Sparse PCA With Multiple Components

Ryan Cory-Wright

Department of Analytics, Marketing and Operations, Imperial Business School, London, UK
ORCID: 0000-0002-4485-0619
r.cory-wright@imperial.ac.uk

Jean Pauphilet

Management Science and Operations, London Business School, London, UK
ORCID: 0000-0001-6352-0984
jpauphilet@london.edu

Sparse principal component analysis is a fundamental technique for obtaining interpretable combinations of features, or principal components (PCs), that explain the variance of high-dimensional datasets. This involves solving a sparsity- and orthogonality-constrained convex maximization problem, which is extremely computationally challenging. Most existing work addresses sparse PCA via methods—such as iteratively computing one sparse PC and deflating the covariance matrix—that do not guarantee the orthogonality, let alone the optimality, of the resulting solution when we seek multiple mutually orthogonal PCs. We challenge this status quo by investigating three competitive approaches that each produce valid relaxations and high-quality solutions. Together, these relaxations and their associated heuristics yield solutions with an average bound gap on the order of 3% for real-world datasets with hundreds or thousands of features and $r \in \{2, 3\}$ components. The first approach reformulates orthogonality conditions as rank constraints and thereby derives tight semidefinite relaxations, strengthened via additional second-order cone inequalities. The second approach uses Lagrangian decompositions to relax the orthogonality constraints via a penalty term in the objective, thus solving the problem as a sequence of $r = 1$ sparse PC problems. The third approach is based on a new combinatorial upper bound on the variance explained for a given support pattern, obtained by solving a mixed-integer linear optimization problem. Numerically, our algorithms match (and sometimes surpass) the best-performing methods in terms of the fraction of variance explained and systematically return PCs that are sparse and orthogonal. In contrast, we find that existing methods, such as deflation, return solutions that violate the orthogonality constraints, even when the data are generated from sparse orthogonal PCs. Altogether, our approaches solve sparse PCA problems with multiple components to certifiable near-optimality in a practically tractable fashion.

Key words: Sparse Principal Component Analysis; Semidefinite Optimization; Practical Tractability

1. Introduction

Principal Component Analysis (PCA), first proposed by Pearson (1901), is one of the most popular techniques for reducing the dimension of a dataset (see also Hotelling 1933, Eckart and Young 1936). Given a normalized and centered data matrix $\boldsymbol{A} \in \mathbb{R}^{n \times p}$, and its sample covariance matrix $\boldsymbol{\Sigma} := \frac{1}{n-1} \boldsymbol{A}^\top \boldsymbol{A}$, the top r principal components ($r \ll p$) of $\boldsymbol{\Sigma}$ are obtained by solving:

$$\max_{\boldsymbol{U} \in \mathbb{R}^{p \times r}} \quad \langle \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{\Sigma} \rangle \text{ s.t. } \boldsymbol{U}^\top \boldsymbol{U} = \mathbb{I}. \tag{1}$$

As described by Hotelling (1933), the principal components of $\boldsymbol{\Sigma}$ correspond to its r leading eigenvectors and can efficiently be obtained via a greedy procedure: at each iteration, one computes the leading eigenvector of $\boldsymbol{\Sigma}$, $\boldsymbol{u}$, (for instance, by solving (1) with $r = 1$) and then updates (or *deflates*) $\boldsymbol{\Sigma}$ to eliminate the influence of $\boldsymbol{u}$. PCA is now a fundamental unsupervised learning paradigm that is practically useful across a range of fields, including pattern recognition (Naikal et al. 2011), sequence classification (Tan et al. 2014), factor models in finance (Fan et al. 2016), and variable selection in statistics (Wang et al. 2023).

Despite efficient modern implementations (Udell et al. 2016, Tropp et al. 2017), PCA suffers from two limitations. First, it generates components that are dense linear combinations of the original features and hence uninterpretable (Rudin et al. 2022). Second, it yields inconsistent estimates in high-dimensional settings where $p/n \to \alpha > 0$ (Johnstone and Lu 2009). Accordingly, several authors (such as Jolliffe et al. 2003, d'Aspremont et al. 2007) have proposed sparse PCA, namely, augmenting Problem (1) with a sparsity constraint. When $r = 1$, sparse PCA can be formulated as the following optimization problem:

$$\max_{\boldsymbol{u} \in \mathbb{R}^p} \quad \langle \boldsymbol{u}\boldsymbol{u}^\top, \boldsymbol{\Sigma} \rangle \text{ s.t. } \|\boldsymbol{u}\|_2^2 = 1, \|\boldsymbol{u}\|_0 \leq k, \tag{2}$$

where $\|\boldsymbol{u}\|_0$ denotes the cardinality of $\boldsymbol{u}$ or the size of its support: $\|\boldsymbol{u}\|_0 = |\operatorname{supp}(\boldsymbol{u})| = |\{j : \boldsymbol{u}_j \neq 0\}|$. From a statistical recovery perspective, suppose that the data are generated according to a 'true' covariance matrix of the form $\boldsymbol{\Sigma}^\star = \beta \boldsymbol{v}\boldsymbol{v}^\top + \boldsymbol{I}_p$, for some $\boldsymbol{v}$ with $\|\boldsymbol{v}\|_0 \leq k$ and some $\beta > 0$. Then, Amini and Wainwright (2008) have shown that an exhaustive search algorithm can reliably identify the support of $\boldsymbol{v}$ provided that the number of samples satisfies $n \gtrsim k \log(p)$, which constitutes a significant improvement over the traditional PCA formulation that requires $n \gtrsim p$. Moreover, they have also shown that no method can succeed when $n \lesssim k \log(p)$, because of information-theoretic limitations. Berthet and Rigollet (2013) analyze the support recovery ability of polynomial-time algorithms and show that no polynomial-time algorithm can succeed when $n \lesssim k^2$ (assuming planted clique cannot be solved in randomized polynomial time). Hence, there exists a regime, $k \log(p) \lesssim n \lesssim k^2$, where exhaustive search successfully detects the support of $\boldsymbol{v}$, while no polynomial-time algorithm can do so.

