{Y_{11}^*}(y_S^*)^\top H^0 u\Big|$ is upper bounded by $\sigma\Big(1 + C_7\sqrt{\frac{\sigma\log(d)}{n}}\Big) \cdot \frac{1}{\sigma}\Big(1 + 2c_1\sqrt{\frac{\sigma}{n}} + 2C_4\sqrt{\frac{\varrho^2\log(d)}{n}}\Big) \cdot$

$\|H^0 y_S^*\|_2 \|u\|_2$. The term can be further upper bounded by $\mathcal{O}(\sqrt{\sigma})$ by **9D** and **8G**. For last term $\frac{(u^\top x_S^*)^2}{(Y_{11}^*)^2}(y_S^*)^\top H^0 y_S^*$ in $u^\top B u$, it can be upper bounded by $\Big(1+C_7\sqrt{\frac{\sigma\log(d)}{n}}\Big)^2 \cdot \Big(1+2c_1\sqrt{\frac{\sigma}{n}}+ 2C_4\sqrt{\frac{\varrho^2\log(d)}{n}}\Big)^2 \cdot \frac{1}{4} \le \mathcal{O}(1)$ as a result of **8G**,**9B**, and **9D**. Finally, we take $0 < \tilde{c}, \check{c} \ll 1$ small enough, and $\bar{c} = c'\sigma - \hat{c} - C_8\sigma\sqrt{\log(d)/n} - \tilde{c} - \check{c}$, to enforce $c'\sigma - C_8\sigma\sqrt{\log(d)/n} - (\bar{c}+\hat{c}+\tilde{c}+\check{c}) \ge 0$. We can verify that $\bar{c} > 0$ if n and σ are sufficiently large and $\tilde{c}, \check{c}$ are chosen to be sufficiently small.

Checking the validity of (b) is straightforward by direct calculation. For (c), we first show $\|\Theta_2^F\|_\infty = \mathcal{O}\Big(\sigma\log(d)/n + \sqrt{\log(d)/n}\Big)$, which is indeed true because $\|\Theta_2^F\|_\infty \le \|vv^\top/\check{c}\|_\infty + \|(M_2^\top M_2/n)_{S^c,S^c} - c''I_{d-\sigma}\|_\infty = \mathcal{O}\Big(\sigma\log(d)/n + \sqrt{\log(d)/n}\Big)$, where the last equality is due to **8E** and **6B** with $L^2 = c''$. Combing this fact and Lemma 10, we obtain that

$$
\begin{aligned}
&\|\Theta_2\|_\infty \le \|\Theta_2^A\|_\infty + \|\Theta_2^B\|_\infty + \|\Theta_2^C\|_\infty + \|\Theta_2^D\|_\infty + \|\Theta_2^E\|_\infty + \|\Theta_2^F\|_\infty \\
&\le \mathcal{O}\Big(\frac{\varrho^2\log(d)}{n}\Big) + \mathcal{O}\Big(\frac{\varrho\sigma\log(d)}{n} + \frac{\varrho^2\sigma\log(d)}{n}\Big) + \mathcal{O}\Big(\sqrt{\frac{\varrho^2\log(d)}{n}} + \frac{\sqrt{\sigma}\varrho\log(d)}{n}\Big) \\
&\quad + \mathcal{O}\Big(\frac{\varrho^2\log(d)}{n\sigma}\Big) + \mathcal{O}\Big(\frac{(\sqrt{\sigma\log(d)}+\sigma)^2}{n} + \frac{\sigma\log(d)}{n}\Big) + \mathcal{O}\Big(\frac{\sigma\log(d)}{n} + \sqrt{\frac{\log(d)}{n}}\Big) \\
&\le \frac{1}{4\sigma}\min\{c-1, c''\} < \frac{1}{2\sigma}\min\{c-1, c''\} \le \mu_3^*.
\end{aligned}
$$

w.h.p. when $n \ge C\varrho^2\sigma^2\log(d)$, for some large constant $C = C(c, c', c'') > 0$. □

F Proof of Theorem 10

In this section we prove Theorem 10 using Theorem 6. The proof idea is very similar to the one in the proof of Theorem 7, and hence we will skip some of the detailed calculation. Note that $L = \mathcal{O}(1)$ when $\Sigma = I_d$. We first see $\hat{y}_S^* = -z_S^*$ and then $\hat{Y}_{11}^*/\sigma = (z_S^*)^\top I_d z_S^*/\sigma = 1$. Throughout the proof, we take $n \ge C\big(\sigma^2+\varrho^2\big)\log(d)$ for some absolute constant C. For brevity, we say that n is sufficiently large if we take a sufficiently large C.

For condition **D1**, we can show that $l_n = \mathcal{O}(1/\sqrt{\sigma})$ if n is large enough, in a similar way to the proof of Theorem 7. For **D2**, we set $\delta = 1/2$, and obtain that $\hat{\mu}_3^* > 1(1\sigma)$ when n is sufficiently large. Since $\hat{y}_{S^c}^* = 0_{d-\sigma}$ and $\Sigma_{S,S^c} = O_{\sigma\times(d-\sigma)}$, **D2** indeed holds for n sufficiently large. To show **D3**, we first set $\hat{\mu}_2^* = -1$. Observe that $\hat{\mu}_2^* \le -1/2 - c_1\sqrt{\sigma/n}$ indeed holds when n is sufficiently large. Therefore, we see $\Sigma_{S^c,S^c} + \hat{\mu}_2^* I_{d-\sigma} = O_{d-\sigma}$. Furthermore, since $x_S^* = z_S^*$, we have $\cos(\hat{\theta}) = 1$, and it remains to check whether $B\sqrt{\frac{\log(d)}{n}} + \frac{\lambda_n^2}{\hat{Y}_{11}^* - \sigma\lambda_n} + 2\ell_n^2\lambda_n^4 \le \hat{\mu}_3^*$, which is indeed true for a sufficiently large n. □

G Extended empirical results

In this section, we provide detailed and additional empirical results deferred in Section 5. In Appendix G.1, we provide detailed discussion for the tests conducted in Section 5.1. In Appendix G.2, we provide detailed discussion on performance of SILS'-SDP under other statistical models, and provide empirical results regarding empirical probability of recovery. In Appendix G.3, we provide numerical results for applying Algorithm 1 to problems with non-convex objective functions.