This gap motivated the development of tailored discrete optimization algorithms to efficiently implement exhaustive search for sparse PCA with a single PC. Indeed, Problem (2) can be formulated as a mixed-integer semidefinite optimization problem and solved via global optimization techniques such as branch-and-bound (Berk and Bertsimas 2019), or branch-and-cut (Bertsimas et al. 2022b). Confirming the statistical theory, certifiably optimal methods for (2) are often significantly more accurate than polynomial-time methods.

To our knowledge, there is no widely agreed-upon formulation in the literature that generalizes Problem (2) to $r > 1$, and there are no practically relevant algorithms with optimality guarantees

that successfully address this extension. Indeed, most algorithmic work for sparse PCA with one PC cannot be readily generalized to the case with $r > 1$. This is because, in the multiple component case, the sparsity constraint in (2) causes sparse principal components to no longer be eigenvectors of $\boldsymbol{\Sigma}$ and deflation methods to no longer lead to orthogonal, let alone optimal, PCs (Mackey 2008). In other words, the conventional PCA wisdom that multiple components can be computed one by one fails as soon as sparsity is required. This observation calls for new methods for sparse PCA with multiple PCs that simultaneously optimize the components. In response, we study a generic optimization formulation that extends Problem (2) to $r > 1$, and investigate three competitive approaches that produce provably near-optimal solutions with certificates of optimality.

1.1. A Generic Formulation for Sparse PCA with Multiple PCs

Perhaps the most natural extension of sparse PCA to multiple principal components, and the one which we advocate in this paper, is to augment Problem (1) with a constraint on the number of non-zero entries in the matrix $\boldsymbol{U}$,

$$\|\boldsymbol{U}\|_0 = |\mathrm{supp}(\boldsymbol{U})| = |\{(i,j) \in [p] \times [r] : U_{i,j} \neq 0\}| \leq k.$$

This gives a formulation which enforces two desirable properties on the matrix $\boldsymbol{U}$: orthogonality ($\boldsymbol{U}^\top \boldsymbol{U} = \mathbb{I}$), as present in the prototypical formulation of PCA (see, e.g., Horn and Johnson 1985); and sparsity, to address interpretability and accuracy concerns (cf. Rudin et al. 2022).

Formally, introducing a binary matrix $\boldsymbol{Z}$ to encode the support of $\boldsymbol{U}$, we consider the problem:

$$\max_{\boldsymbol{Z} \in \{0,1\}^{p \times r}} \max_{\boldsymbol{U} \in \mathbb{R}^{p \times r}} \quad \langle \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{\Sigma} \rangle \quad \text{s.t.} \quad \boldsymbol{U}^\top \boldsymbol{U} = \mathbb{I}, \tag{3}$$
$$U_{i,t} = 0 \text{ if } Z_{i,t} = 0, \ \forall i \in [p], \ \forall t \in [r],$$
$$\sum_{i \in [p], t \in [r]} Z_{i,t} \leq k \text{ or } \sum_{i \in [p]} Z_{i,t} \leq k_t, \ \forall t \in [r],$$

where $[r]$ denotes the index set $\{1, \ldots, r\}$. We either impose a global sparsity constraint on $\boldsymbol{U}$, $\|\boldsymbol{U}\|_0 \leq k$, via $\sum_{i \in [p], t \in [r]} Z_{i,t} \leq k$ (with $pr > k > r$, in this case, for the problem to be well-posed and non-trivial), or a per-component sparsity requirement, $\|\boldsymbol{U}_t\|_0 \leq k_t$, via the constraints $\sum_{i \in [p]} Z_{i,t} \leq k_t$ (with $0 < k_t < p$). We address both modeling options in this paper, although the second option is more common in the literature (e.g., Hein and Bühler 2010, Deshpande and Montanari 2014a, Berk and Bertsimas 2019).

Observe that optimal solutions to Problem (3) may not be eigenvectors of a submatrix of $\boldsymbol{\Sigma}$, especially when the support of the columns of $\boldsymbol{Z}$ is partly overlapping. This fundamental difference from sparse PCA with one PC makes Problem (3) substantially more difficult to solve.

We remark that although Problem (3) is a very natural extension of Problem (2), we are not aware of any works that explicitly formulate sparse PCA with multiple components as an orthogonality-constrained problem with logical constraints. Nor are we aware of any works that study (3) at this

level of generality. As we review in the next section, many works implicitly study (3) by computing PCs that are simultaneously sparse and explain most of the dataset's variance without defining them as formal solutions of a cardinality- and orthogonality-constrained optimization problem. In the discrete optimization literature, several authors study special cases of (3) with additional constraints on $\boldsymbol{Z}$ that ensure it admits a mixed-integer semidefinite reformulation (row sparsity and disjoint sparsity, see below). In practice, however, the support of each PC may neither be disjoint nor fully overlapping. For example, in stock return data, each PC can be interpreted as a long-short portfolio of an industry sector (Avellaneda and Lee 2010), so we should not expect the supports of each PC to be perfectly disjoint nor fully overlapping. Closest to our formulation are PCA problems with orthogonality constraints and an ℓ_1 penalty term to induce sparsity (Zou et al. 2006, Lu and Zhang 2012, Vu et al. 2013, Benidis et al. 2016).

Finally, from a generative model perspective, Problem (3) is consistent with a spiked covariance model (see, e.g., Amini and Wainwright 2008, d'Aspremont et al. 2008), where the true covariance matrix $\boldsymbol{\Sigma}^\star$ can be decomposed as the sum of a sparse and low-rank term plus some noise:

$$\boldsymbol{\Sigma}^\star = \sum_{t \in [r]} \beta_t \boldsymbol{v}_t \boldsymbol{v}_t^\top + \boldsymbol{N}, \tag{4}$$

where $\boldsymbol{v}_t$ are sparse vectors that may have non-overlapping, partially overlapping, or completely overlapping support, and $\boldsymbol{N}$ is a noise matrix. This generative model is referred to as the spiked Wishart model when $\boldsymbol{N} = \boldsymbol{I}_p$, and as the spiked Wigner model when $\boldsymbol{N}$ is drawn from the Gaussian orthogonal ensemble (see Ding et al. 2024, for a general theory of both families of models).