G.1 Detailed algorithmic results

In this section, we provide comprehensive computational results summarized in Section 5.1. We start by introducing the use of the Conditional Gradient Augmented Lagrangian framework

(CGAL) in Section G.1.1, provide detailed description of datasets involved in Appendix G.1.2, and then provide extended computational results in Appendix G.1.3.

G.1.1 Introduction to CGAL.

In this section, we introduce CGAL, an iterative method designed for approximating solutions to the optimization problem:

$$x^* := \arg\min \ f(x) \quad \text{s.t. } x \in \mathcal{X},\ Cx \in \mathcal{K},$$

where f is a convex and L-smooth function, C is a matrix, $\mathcal{X}$ a convex compact set, and $\mathcal{K}$ a convex set. By the m-th iteration, CGAL yields a vector x_m such that $|f(x_m) - f(x^*)| \leq \mathcal{O}(m^{-1/2})$ and $\text{dist}(Cx_m, \mathcal{K}) \leq \mathcal{O}(m^{-1/2})$. It is noteworthy that the most computationally intensive step in each iteration involves finding the minimum eigenvector of a $(1+d) \times (1+d)$ matrix, which can be efficiently executed using the Lanczos method.

In the context of SDP(c, P), we define the function $f(W) = \text{tr}(Q(c, P)W)$, set $\mathcal{X}$ as the compact convex set $\{W \in \mathbb{R}^{(1+d)\times(1+d)} : \text{tr}(W) \leq \sigma + 1, W \succeq 0\}$, set $C = I_{1+d}$, and specify $\mathcal{K}$ as the convex set $\{W \in \mathbb{R}^{(1+d)\times(1+d)} : W_{11} = 1, 1_d^\top |W_x| 1_d \leq \sigma^2, \text{diag}(W_x) \leq 1_d\}$. We initiate CGAL with parameter $\lambda_0 = 0.01$ and start from the solution $\begin{pmatrix} 1 & \\ & O_d \end{pmatrix}$. From a practical viewpoint, we oftentimes limit CGAL to 20 iterations, observing that this suffices for Algorithm 1 to produce high-quality solutions, even though the SDP objective value may not be precisely accurate due to the limited iterations. Further iterations of CGAL enhance the SDP objective value but do not significantly improve the performance of Algorithm 1, leading us to report only the results from 20 iterations.

G.1.2 Description of datasets.

Below are the detailed specifications for the datasets involved in Section 5.1:

1. (Synthetic dataset) We take $\sigma = 20$, d ranging from 1000 to 10000, $c = 0.6$, and $n = \lceil 2\sigma \log d/(1-c)^2 \rceil$ in Example 1 in [6], i.e., the inputs satisfies (LM), the rows of M are drawn from i.i.d. $\mathcal{N}(0_d, \Sigma)$, with $\Sigma_{ij} = c^{|i-j|}$, $z^* \in \{0, \pm 1\}^d$ by assigning a random subset of cardinality σ to be nonzero, and $\epsilon \sim \mathcal{N}(0_d, I_d)$.

2. (Synthetic dataset) We take $\sigma = 20$, d ranging from 1000 to 10000, $n = \lceil 4\sigma \log d \rceil$, and $\varrho = 1$ in Model 2.

3. (Synthetic dataset) We take $\sigma = 20$, d ranging from 1000 to 10000, $n = \lceil 4\sigma \log d \rceil$, and $\varrho = 1$ in Model 3.

4. (Diabete dataset in [19]) In this dataset, a matrix M can be obtained with $n = 442$ and $d = 10$, where we drop the column for the response vector y. We randomly generate $z^* \in \{0, \pm 1\}^d$ by assigning a random subset of cardinality σ to be nonzero, and then we obtain a semisynthetic input (M, b) by assigning $b = Mz^* + \epsilon$, with $\epsilon \sim \mathcal{N}(0_d, I_d)$. This dataset can be downloaded from https://www4.stat.ncsu.edu/%7Eboos/var.select/.

5. (Leukemia dataset in [15]) In this dataset, a matrix M can be obtained with $n = 72$ and $d = 3571$. We randomly generate $z^* \in \{0, \pm 1\}^d$ by assigning a random subset of cardinality σ to be nonzero, and then we obtain a semisynthetic input (M, b) by assigning $b = Mz^* + \epsilon$, with $\epsilon \sim \mathcal{N}(0_d, I_d)$. This dataset can be downloaded from https://stat.ethz.ch/Manuscripts/dettling/leukemia.rda.

6. (Prostate dataset in [15]) In this dataset, a matrix M can be obtained with $n = 102$ and $d = 6033$. We randomly generate $z^* \in \{0, \pm 1\}^d$ by assigning a random subset

of cardinality σ to be nonzero, and then we obtain a semisynthetic input (M, b) by assigning $b = Mz^* + \epsilon$, with $\epsilon \sim \mathcal{N}(0_d, I_d)$. This dataset can be downloaded from https://stat.ethz.ch/Manuscripts/dettling/prostate.rda.

The random seed is set to 42 for all tests to ensure reproducibility. The goal of employing these diverse datasets is to assess the scalability and robustness of CGAL + Algorithm 1 across different data complexities and sizes. It is important to note that in the first and second synthetic datasets, we consciously avoid biasing towards SILS-SDP by using excessively large sample sizes n. Instead, we cap n at $\lceil 4\sigma \log d \rceil$, a threshold that is insufficient for recovery, as it requires $\Omega(\sigma^2 \log d)$ samples according to Theorems 9 and 10. We set the sample size to $\mathcal{O}(\sigma \log d)$ to explore the performance of Algorithm 1 under suboptimal conditions, with a deliberate focus on scenarios featuring an inadequate number of samples.

G.1.3 Extended algorithmic results.

In this section, we run CGAL for $m = 20$ iterations (specific deviations will be noted), and provide detailed numerical results that are used to obtain Table 1 in Section 5.1. We summarize the results in Tables 2 to 7. Recall that obj_{z^*} stands for the objective value for the feasible solution z^* we used to generate the dataset, as discussed in Section 5.1. In the column "CGAL + Algorithm 1", we report the average objective value among the 1001 we obtained from Algorithm 1 in the sub-column "mean val"; we report the objective value of the best feasible solution obtained among the 1001 solutions in the sub-column "best val". We also report the run time of these algorithms, and MIP gaps obtained by Gurobi.