1.2. Literature Review

To identify the extent to which the state-of-the-art for sparse PCA can be improved, we now review methods for approximately solving sparse PCA with multiple PCs.

Methods for sparse PCA with $r = 1$: Several polynomial-time algorithms have been proposed to obtain high-quality solutions to (2), including greedy heuristics (d'Aspremont et al. 2008), ℓ_1 relaxations (Zou et al. 2006, d'Aspremont et al. 2007, Dey et al. 2022a), linear regression-based estimators (Bresler et al. 2018, Behdin and Mazumder 2025), or thresholding techniques (Johnstone and Lu 2009, Deshpande and Montanari 2014b). Note that the covariance thresholding method of Deshpande and Montanari (2014b) provably recovers the support in the spiked Wishart model whenever $n \gtrsim k^2$, which is the best achievable rate for polynomial-time methods (Berthet and Rigollet 2013). Over the past decade, various authors including Gally and Pfetsch (2016), Bertsimas et al. (2022b), Kim et al. (2022), Li and Xie (2025) have shown that Problem (2) can be recast as a mixed-integer semidefinite optimization (MISDO) problem and have derived both high-quality solutions and valid dual bounds using this discrete optimization lens.

Deflation methods for sparse PCA: It is well known that an optimal solution to Problem (1) can be obtained via a greedy deflation procedure. Consequently, Mackey (2008) proposes a sparse extension of this deflation scheme where, at each iteration, a sparse PC is computed (e.g., by solving (2) or a relaxation) and $\boldsymbol{\Sigma}$ is updated by projecting out the eigenspace modeled by $\boldsymbol{u}$: $\boldsymbol{\Sigma}_{\text{new}} = (\mathbb{I} - \boldsymbol{u}\boldsymbol{u}^\top)\boldsymbol{\Sigma}(\mathbb{I} - \boldsymbol{u}\boldsymbol{u}^\top)$. Empirically, this method often performs reasonably well (see Berk and Bertsimas 2019, Section 5.3), particularly when Problem (2) is solved to global optimality; see also Hein and Bühler (2010), Bühler (2014) for a related deflation-based scheme. However, unlike in the traditional case, deflation need not return an optimal solution to (3)—see Asteris et al. (2015) for a simple four-dimensional example. Moreover, deflation need not even return a feasible solution, since the orthogonality constraint is not explicitly imposed by the method, and is therefore often violated in practice. Because they cannot guarantee the feasibility (i.e., orthogonality) of the returned PCs, we consider deflation-based methods as heuristic methods for our sparse PCA problem with multiple PCs. Furthermore, because of their iterative nature, deflation-based procedures can easily control the sparsity of each component but usually struggle to enforce global sparsity.

Methods for generic sparse PCA: A second approach for solving Problem (3) is to apply a heuristic that approximately optimizes all r PCs simultaneously, rather than sequentially. Among others, Zou et al. (2006) propose an alternating maximization scheme for an ℓ_1 relaxation of Problem (3), Journée et al. (2010) propose an iterative conditional gradient method to identify a local optimum of Problem (3) without the orthogonality constraints, Lu and Zhang (2012) apply an augmented Lagrangian method which solves an ℓ_1 relaxation of (3), Vu et al. (2013) solve a semidefinite relaxation of Problem (3)'s ℓ_1 relaxation, and Benidis et al. (2016) adopt a minorization-maximization approach which also solves Problem (3) approximately. Unfortunately, these approaches are often suboptimal even when $r = 1$, and provide no indication of the optimality gap. Indeed, none of these approaches explicitly control both the sparsity and orthogonality constraints, and therefore none of the methods reviewed here are guaranteed to return feasible solutions to (3). Note that the covariance thresholding method of Deshpande and Montanari (2014b) can be applied to the $r > 1$ case and can return PCs that are asymptotically orthogonal. However, for a given dataset, covariance thresholding cannot guarantee the orthogonality of the returned solution, as we observe empirically in Sections 5.2 and 5.3.

Row Sparsity: Motivated by tractability concerns, another line of work studies a special case of (3), namely row-sparse principal component analysis or principal component analysis with global support. This formulation replaces the sparsity constraint on $\boldsymbol{U}$ with one requiring that the matrix $\boldsymbol{U}$ has at most k non-zero rows, as advocated by Boutsidis et al. (2011), Probel and Tropp (2011),

Vu and Lei (2013). This rewrites sparse PCA as performing a top-r SVD on the $k \times k$ principal submatrix of $\mathbf{\Sigma}$, i.e.,

$$\max_{\boldsymbol{z} \in \{0,1\}^p : \boldsymbol{e}^\top \boldsymbol{z} \leq k} \max_{\boldsymbol{U} \in \mathbb{R}^{p \times r}} \quad \langle \boldsymbol{U}\boldsymbol{U}^\top, \mathbf{\Sigma} \rangle \text{ s.t. } \boldsymbol{U}^\top \boldsymbol{U} = \mathbb{I}_{r \times r}, \ U_{i,t} = 0 \text{ if } z_i = 0 \ \forall i \in [p], \forall t \in [r]. \qquad (5)$$

Problem (5) is a special case of Problem (3) where each PC has the same support, which corresponds to constraining $\boldsymbol{Z}$ such that $Z_{i,t_1} = Z_{i,t_2}, \ \forall t_1, t_2 \in [r], \forall i \in [p]$ in (3), and dividing the "k" in (3) by r. This restriction is advantageous from a computational perspective, but disadvantageous from a statistical one. Indeed, because of the global support assumption, Problem (5) can be reformulated as a mixed-integer semidefinite problem (see Bertsimas et al. 2022b, Li and Xie 2024, for derivations) and solved via relax-and-round and local search strategies (Li and Xie 2024), mixed-integer linear approximations (Li and Xie 2024, Bertsimas and Kitane 2023), mixed-integer second-order-cone approximations (Dey et al. 2022b), or branch-and-cut (Li and Xie 2024). Although derived for theoretical purposes and not implemented (let alone evaluated computationally), Del Pia (2023) also describes a partitioning procedure to solve (5). However, from the perspective of the generative model in (4), it is equivalent to making the very strong assumption that all leading eigenvectors $\boldsymbol{v}_t$ have the same sparsity pattern. Indeed, if one PC $\boldsymbol{v}_t$ is sparser than k ($\|\boldsymbol{v}_t\|_0 \leq k_t < k$) or two PCs have partially overlapping support then Problem (5) will nonetheless estimate that the support of each column $\boldsymbol{U}_t$ is of size k. Therefore, Problem (5) is vulnerable to consistently making false discoveries in the identification of relevant features.

Disjoint Sparsity: Another relevant special case of sparse PCA with multiple PCs is when the supports of all r columns of $\boldsymbol{U}$ are assumed mutually disjoint, as originally proposed by Asteris et al. (2015). This gives rise to the formulation:

$$\max_{\substack{\boldsymbol{Z} \in \{0,1\}^{p \times r} : \\ \langle \boldsymbol{E}, \ \boldsymbol{Z} \rangle \leq k, \boldsymbol{Z}\boldsymbol{e} \leq \boldsymbol{e}}} \max_{\boldsymbol{U} \in \mathbb{R}^{p \times r}} \quad \langle \boldsymbol{U}\boldsymbol{U}^\top, \mathbf{\Sigma} \rangle \text{ s.t. } \boldsymbol{U}^\top \boldsymbol{U} = \mathbb{I}_{r \times r}, \ U_{i,t} = 0 \text{ if } Z_{i,t} = 0 \ \forall i \in [p], \ t \in [r], \qquad (6)$$

where $\boldsymbol{E}$ denotes a matrix of all ones of the appropriate dimension. Note that (6) is a special case of (3) where we additionally require that $\sum_{t \in [r]} Z_{i,t} \leq 1 \ \forall i \in [p]$, i.e., that each feature can be included in at most one PC. Interestingly, this restriction allows (6) to be recast as an MISDO and solved as such (cf. Bertsimas et al. 2022b), although we are not aware of any work that has exploited this MISDO reformulation. An obvious criticism of formulation (6) is that, in practice, we may wish to include a feature in multiple PCs. Therefore, (6) is best thought of as a special case of sparse PCA. Indeed, if the true generative model involves vectors $\boldsymbol{v}_t$ with partially overlapping supports, then (6) cannot recover them. Nonetheless, as we explore in our numerical experiments (Section 5), disjoint solutions often perform well when k is very small relative to p. This is because, when $k \ll p$, there are often several disjoint submatrices that are near-optimal in the rank-one case, and selecting the leading PCs from each of them is often a reasonable approach in practice.

Summary of Sparsity Constraints: We now summarize and contrast the different types of sparsity patterns considered throughout the literature and in this paper (see Table 1). While special cases of Problem (3) admit formulations as mixed-integer semidefinite optimization problems (MISDO), sparse PCA with multiple PCs is a mixed-integer low-rank optimization problem (MIRO) in general, as we establish in the next section. Moreover, even when $\boldsymbol{Z}$ is fixed, we show that estimating the coefficients of $\boldsymbol{U}$ in Problem (3) is NP-hard (Theorem 3), while disjoint sparse PCA and row sparse PCA can both be solved in polynomial time once $\boldsymbol{Z}$ is fixed. Furthermore, for a fixed $\boldsymbol{Z}$, semidefinite relaxations of disjoint or row sparse PCA are tight, while they need not be for full sparse PCA (Section EC.10). This suggests that (full) sparse PCA is computationally harder than its special cases commonly studied in the literature (assuming $P \neq NP$).

Table 1 Types of sparsity for sparse PCA with multiple principal components. We formulate the constraints on $\boldsymbol{Z}$ assuming a global sparsity budget k. Similar constraints with a per-component budget k_t could be derived.

Name	Constraints on $\boldsymbol{Z} \in \{0,1\}^{p \times r}$	Formulation	Complexity when $\boldsymbol{Z}$ fixed	Reference
Disjoint Sparsity	$Z_{i,t_1} + Z_{i,t_2} \leq 1 \; \forall i \in [p], \; \forall t_1 \neq t_2 \in [r]$ $\sum_{i \in [p], t \in [r]} Z_{i,t} \leq k$	MISDO	P-time	Asteris et al. (2015)
Row Sparsity	$Z_{i,t_1} = Z_{i,t_2} \; \forall i \in [p], \; \forall t_1, t_2 \in [r],$ $\sum_{i \in [p], t \in [r]} Z_{i,t} \leq k$	MISDO	P-time	Boutsidis et al. (2011)
Sparsity	$\sum_{i \in [p], t \in [r]} Z_{i,t} \leq k$	MIRO	NP-hard	This paper

1.3. Contributions and Structure

To our knowledge, no existing algorithm solves sparse PCA problems with multiple components and obtains certificates of optimality, except in the aforementioned special cases of row sparsity or disjoint support. Accordingly, we undertake a detailed study of Problem (3) in its full generality.

This paper develops and evaluates three alternative strategies for obtaining both upper and lower bounds on Problem (3), i.e., feasible solutions with optimality certificates.

- In Section 2, we propose a semidefinite relax-and-round scheme (Algorithm 1). To do so, we first derive a sparse and low-rank reformulation of Problem (3). This reformulation differs from the single sparse PC case, where the rank constraints are redundant. Further, we propose a semidefinite relaxation strengthened by second-order cone valid inequalities. Finally, we propose an algorithm (Algorithm 1) that rounds an optimal solution to this semidefinite relaxation into r sparse PCs with disjoint support, hence giving orthogonal PCs by default. Even when the sparsity pattern of each PC is fixed, we show that computing the coefficients of each PC can be NP-hard (Theorem 3), justifying the need to restrict our attention to disjoint supports for Algorithm 1 to be tractable.

- In Section 3, we develop a Lagrangian alternating maximization method (Algorithm 2). First, we derive a relaxation of Problem (3) by penalizing the orthogonality constraints and demonstrate that this gives rise to an r-factor upper bound on the overall objective, which is computable by solving r different single component sparse PCA problems to optimality. Second, we leverage our relaxation to design an alternating maximization heuristic, which provides asymptotically feasible solutions.