From Tables 2 to 7, CGAL + Algorithm 1 outperforms both SBQP and MIO in 32 out of 41 instances (78%), with a distinct advantage in 18 instances (44%). In the nine instances where CGAL + Algorithm 1 is less effective, seven showed improvement when CGAL iterations were increased; these adjustments are reflected in dual-row entries for CGAL + Algorithm 1 in the tables. For the remaining two instances $\sigma = 10, 20$ in leukemia dataset shown in Table 6, we found that increasing m to 100 would not enhance the performance of CGAL + Algorithm 1, leading us to also include the best solutions to SBQP and MIO within similar operational timeframes. We can see that within the same time constraint, CGAL + Algorithm 1 can indeed produce solution of similar quality compared to these two other algorithms.

The tables also illustrate that CGAL + Algorithm 1 excels at handling large-scale instances, with d up to 10000, a scale challenging for SBQP and MIO. Although MIO performs well in Table 2, which is owing to the fact that Gurobi finds a high quality heuristic solution efficiently in this model (about 30s for $d = 1000$ and about 10 minutes for $d = 10000$), it is generally outperformed by CGAL + Algorithm 1 for large instances with $d \geq 5000$. Notably, in Table 7, MIO yields only positive objective values, contradicting the expectation of non-positive objectives for $\text{SDP}(P, c)$. Similarly, in Table 4, SBQP fails to surpass a trivial zero solution for $d \geq 3000$, resulting in significant objective discrepancies compared to the other methods.

Finally, we comment on the fact that the performance of CGAL + Algorithm 1 does not match well with the column obj_{z^*} in Tables 6 and 7 for large σ. This may stem from insufficient data to obtain recovery or even to obtain an approximate recovery with $\text{SDP}(P, c)$, as n is significantly lower than d. Consequently, the optimal solution W^* does not approximate the ideal $\begin{pmatrix} 1 \\ z^* \end{pmatrix} \begin{pmatrix} 1 \\ z^* \end{pmatrix}^\top$, limiting the effectiveness of Algorithm 1 and the associated greedy algorithm in approximating z^*.

G.2 Detailed statistical results

In this section, we provide extended statistical results on SILS'-SDP. We first report the performance of feature extraction problem in Appendix G.2.1, by discussing statistical performance of

		SBQP			MIO			CGAL + Algorithm 1		
	obj_{z^*}	obj	time	mipgap	obj	time	mipgap	mean val	best val	time
$d = 1000$ $n = 1727$	-19.25	**-19.25**	9	0	**-19.06**	7	0	-0.75	**-19.25**	13
$d = 2000$ $n = 1901$	-19.01	**-19.01**	46	0	**-19.01**	45	0	-0.28	**-19.01**	40
$d = 3000$ $n = 2002$	-18.38	**-18.38**	175	0	**-18.38**	117	0	-0.32 -0.71	-14.87 **-18.38**	79 335
$d = 4000$ $n = 2074$	-20.23	**-20.23**	339	0	**-20.23**	316	0	-0.16	**-20.23**	132
$d = 5000$ $n = 2130$	-19.99	-0.08	1003	$\geq 10^6$	-19.33	1001	$\geq 10^4$	-0.24	**-19.99**	197
$d = 6000$ $n = 2175$	-19.27	0	1001	-	**-19.27**	1003	$\geq 10^4$	-0.07 -0.32	-17.34 **-19.27**	266 559
$d = 7000$ $n = 2214$	-19.95	0	1002	-	**-19.95**	1001	$\geq 10^4$	-0.08	**-19.95**	367
$d = 8000$ $n = 2247$	-18.91	0	1002	-	**-18.93**	1001	$\geq 10^4$	0.12	**-18.91**	452
$d = 9000$ $n = 2277$	-19.96	0	1005	-	**-19.96**	1006	$\geq 10^4$	0.09	**-19.96**	602
$d = 10000$ $n = 2303$	-21.32	0	1006	-	**-21.32**	1002	$\geq 10^4$	-0.11	**-21.32**	790

Table 2: Performance under Example 1 in [6] via CGAL ($\sigma = 20$). Time limit is set to 1000s. Two rows for each of the instances $d = 3000$ and $d = 6000$ are reported, detailing the performance of CGAL + Algorithm 1 for varying iteration of CGAL (m). Specifically, for $d = 3000$, m is set to 20 in the first row and 100 in the second. For $d = 6000$, m is set to 20 and 40 in the first and second rows, respectively.

SILS'-SDP under Model 1. Then, we report the performance of integer sparse recovery problem. We provide numerical results on recovery under Model 2 in Appendix G.2.2, and then report statistical performance under Model 3 in Appendix G.2.3.

G.2.1 Statistical performance under Model 1.

In this section, we report numerical performance of SILS'-SDP in the feature extraction problem under Model 1, as studied in Section 4.1.1. We assume that the entries of M in Model 1 are i.i.d. standard Gaussian, and $\epsilon \sim \mathcal{N}(0_d, \varrho^2 I_d)$. For simplicity, we take the first σ entries of z^* to be ± 2, and the remaining entries to be ± 1, and note that (8) indeed holds in this case.

In Figure 2, we first validate Theorem 7 numerically, by plotting the *empirical probability of recovery*, i.e., the percentage of times SILS'-SDP solves SILS' over 100 instances, for each $n = \lceil cd\log(d) \rceil$, with control parameter c ranging from 0.25 to 4. Note that, here, $d\log(d)$ is the dominating term in the lower bound on n in Theorem 7. As discussed after Theorem 7, for small values of n, the recovered sparse integer vector is not necessarily the vector x^* in the proof of Theorem 7. In Figure 3, we then plot the *empirical probability of recovery of* x^*, i.e., the percentage of times SILS'-SDP recovers x^* over 100 instances. The instances considered in Figure 3 are identical to those considered in Figure 2. As shown in Figures 2 and 3, both the empirical probability of recovery and the empirical probability of recovery of x^* go to 1 as c grows larger. However, the empirical probability of recovery is much closer to one also for small values of c.

		SBQP			MIO			CGAL + Algorithm 1		
	obj_{z^*}	obj	time	mipgap	obj	time	mipgap	mean val	best val	time
$d = 1000$ $n = 553$	-107.53	**-107.53**	24	0	**-107.54**	22	0	164.94	**-107.53**	11
$d = 2000$ $n = 609$	-1484.19	-1447.82	1000	$\geq 10^4$	-1145.19	1000	$\geq 10^4$	1.83 2.42	0 **-1462.65**	35 74
$d = 3000$ $n = 641$	-358.96	-318.26	1000	$\geq 10^5$	-337.34	1000	$\geq 10^5$	19.60	**-358.96**	71
$d = 4000$ $n = 664$	-741.44	-704.32	1001	$\geq 10^5$	-713.93	1001	$\geq 10^6$	3.25 86.4	0 **-720.72**	112 257
$d = 5000$ $n = 682$	-499.82	-468.24	1001	$\geq 10^7$	-479.61	1001	$\geq 10^6$	-0.91 96.92	-309.09 **-488.73**	185 395
$d = 6000$ $n = 696$	-360.82	-325.53	1002	$\geq 10^6$	-341.06	1001	$\geq 10^6$	21.15	**-360.81**	273
$d = 7000$ $n = 709$	-99.73	-55.46	1002	$\geq 10^7$	-79.32	1001	$\geq 10^7$	129.5	**-99.73**	362
$d = 8000$ $n = 719$	-106.54	-67.03	1002	$\geq 10^7$	-84.66	1002	$\geq 10^7$	137.58	**-106.54**	461
$d = 9000$ $n = 729$	-1198.25	-1159.03	1002	$\geq 10^6$	-1173.33	1003	$\geq 10^6$	1.57 6.84	0 **-1194.22**	597 1158
$d = 10000$ $n = 739$	-21.08	-18.13	1009	$\geq 10^8$	-0.83	1004	$\geq 10^9$	147.63	**-21.08**	725

Table 3: Performance under Model 2 via CGAL ($\sigma = 20$). Time limit is set to 1000s, except for one instance for $d = 9000$. Two rows for each of the instances $d = 3000, 4000, 5000, 9000$ are reported, detailing the performance of CGAL + Algorithm 1 for varying iteration of CGAL (m). Specifically, for $d = 4000, 5000, 9000$, m is set to 20 in the first row and 40 in the second. For $d = 2000$, m is set to 20 and 50 in the first and second rows, respectively.

G.2.2 Performance of recovery under Model 2.

In this section, we provide numerical results on the ability of SILS'-SDP recovering z^* under Model 2, by plotting the empirical probability of recovery of z^*.

In Figure 4, we study the setting where $z^* = \begin{pmatrix} a \\ 0_{d-\sigma} \end{pmatrix}$ with a uniformly drawn in $\{\pm 1\}^\sigma$. We plot the empirical probability of recovery of z^* for each $n = \lceil c\varrho^2\sigma^2 \log(d) \rceil$, with control parameter c ranging from 1 to 15. As predicted in Theorem 9, when c is large enough, the empirical probability of recovery of z^* goes to 1 as the control parameter c increases. Empirically, we also observe there is a transition to failure of recovery when the control parameter c is sufficiently small.

G.2.3 Statistical performance under Model 3.

In this section, we report the numerical performance of SILS'-SDP in the integer sparse recovery problem under Model 3, as studied in Section 4.2.2. Note that Model 3 has a low coherence when $n \geq \sigma^2 \log(d)$. We restrict ourselves to the scenario where each entry of M is i.i.d. standard Gaussian, $z^* = \begin{pmatrix} a \\ 0_{d-\sigma} \end{pmatrix}$ with a uniformly drawn in $\{\pm 1\}^\sigma$, and $\epsilon \sim \mathcal{N}(0_d, \varrho^2 I_d)$.

In Figure 5, we plot the empirical probability of recovery of z^*, for each $n = \lceil c(\sigma^2 + \varrho^2) \log(d) \rceil$ with control parameter c ranging from 1/8 to 2. As predicted in Proposition 10, when c grows, the probability that SILS'-SDP recovers z^* goes to 1. Empirically, we also observe that there is a transition to failure of recovery when the control parameter c is sufficiently small.

In Figure 6, we compare the numerical performance of SILS'-SDP, Lasso, and DS. From [53]

		SBQP			MIO			CGAL + Algorithm 1		
	obj_{z^*}	obj	time	mipgap	obj	time	mipgap	mean val	best val	time
$d = 1000$ $n = 553$	-18.86	**-18.86**	89	0	**-18.87**	82	0	-0.09	**-18.86**	13
$d = 2000$ $n = 609$	-22.43	**-22.43**	689	0	**-22.43**	746	0	0.11	**-22.43**	39
$d = 3000$ $n = 641$	-20.11	0	1001	-	-13.56	1000	$\geq 10^4$	1.22	**-20.11**	79
$d = 4000$ $n = 664$	-20.68	0	1001	-	-13.80	1001	$\geq 10^5$	0.91	**-18.54**	132
$d = 5000$ $n = 682$	-19.88	0	1001	-	**-19.89**	1001	$\geq 10^5$	0.91	**-19.88**	197
$d = 6000$ $n = 696$	-19.72	0	1001	-	-17.25	1001	$\geq 10^4$	0.74	**-18.04**	276
$d = 7000$ $n = 709$	-20.94	0	1001	-	-14.88	1001	$\geq 10^4$	0.57	**-20.94**	369
$d = 8000$ $n = 719$	-22.27	0.49	1001	$\geq 10^6$	-12.18	1001	$\geq 10^5$	0.25	**-22.27**	482
$d = 9000$ $n = 729$	-20.32	0	1001	-	-17.15	1002	$\geq 10^5$	0.29	**-20.32**	610
$d = 10000$ $n = 729$	-21.08	0	1001	-	-17.73	1002	$\geq 10^5$	0.23	**-21.08**	734

Table 4: Performance under Model 3 via CGAL ($\sigma = 20$). Time limit is set to 1000s.