- In Section 4, we propose a combinatorial relax-and-round approach (Algorithm 3). We first generalize the Gershgorin Circle Theorem to multiple PCs with sparsity constraints to obtain a mixed-integer linear upper bound on Problem (3)'s objective value as a function of the support of the PCs $\boldsymbol{Z}$ only. We then generate feasible solutions from this relaxation by focusing on solutions with disjoint supports.

Finally, in Section 5, we thoroughly evaluate the quality of these three approaches on a collection of UCI and synthetic datasets. In terms of upper bounds, we find that the semidefinite approach provides the strongest bounds (within 2–5% of the optimal solutions for the small instances that can be solved with a commercial solver) and can be solved in minutes for $p \approx 100$s. However, for larger problem sizes, the Lagrangian relaxation provides the best tightness/tractability trade-off. In terms of generating feasible solutions, our three algorithms generate systematically feasible solutions, while existing approaches routinely violate the orthogonality requirements. In addition, they also improve in terms of the fraction of variance explained, and collectively generate solutions with bound gaps of $< 3\%$ for p up to 1000.

Thus, the methods developed in this paper provide near-optimal solutions of the sparse PCA problem with multiple components in minutes for problems with $p \approx 1000$s, together with certificates of their near optimality, providing significant benefit compared with the state-of-the-art. For example, on the `pitprops` dataset, with 6 PCs, we explain 81% of the variance with an overall sparsity of 24, whereas previous studies (Lu and Zhang 2012) could explain less than 70% of the variance with twice as many variables.

1.4. Preliminaries and Notation

We let nonbold face characters such as u denote scalars, lowercase bold-faced characters such as $\boldsymbol{u}$ denote vectors, uppercase bold-faced characters such as $\boldsymbol{U}$ denote matrices, and calligraphic uppercase characters such as $\mathcal{Z}$ denote sets. If $\boldsymbol{U}$ is a matrix then $\boldsymbol{U}_t$ denotes the tth column vector of $\boldsymbol{U}$, and $U_{i,t}$ denotes the (i,t)th entry of $\boldsymbol{U}$. We let $[p]$ denote the index set $\{1, ..., p\}$. We let $\mathbf{e}$ denote a vector of all 1's, $\boldsymbol{0}$ denote a vector of all 0's, $\boldsymbol{E}$ denote a matrix of all ones, and $\mathbb{I}$ denote the identity matrix, with dimension implied by the context.

We also use an assortment of matrix operators. We let $\langle \cdot, \cdot \rangle$ denote the Euclidean inner product between two matrices, $\| \cdot \|_F$ denote the Frobenius norm, and $\mathcal{S}_+^p$ denote the $p \times p$ positive semidefinite cone; see Horn and Johnson (1985) for a general theory of matrix operators.

Further, we use some basic properties of orthogonal projection matrices. Let $\mathcal{Y}_n := \{ \boldsymbol{Y} \in \mathcal{S}^n : \boldsymbol{Y}^2 = \boldsymbol{Y} \}$ denote the set of $n \times n$ projection matrices and $\mathcal{Y}_n^k := \{ \boldsymbol{Y} \in \mathcal{Y}_n : \mathrm{tr}(\boldsymbol{Y}) \leq k \}$ denote projection matrices with rank at most k: note that $\mathrm{Rank}(\boldsymbol{Y}) = \mathrm{tr}(\boldsymbol{Y})$ for any projection matrix $\boldsymbol{Y}$. Among others, the convex hulls of $\mathcal{Y}_n$ and $\mathcal{Y}_n^k$ are well-studied, as we now remind the reader:

LEMMA 1. *(Theorem 3 of Overton and Womersley 1992)* $\mathrm{Conv}(\mathcal{Y}_n) = \{ \boldsymbol{P} : 0 \preceq \boldsymbol{P} \preceq \mathbb{I} \}$ *and* $\mathrm{Conv}(\mathcal{Y}_n^k) = \{ \boldsymbol{P} : 0 \preceq \boldsymbol{P} \preceq \mathbb{I}, \mathrm{tr}(\boldsymbol{P}) \leq k \}$. *Moreover, the extreme points of* $\mathrm{Conv}(\mathcal{Y}_n)$ *are* $\mathcal{Y}_n$, *and the extreme points of* $\mathrm{Conv}(\mathcal{Y}_n^k)$ *are* $\mathcal{Y}_n^k$.

2. Semidefinite Optimization Approach

In this section, we reformulate Problem (3) as a mixed-integer low-rank problem, study its semidefinite and second-order cone relaxations, and propose valid inequalities for strengthening them.

2.1. An Extended Formulation With Binary and Low-Rank Variables

Our sparse PCA formulation (3) is a mixed-integer quadratic optimization problem that exhibits three primary sources of difficulty. First, as is typical in PCA problems, Problem (3) maximizes a convex quadratic function in the decision variable $\boldsymbol{U}$. Second, there is a sparsity constraint. Finally, and more consequentially, there is an orthogonality constraint. To our knowledge, existing generic non-convex solvers such as `Gurobi` cannot optimize over such orthogonality constraints at a scale of p in the hundreds of features.

To address the three aforementioned difficulties, we now derive an orthogonality-free reformulation in five steps. First, we introduce the matrix $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{U}^\top \preceq \mathbb{I}$, thus linearizing the non-convex objective. Second, we introduce rank-one matrices $\boldsymbol{Y}^t$ to model the outer products of each column of $\boldsymbol{U}$, $\boldsymbol{U}_t$, with itself, $\boldsymbol{U}_t \boldsymbol{U}_t^\top$. Third, we reassign the indicator variable $Z_{i,t}$ to model whether $Y_{i,j}^t$, rather than $U_{i,t}$, is non-zero. Fourth, by letting $\boldsymbol{Y} = \sum_{t=1}^r \boldsymbol{Y}^t$, we observe that we can omit the matrix $\boldsymbol{U}$ (and the constraints involving $\boldsymbol{U}$) without altering the set of feasible $\boldsymbol{Y}$'s. Finally, we use the fact that $\boldsymbol{Y}_{i,j}^t$ is only supported on indices i, t where $Z_{i,t} > 0$ to strengthen the constraint $\boldsymbol{Y} \preceq \mathbb{I}$ to $\boldsymbol{Y} \preceq \mathrm{Diag}\left(\min\left(\boldsymbol{e}, \sum_t \boldsymbol{Z}_t \right) \right)$, where the minimization operator acts componentwise. We have:

THEOREM 1. *Problem (3) with the global sparsity constraint* $\langle \boldsymbol{E}, \boldsymbol{Z} \rangle \leq k$ *attains the same optimal objective value as the problem:*

$$\max_{\substack{\boldsymbol{Z} \in \{0,1\}^{p \times r} : \\ \langle \boldsymbol{E}, \boldsymbol{Z} \rangle \leq k}} \max_{\boldsymbol{Y} \in \mathcal{S}^p, \boldsymbol{Y}^t \in \mathcal{S}_+^p} \langle \boldsymbol{Y}, \boldsymbol{\Sigma} \rangle \; s.t. \quad \boldsymbol{Y} \preceq \mathrm{Diag}\left(\min\left(\boldsymbol{e}, \sum_t \boldsymbol{Z}_t \right) \right), \boldsymbol{Y} = \sum_{t=1}^r \boldsymbol{Y}^t, \tag{7}$$

$$\mathrm{tr}(\boldsymbol{Y}^t) = 1, \; \forall t \in [r], \; Y_{i,j}^t = 0 \; \text{if } Z_{i,t} = 0, \; \forall t \in [r], i, j \in [p],$$

$$\mathrm{Rank}(\boldsymbol{Y}^t) = 1, \; \forall t \in [r].$$

REMARK 1. We do not explicitly require $\boldsymbol{Y} \succeq \boldsymbol{0}$, as $\boldsymbol{Y}$ is the sum of positive semidefinite matrices.

REMARK 2. Problem (7) is not a convex mixed-integer semidefinite optimization problem because of the presence of rank constraints which are not mixed-integer convex representable as proven by Lubin et al. (2022). Indeed, the rank constraints are not redundant and cannot be dropped from the formulation (7) without altering its optimal value. We demonstrate this via a simple example in Section EC.10. This is notably different from the special cases of (7) reviewed in the introduction, where $\boldsymbol{Z}$ is restricted to have fully overlapping or fully disjoint supports and the rank constraints are redundant.

The proof of Theorem 1 requires an intermediate result (Proposition 1). Proposition 1 shows that by imposing rank-one constraints on each $\boldsymbol{Y}^t$, the condition that the $\boldsymbol{Y}^t$'s are mutually orthogonal can be reformulated as a linear semidefinite constraint (proof deferred to Section EC.1).

PROPOSITION 1. *Consider r matrices, $\boldsymbol{Y}^t \in \mathcal{S}_+^p$, such that $\mathrm{tr}(\boldsymbol{Y}^t) = 1$ and $\mathrm{Rank}(\boldsymbol{Y}^t) = 1$. Then, $\sum_{t \in [r]} \boldsymbol{Y}^t \preceq \mathbb{I}$ if and only if $\langle \boldsymbol{Y}^t, \boldsymbol{Y}^{t'} \rangle = 0 \ \forall t, t' \in [r]: \ t \neq t'$.*

Proof of Theorem 1 It suffices to show that for any feasible solution to (3), we can construct a feasible solution to Problem (7) with an equal or greater payoff, and vice versa.

- Let $(\boldsymbol{U}, \boldsymbol{Z})$ be a solution to Problem (3). Then, since $U_{i,t}$ can only be non-zero if $Z_{i,t} = 1$ and $\boldsymbol{U}^\top \boldsymbol{U} \preceq \mathbb{I}$, it follows that $\boldsymbol{U}\boldsymbol{U}^\top \preceq \mathrm{Diag}\left(\min\left(\boldsymbol{e}, \sum_t \boldsymbol{Z}_t\right)\right)$. Therefore, $(\boldsymbol{Y} := \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{Y}^t := \boldsymbol{U}_t \boldsymbol{U}_t^\top, \boldsymbol{Z})$ is a feasible solution to (7) with an equal cost.

- Let $(\boldsymbol{Y}, \boldsymbol{Y}^t, \boldsymbol{Z})$ denote a feasible solution to Problem (7). Then, since each $\boldsymbol{Y}^t$ is symmetric and rank-one, we can decompose $\boldsymbol{Y}^t$ as $\boldsymbol{Y}^t = \boldsymbol{U}_t \boldsymbol{U}_t^\top$ for a vector $\boldsymbol{U}_t$ such that $U_{i,t} = 0$ if $Z_{i,t} = 0$, and concatenate these vectors $\boldsymbol{U}_t$ into a matrix $\boldsymbol{U}$ such that $(\boldsymbol{U}, \boldsymbol{Z})$ has the same cost in (3) as $(\boldsymbol{Y}, \boldsymbol{Y}^t, \boldsymbol{Z})$ does in (7). Therefore, it remains to show that $\boldsymbol{U}^\top \boldsymbol{U} = \mathbb{I}$. To see this, observe that $\boldsymbol{Y} \preceq \mathbb{I}$ implies $(\boldsymbol{U}_t^\top \boldsymbol{U}_{t'})^2 = \langle \boldsymbol{Y}^t, \boldsymbol{Y}^{t'} \rangle = 0$ if $t \neq t'$ by Proposition 1. $\qquad \square$

Theorem 1 provides a formulation that is less compact than (3) but contains rank constraints rather than orthogonality constraints. Therefore, it is amenable to exact approaches for addressing sparsity (Bertsimas et al. 2021) and rank (Bertsimas et al. 2022a) constraints.

2.2. Semidefinite Relaxation With Global Sparsity

We leverage this mixed-integer low-rank reformulation of Problem (3) to derive a semidefinite relaxation. The valid inequalities we derive in this section rely on the presence of a global sparsity constraint $\langle \boldsymbol{E}, \boldsymbol{Z} \rangle \leq k$. In particular, per-component sparsity constraints $\left(\sum_{i \in [p]} Z_{i,t} \leq k_t\right)$ imply a global sparsity constraint $\langle \boldsymbol{E}, \boldsymbol{Z} \rangle \leq \sum_{t \in [r]} k_t$, so our results apply to both cases. However, tighter relaxations can be obtained when per-component sparsity is specified, as we investigate in the following section.