		SBQP			MIO			CGAL + Algorithm 1		
	obj_{z^*}	obj	time	mipgap	obj	time	mipgap	mean val	best val	time
$\sigma = 2$	-2.46	**-2.46**	0.09	0	**-2.46**	0.077	0	0.86	**-2.46**	0.17
$\sigma = 5$	-7.22	**-7.22**	0.08	0	**-7.22**	0.10	0	2.66 3.47	-7.01 **-7.22**	0.11 0.12
$\sigma = 6$	-6.76	**-6.76**	0.05	0	**-6.76**	0.11	0	3.57	**-6.76**	0.15

Table 5: Performance under diabete dataset ($d = 10, n = 442$). Objectives are scaled by 10^6. Two rows for the instance $\sigma = 5$ are reported, detailing the performance of CGAL + Algorithm 1 for varying iteration of CGAL (m). m is set to 20 in the first row and 50 in the second.

and [33], we know that Lasso and DS converge to z^*, provided that we set $\lambda = 2\sqrt{\log(d)/n}$ in Lasso and $\eta = 2\varrho(5/4 + \sqrt{\log(d)})$ in DS. Hence, we set the parameters λ and η to these values without performing cross-validation. In Figure 6, we report three significant quantities: the first two are the number of nonzeros and the true positive rate, as defined in Section 5.2. The third one is the *successful recovery rate*, defined as

$$\text{successful recovery rate}(z) := \frac{|\operatorname{Supp}(z^*) \cap S^\sigma_{\max}(z)|}{|\operatorname{Supp}(z^*)|},$$

where $S^\sigma_{\max}(z)$ is the set indices corresponding to the top σ entries of z having largest absolute values. The reason we consider here the successful recovery rate instead of the prediction error, considered for Model 2, is that in all three algorithms z converges to z^* in Model 3. Hence, for n large enough, $|z_i|$ is close to 0 when $z^*_i = 0$, and $|z_j|$ is close to one if $z^*_j = \pm 1$. Hence, we can recover z^* by simply looking at the σ largest entries of $|z|$. We conclude from Figure 6 that all three algorithms obtain great results in Model 3, and this is mainly due to the low coherence of the model. Since all three algorithms perform well, Lasso and DS should be preferred since they run significantly faster than SILS'-SDP. In particular, SILS'-SDP can be solved in about

		SBQP			MIO			CGAL + Algorithm 1		
	obj_{z^*}	obj	time	mipgap	obj	time	mipgap	mean val	best val	time
$\sigma = 2$	-18.00	12.44	600	$\geq 10^7$	9.55	600	$\geq 10^7$	141.189	**0**	119
$\sigma = 5$	-138.33	-71.17	602	174%	-74.78	603	161%	361.38	**-78.98**	114
$\sigma = 10$	-800.06	-829.27 -431.01	604 196	5.84% 104%	**-839.96** -630.68	600 209	4.5% 104%	372.58	-716.17	107
$\sigma = 20$	-986.59	-999.89 -282.92	602 121	-6.23% 275%	**-1023.31** -812.17	601 136	3.79% 30.8%	727.31	-645.22	107

Table 6: Performance under leukemia dataset ($d = 3571, n = 72$). Time limit is set to 600s. For the instances $\sigma = 10$ and $\sigma = 20$, the results for SBQP and MIO are presented in two rows. The first row details the optimal solution obtained upon time limit, while the second row captures the best solution at approximately the termination time of CGAL + Algorithm 1.

		SBQP			MIO			CGAL + Algorithm 1		
	obj_{z^*}	obj	time	mipgap	obj	time	mipgap	mean val	best val	time
$\sigma = 2$	-70.08	-6.23	1002	$\geq 10^8$	52.84	1002	$\geq 10^7$	64.26	**-38.34**	323
$\sigma = 5$	-398.16	-287.61	1003	$\geq 10^6$	4021.42	1002	$\geq 10^5$	179.96	**-315.16**	325
$\sigma = 10$	-261.57	51.51	1003	$\geq 10^7$	369.63	1002	$\geq 10^6$	523.69	**-131.13**	313
$\sigma = 20$	-792.74	-232.24	1003	$\geq 10^6$	1170.91	1002	$\geq 10^6$	883.52	**-434.35**	314

Table 7: Performance under prostate dataset ($d = 6033, n = 102$). Time limit is set to 1000s. The number of iterations of CGAL is set to $m = 50$.

one second with $d = 40$ and in about one minute with $d = 100$, while the other two can be solved in less than 0.1 second in both cases.

G.3 Additional results for non-convex objectives

In this section, we report additional numerical results on the performance of applying Algorithm 1 to SBQP with indefinite matrix P. In Appendix G.3.1, we solve SDP(c, P) using general SDP solver Mosek and evaluate the performance of the algorithm on datasets with indefinite P. In Appendix G.3.2, we employ CGAL, as proposed by [56], to find approximate solutions to SDP(c, P). We compare the outcomes of CGAL + Algorithm 1 on the same datasets

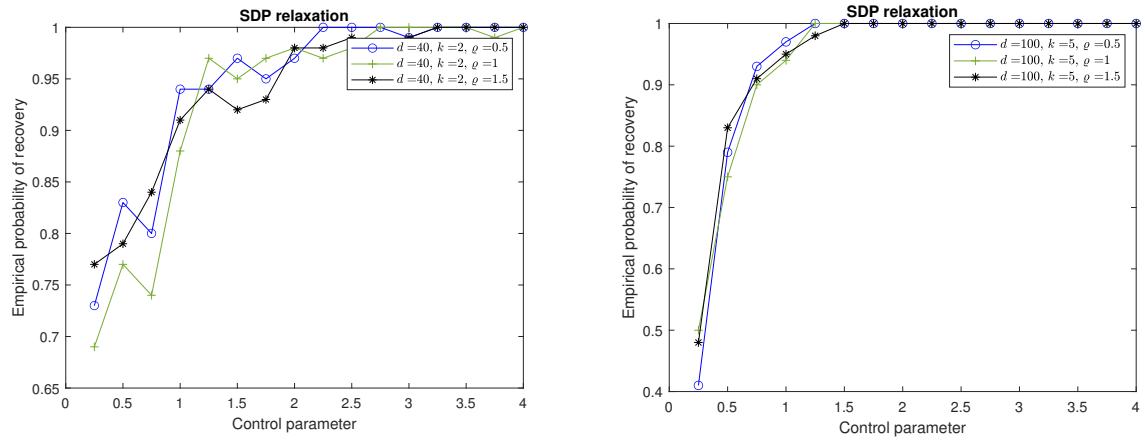

Figure 2: Performance of SILS'-SDP under Model 1: empirical probability of recovery.

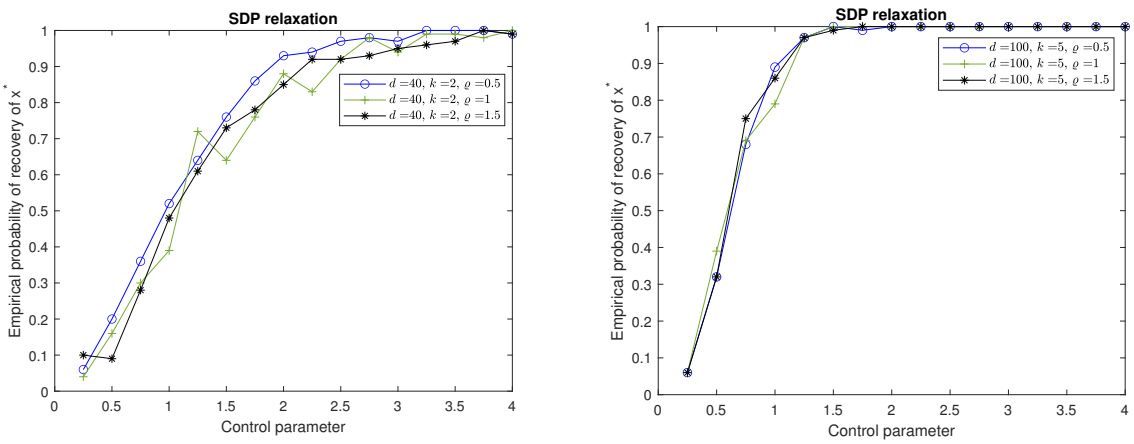

Figure 3: Performance of SILS'-SDP under Model 1: empirical probability of recovery of x^*.

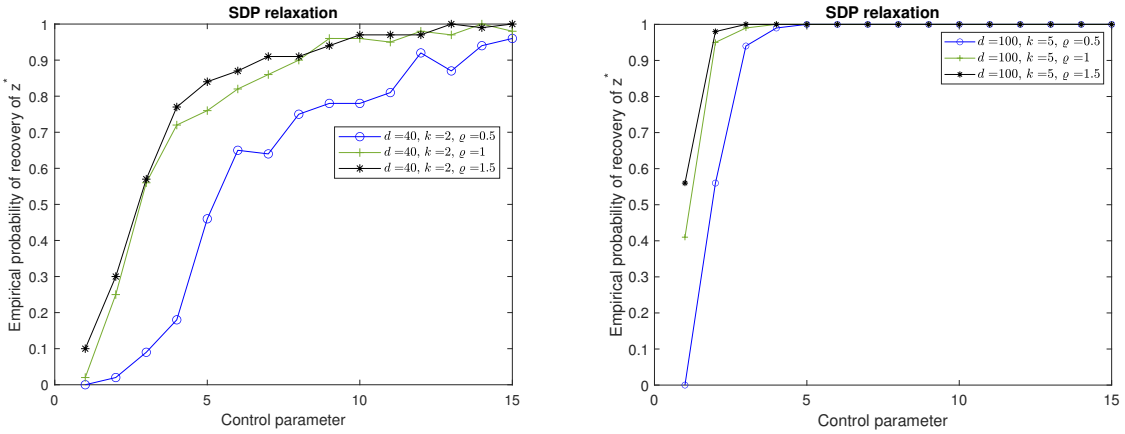

Figure 4: Performance of SILS'-SDP under Model 2: empirical probability of recovery of z^*.

used in Appendix G.3.1 and demonstrate that CGAL + Algorithm 1 not only accelerates the computation but also maintains excellent solution quality relative to exact solutions followed by Algorithm 1.

G.3.1 Solving SDP(c, P) via Mosek.

In this section, we test the performance of Algorithm 1 under a Binary Quadratic Programming (BQP) benchmark maintained by J E Beasley [5]. We need to clarify that the benchmark is not initially intended for SILS or SBQP, but we believe using the data therein will provide interested readers a sense of how Algorithm 1 performs under real-world datasets with indefinite matrix input. We utilize the symmetric matrices provided therein as P in SBQP, and zero out all negative entries on diagonal to keep aligned with the assumptions in Theorem 1. Note that the matrix P is not necessarily positive semidefinite, and hence SBQP can be a non-convex problem. Since the vector c is not provided by the benchmark data set, we generate it as a random vector $c \sim \mathcal{N}(0_d, I_d)$. Due to the large number of testing problems, we only report the performance on the first two benchmark data sets, for different sets of σ.

We summarize the results in Tables 8 to 10, where we take $T = \sqrt{\log d}$ and $C = 0.1$ as input threshold constants in Algorithm 1. After finding an (approximate) optimal solution to SILS-SDP via Mosek, we run Algorithm 1 for a thousand times, and report the mean value of objective value for $\bar{x}$ in SBQP (mean val), and also report the best $\bar{x}$ that is feasible to SBQP and that achieves the minimum objective value (best val). Since we need to find out the

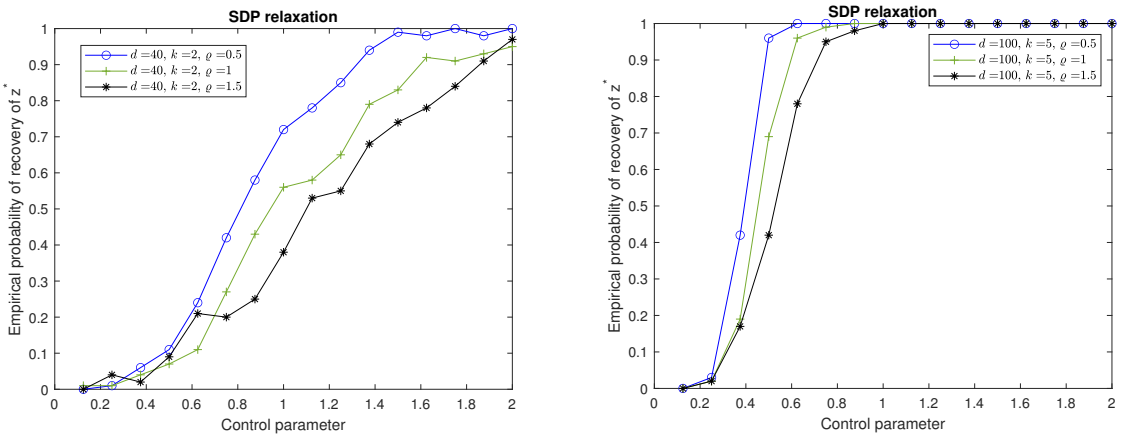

Figure 5: Performance of SILS'-SDP under Model 3: empirical probability of recovery of z^*.