By relaxing the rank-one constraints on $\boldsymbol{Y}^t$ and the integrality constraints on $\boldsymbol{Z}$ in (7), we get a first semidefinite relaxation:

$$\max_{\substack{\boldsymbol{Z} \in [0,1]^{p \times r}: \\ \langle \boldsymbol{E}, \boldsymbol{Z} \rangle \leq k}} \max_{\substack{\boldsymbol{Y} \in \mathcal{S}^p, \boldsymbol{Y}^t \in \mathcal{S}_+^p, \\ \boldsymbol{w} \in [0,1]^p}} \langle \boldsymbol{Y}, \boldsymbol{\Sigma} \rangle \tag{8}$$

$$\text{s.t.} \quad \boldsymbol{Y} \preceq \mathrm{Diag}(\boldsymbol{w}), \boldsymbol{Y} = \sum_{t=1}^r \boldsymbol{Y}^t, \mathrm{tr}(\boldsymbol{Y}^t) = 1, \boldsymbol{w} \leq \boldsymbol{Z}\boldsymbol{e}$$

$$|Y_{i,j}^t| \leq M_{i,j} Z_{i,t} \qquad\qquad \forall i, j \in [p], t \in [r],$$

where the variable w_i models $\min(1, \sum_{t=1}^r Z_{i,t})$ and the M-constants are set to $M_{i,i} = 1$ and $M_{i,j} = 1/2$ if $i \neq j$—this is an upper bound on $|Y_{i,j}^t|$ because $\boldsymbol{Y}^t$ was, before relaxing the rank constraint, a rank one matrix (cf. Bertsimas et al. 2022b).

We propose to strengthen the relaxation (8) via the following valid inequalities:

THEOREM 2. *Consider a feasible solution to the mixed-integer low-rank problem* (3). *Then, the following inequalities hold:*

$$\sum_{j=1}^p (Y_{i,j}^t)^2 \leq Y_{i,i}^t Z_{i,t} \qquad\qquad \forall i \in [p], t \in [r],$$

$$\sum_{j=1}^p Y_{i,j}^2 \leq r Y_{i,i} w_i \qquad\qquad \forall i \in [p],$$

$$\left(\sum_{j=1}^p |Y_{i,j}| \right)^2 \leq k Y_{i,i} w_i \qquad\qquad \forall i \in [p],$$

$$\sum_{i \in [p]: i \neq j} Y_{i,j}^2 \leq (k - r + 1) w_j (w_j - Y_{j,j}), \qquad\qquad \forall j \in [p].$$

Note that these constraints are rotated second-order cone constraints (see, e.g., Alizadeh and Goldfarb 2003).

The proof of Theorem 2 is detailed in Section EC.2. We remark that, out of the four groups of valid inequalities, the first inequality has been previously stated in the case of sparse PCA with one component by Bertsimas and Cory-Wright (2020), Bertsimas et al. (2022b), Li and Xie (2025), but the three other groups of valid inequalities are, to our knowledge, new.

Based on Theorem 2, we obtain the following semidefinite relaxation:

$$\max_{\substack{\boldsymbol{Z} \in [0,1]^{p \times r}: \\ \langle \boldsymbol{E}, \boldsymbol{Z} \rangle \leq k}} \max_{\substack{\boldsymbol{Y} \in \mathcal{S}^p, \boldsymbol{Y}^t \in \mathcal{S}_+^p, \\ \boldsymbol{w} \in [0,1]^p}} \langle \boldsymbol{Y}, \boldsymbol{\Sigma} \rangle \tag{9}$$

$$\text{s.t.} \quad \boldsymbol{Y} \preceq \mathrm{Diag}(\boldsymbol{w}), \boldsymbol{Y} = \sum_{t=1}^r \boldsymbol{Y}^t, \mathrm{tr}(\boldsymbol{Y}^t) = 1, \boldsymbol{w} \leq \boldsymbol{Z}\boldsymbol{e}$$

$$|Y_{i,j}^t| \leq M_{i,j} Z_{i,t} \qquad\qquad \forall i, j \in [p], t \in [r],$$

$$\sum_{j=1}^p {Y_{i,j}^t}^2 \leq Y_{i,i}^t Z_{i,t} \qquad\qquad \forall i \in [p], t \in [r],$$

$$\sum_{j=1}^{p} Y_{i,j}^2 \leq r Y_{i,i} w_i \qquad\qquad \forall i \in [p],$$

$$\left(\sum_{j=1}^{p} |Y_{i,j}|\right)^2 \leq k Y_{i,i} w_i \qquad\qquad \forall i \in [p],$$

$$\sum_{i \in [p]: i \neq j} Y_{i,j}^2 \leq (k - r + 1) w_j (w_j - Y_{j,j}), \qquad\qquad \forall j \in [p],$$

2.3. Stronger Relaxation with a Per-Component Sparsity Budget

In this section, we further strengthen our relaxation when specifying a sparsity budget k_t for each component $\boldsymbol{Y}^t$. This can be understood as allocating the total sparsity on $\boldsymbol{U}$ between the different columns of $\boldsymbol{U}$ via $\boldsymbol{Y}_t$, which models the outer product $\boldsymbol{U}_t \boldsymbol{U}_t^\top$.

Formally, we have the following result (proof deferred to Section EC.3):

PROPOSITION 2. *Suppose that $\sum_{i \in [p]} \boldsymbol{Z}_{i,t} \leq k_t$ in Problem* (7). *Then, the following inequalities hold:*

$$\left(\sum_{j=1}^{p} |Y_{i,j}^t|\right)^2 \leq k_t Y_{i,i}^t Z_{i,t} \quad \forall i \in [p], \forall t \in [r], \tag{10}$$

$$\sum_{i \in [p]: i \neq j} (Y_{i,j}^t)^2 \leq (k_t - 1) Z_{j,t}(Z_{j,t} - Y_{j,j}^t) \quad \forall j \in [p]. \tag{11}$$

Interestingly, as we observe in Section 5.1, combining Problem (9) with constraints (10)-(11) often yields much tighter upper bounds than (9) alone, even if we take the worst-case upper bound over all feasible splits $\{k_t\}_{t \in [r]}$ which sum to k.