optimal value of SBQP and SDP(c, P), we also report the running time of these two programs for interested readers. The time limit for SBQP is 45000 seconds (12.5 hours), and we report the MIP gap generated by Gurobi as well. It should be pointed out that running time comparison is not the main focus of this paper, as the main focus of this paper is the approximability and even the solvability of SILS and SILS' in polynomial time. It can be seen from the tables that the approximation gap indeed holds as proposed in Theorem 1. Moreover, we are surprised to see that best value obtained by Algorithm 1 seems to differ from the true optimal value by a constant multiple, which suggests that Algorithm 1 is more practical than what Theorem 1 states.

	SBQP			SDP(c, P)		Algorithm 1	
	optval	time	mipgap	optval	time	mean val	best val
$\sigma = 2$	-197.26	0.35	0	-201.42	2.14	-1.59	-185.09
	-200.73	0.13	0	-213.89	2.19	-2.89	-186.04
$\sigma = 5$	-830.42	0.17	0	-936.11	2.41	-13.25	-778.68
	-935.56	0.17	0	-1002.38	3.30	-14.04	-661.71
$\sigma = 10$	-1743.66	1.12	0	-2112.93	3.98	-21.15	-1113.5
	-2327.86	0.21	0	-2509.01	3.57	-30.78	-1362.18
$\sigma = 20$	-3692.45	4.64	0	-4324.59	3.15	-58.67	-2576.74
	-4902.50	0.30	0	-5356.67	3.06	-77.17	-3530.11

Table 8: Performance under BQP50 ($d = 50$)

However, from a practical standpoint, Algorithm 1 encounters two significant challenges: (a) While SDPs can theoretically be solved in polynomial time up to an arbitrary accuracy, existing solvers such as Mosek exhibit limited scalability. This limitation becomes evident when addressing instances such as SDP(c, P) for $d \geq 250$ in practice, resulting in a computational time of approximately one hour as shown in Table 10. (b) The algorithm also exhibits a substantial optimality gap. For instance, in Table 9 for $\sigma = 20$, the optimal value is -5446, but the best result achieved through Algorithm 1 is only -2308. Despite predictions from Theorem 2 that Algorithm 1 could achieve a $(1/\log d)$-approximation, further investigation into the class of inputs that enhance the performance of Algorithm 1 is essential for improving solution quality and algorithmic practicality.

To address issue (a), we find that employing CGAL not only accelerates computation compared to traditional SDP solvers like Mosek but also maintains the quality of inputs for Al-

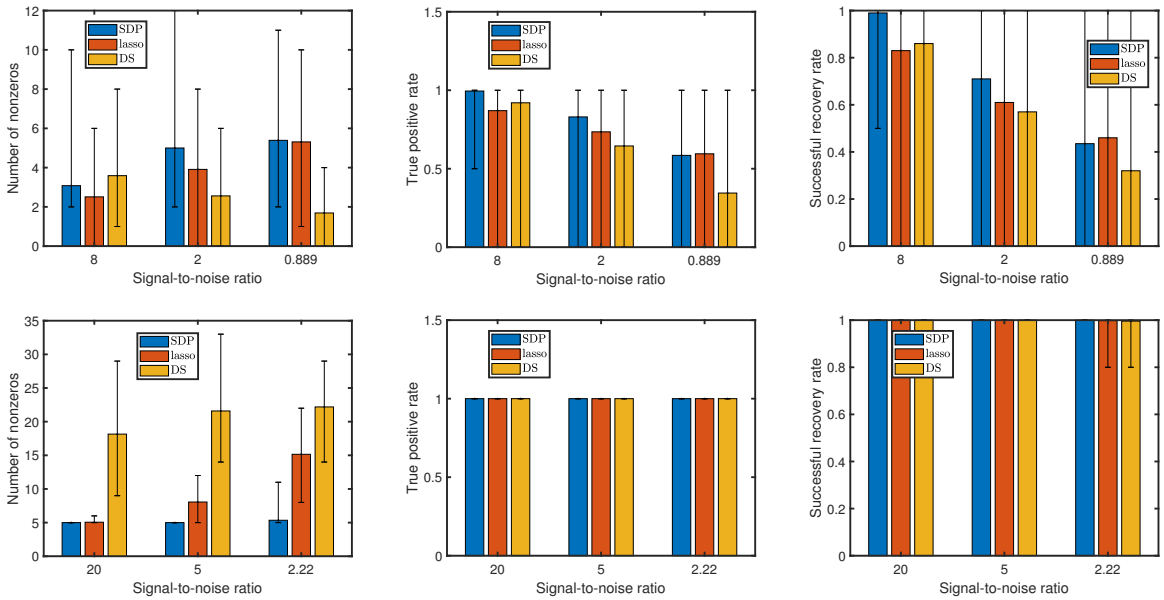

Figure 6: Performance of SILS'-SDP, Lasso, and DS under Model 3, with $d = 40$, $\sigma = 2$, $n = \lceil \sigma^2 \log(d) \rceil = 15$ in the first row, and with $d = 100$, $\sigma = 5$, $n = \lceil \sigma^2 \log(d) \rceil = 116$ in the second row. 100 instances are considered with $\varrho \in \{0.5, 1, 1.5\}$. The average is reported in the histogram, and the minimum and maximum in the box plot.

	SBQP			SDP(c, P)		Algorithm 1	
	optval	time	mipgap	optval	time	mean val	best val
$\sigma = 2$	-202.11	1.33	0	-253.88	31.27	-3.19	-198.54
	-205.17	1.00	0	-218.26	38.59	-1.49	-196.10
$\sigma = 5$	-1062.05	5.41	0	-1225.52	52.14	-15.63	-588.83
	-944.07	12.36	0	-1052.31	56.01	-9.17	-584.16
$\sigma = 10$	-2470.12	255.83	0	-2897.89	74.83	-32.09	-1535.83
	-2254.50	472.24	0	-2647.47	53.38	-18.81	-905.34
$\sigma = 20$	-5445.56	44254.76	0	-6457.71	54.35	-56.92	-2308.37
	-5146.21	40999.62	0	-6175.32	42.63	-54.23	-1899.83

Table 9: Performance under BQP100 ($d = 100$)

gorithm 1, as we will see in Appendix G.3.2. Regarding issue (b), we have not yet developed a method to improve the performance of Algorithm 1 on SBQP with non-convex objective functions. However, as shown in Appendix G.1, Algorithm 1 performs significantly better on SBQP with a convex objective function, which is the primary focus of this paper. Enhancing its performance in the non-convex setting remains an interesting direction for future research.

G.3.2 Solving SDP(c, P) via CGAL.

In this section, we illustrate the capabilities of CGAL by applying it to the BQP instances examined previously. The objective here is to demonstrate that employing CGAL to solve SDP(c, P) does improve the efficacy of Algorithm 1.

Similar to Appendix G.1, we limit CGAL to 20 iterations. Additionally, considering that optimal solutions to SBQP have already been reported in Tables 8 to 10, and given the interest in comparing the solution of Gurobi with that of SILS combined with Algorithm 1 with the same time limit, we only present results from Gurobi under a 2-second time limit for all instances,

	SBQP			SDP(c, P)		Algorithm 1	
	optval	time	mipgap	optval	time	mean val	best val
$\sigma = 2$	-205.73	16.60	0	-261.00	2772.25	-3.72	-200.41
	-207.92	8.54	0	-245.64	2877.39	-4.75	-197.25
$\sigma = 5$	-1250.38	2866.28	0	-1353.48	3335.86	-15.15	-728.86
	-1202.20	4366.75	0	-1287.54	4854.88	-13.02	-980.39
$\sigma = 10$	-3037.26	45000	46.9%	-3599.40	4891.06	-17.82	-1333.36
	-2919.55	45000	55.3%	-3741.36	4285.94	-23.92	-1051.18
$\sigma = 20$	-7363.80	45000	48.3%	-8970.73	4368.48	-41.21	-2717.95
	-6871.36	45000	56.9%	-8648.22	3615.02	-37.63	-2258.16

Table 10: Performance under BQP250 ($d = 250$)

as the runtime for CGAL + Algorithm 1 consistently remains below this limit. The results are summarized in Tables 11 to 13.

	SBQP			SDP(c, P) via CGAL		Algorithm 1	
	optval	time	mipgap	optval	time	mean val	best val
$\sigma = 2$	-197.26	0.31	0	-673.86	0.61	-29.53	-195.10
	-200.73	0.29	0	-868.30	0.03	-50.56	-199.51
$\sigma = 5$	-830.42	0.31	0	-1306.67	0.03	-98.61	-770.08
	-935.56	0.54	0	-1636.98	0.08	-141.63	-827.18
$\sigma = 10$	-1743.66	2	7.90%	-2453.15	0.04	-314.82	-1675.73
	-2327.86	0.46	0	-3044.46	0.02	-377.37	-2013.09
$\sigma = 20$	-3692.45	2	10.89%	-4647.30	0.04	-635.05	-3188.06
	-4902.50	0.80	0	-5807.56	0.03	-962.42	-4028.02

Table 11: Performance under BQP50 using CGAL ($d = 50$)

Compared to the results detailed in Tables 8 to 10, we note the following observations: (i) the use of CGAL + Algorithm 1 is effective and maintains the quality of the obtained solutions compared to Mosek + Algorithm 1. Remarkably, in approximately 75% of the instances, the best value increases, though this could be attributed to the stochastic nature of Algorithm 1; (ii) CGAL accelerates the resolution of SDP(c, P), albeit at the expense of a less accurate SDP objective value. It is important to note that the SDP objective values presented in Tables 11 to 13 might be misleading for those solely focused on solving SDP(c, P) with the specified inputs; (iii) Although CGAL significantly accelerates the solving process for SDP(c, P), the objective gap between solving SBQP with Gurobi and the accelerated method is still big with larger d and σ. We identify the indefinite nature of the input matrix P as one of the primary factors contributing to this issue. In contrast, as demonstrated in Appendix G.1, if P is positive semidefinite, the objective gap between CGAL + Algorithm 1 and SBQP significantly narrows.

	SBQP			SDP(c, P) via CGAL		Algorithm 1	
	optval	time	mipgap	optval	time	mean val	best val
$\sigma = 2$	-202.11	2	29.2%	-1133.54	0.06	-54.59	-200.44
	-205.17	2	143%	-997.86	0.06	-14.47	-186.96
$\sigma = 5$	-1061.94	2	5.84%	-2183.13	0.06	-132.32	-876.35
	-881.63	2	92.98%	-1970.00	0.05	-51.66	-603.85
$\sigma = 10$	-2111.97	2	109%	-3961.33	0.06	-245.01	-1322.19
	-2162.17	2	86.68%	-3601.32	0.06	-188.29	-1203.96
$\sigma = 20$	-5278.27	2	75%	-7290.32	0.06	-699.96	-3806.45
	-5023.71	2	57.1%	-6909.99	0.06	-585.03	-2834.50

Table 12: Performance under BQP100 using CGAL ($d = 100$)

	SBQP			SDP(c, P) via CGAL		Algorithm 1	
	optval	time	mipgap	optval	time	mean val	best val
$\sigma = 2$	-205.73	2	605%	-1740.68	0.26	-13.55	-194.51
	-203.98	2	589%	-1669.47	0.26	-5.22	-198.13
$\sigma = 5$	-1105.80	2	241%	-3475.37	0.26	-60.34	-906.50
	-1110.26	2	224%	-3313.33	0.26	-31.89	-548.65
$\sigma = 10$	-2600.77	2	175%	-6259.85	0.27	-147.82	-1310.93
	-2398.98	2	197%	-5971.82	0.29	-101.60	-1275.57
$\sigma = 20$	-6024.32	2	139%	-11853.28	0.26	-534.21	-3464.31
	-6043.13	2	138%	-11200.42	0.26	-345.32	-2451.18

Table 13: Performance under BQP250 using CGAL ($d = 250$)