Now that we have introduced the sparsity of each PC, k_t, we can further tighten our semidefinite relaxations by leveraging valid inequalities obtained in the case of a single PC. For example, for each component $t \in [r]$, Kim et al. (2022) observe that the feasible set of all k_t-sparse components $\{\boldsymbol{u} \in \mathbb{R}^p : \|\boldsymbol{u}\|_0 \leq k_t\}$ is permutation and sign invariant, i.e., for any feasible vector $\boldsymbol{u}$, any vector obtained by permuting or changing the sign of the coordinates of $\boldsymbol{u}$ is also feasible. Based on this observation, they propose a lifted formulation for sparse PCA with a single PC, which, to the best of our knowledge, leads to the strongest known relaxation for sparse PCA with $r = 1$ which can be solved in polynomial time. For the sake of concision, we denote $(\boldsymbol{Y}^t, \boldsymbol{Z}_t) \in \mathcal{T}(k_t)$ the set of valid inequalities comprised in their "T-relaxation" (see Problem (EC.5) in Section EC.4.1 for an explicit formulation) and consider the following relaxation:

$$\max_{\substack{\boldsymbol{Z} \in [0,1]^{p \times r}: \boldsymbol{Y} \in \mathcal{S}_+^p, \boldsymbol{Y}^t \in \mathcal{S}_+^p \\ \langle \boldsymbol{E}, \boldsymbol{Z} \rangle \leq k, \\ \boldsymbol{w} \in [0,1]^p}} \quad \langle \boldsymbol{Y}, \boldsymbol{\Sigma} \rangle \tag{12}$$

$$\text{s.t.} \quad \boldsymbol{Y} \preceq \mathrm{Diag}(\boldsymbol{w}), \ \boldsymbol{Y} = \sum_{t=1}^{r} \boldsymbol{Y}^t, \ \mathrm{tr}(\boldsymbol{Y}^t) = 1, \ \boldsymbol{w} \leq \boldsymbol{Z} \boldsymbol{e},$$

$$\sum_{j=1}^{p} Y_{i,j}^2 \le r Y_{i,i} w_i \qquad\qquad \forall i \in [p],$$

$$\left(\sum_{j=1}^{p} |Y_{i,j}| \right)^2 \le k Y_{i,i} w_i \qquad\qquad \forall i \in [p],$$

$$\sum_{i \in [p]: i \ne j} Y_{i,j}^2 \le (k - r + 1) w_j (w_j - Y_{j,j}) \qquad\qquad \forall j \in [p],$$

$$(\boldsymbol{Y}^t, \boldsymbol{Z}_t) \in \mathcal{T}(k_t) \qquad\qquad \forall t \in [r].$$

REMARK 3. The relaxation of Problem (12) dominates that of Problem (9), even when (9) is strengthened with the inequalities proposed in (10)-(11). Indeed, Kim et al. (2022, Theorem 13) can be extended to show that (10)-(11) are redundant in (12).

2.4. Feasible Solutions via Greedy Disjoint Rounding

We develop a rounding mechanism that converts an optimal solution to our semidefinite convex relaxations into a high-quality feasible solution. Historically, a useful strategy for similar integer optimization problems has been to (a) solve a convex relaxation in $(\boldsymbol{Z}, \boldsymbol{Y})$, ($b$) greedily round $\boldsymbol{Z}^\star$, the solution to the relaxation, to obtain a feasible binary matrix $\hat{\boldsymbol{Z}}$ that is close to $\boldsymbol{Z}^\star$, and (c) resolve for $\boldsymbol{U}$ under the constraints $U_{i,t} = 0$ if $\hat{Z}_{i,t} = 0$ (cf. Bertsimas et al. 2022b).

Unlike in the case with a single PC, observe that the "resolve" step (c) is non-trivial. Namely, solving for $\boldsymbol{U}$ for some arbitrary and fixed sparsity pattern $\hat{\boldsymbol{Z}}$, i.e., solving

$$\max_{\boldsymbol{U} \in \mathbb{R}^{p \times r}} \quad \langle \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{\Sigma} \rangle \ \text{s.t.} \ \boldsymbol{U}^\top \boldsymbol{U} = \mathbb{I}, \ U_{i,t} = 0 \ \text{if} \ \hat{Z}_{i,t} = 0, \ \forall i \in [p], t \in [r], \qquad (13)$$

cannot be done in closed form in general. Actually, we theoretically characterize the worst-case complexity of Problem (13).

THEOREM 3. *Problem* (13) *is NP-hard.*

The proof of Theorem 3 (deferred to Section EC.5) relies on a reduction from a variant of the exact cover by 3-sets problem, which is known to be NP-complete (Garey and Johnson 1979, Chapter A3.1). The hard instances generated by this reduction have $r \ge p/2$, $k_t = 4$, and neither fully overlapping nor fully disjoint supports. Indeed, when $\hat{\boldsymbol{Z}}$ corresponds to fully overlapping or fully disjoint supports, we can obtain the solution of (13) in polynomial time via an eigenvalue decomposition of the corresponding submatrix/submatrices.

Therefore, we propose in Algorithm 1 a relax-round-and-resolve strategy where the rounding step (b) generates a solution with fully disjoint supports, i.e., where $\sum_{t \in [r]} \hat{Z}_{i,t} \le 1, \ \forall i \in [p]$.

The rounding step searches for the binary disjoint support vector $\boldsymbol{Z}$ that is aligned with the solution of the relaxation as much as possible. Note that maximizing the alignment between the

rounded and the relaxed solution, $\langle \boldsymbol{Z}, \boldsymbol{Z}^{\star} \rangle$, is equivalent to minimizing the distance between the two since $\|\boldsymbol{Z} - \boldsymbol{Z}^{\star}\|^2 = k - 2\langle \boldsymbol{Z}, \boldsymbol{Z}^{\star} \rangle + \|\boldsymbol{Z}^{\star}\|^2$.

Since $\hat{\boldsymbol{Z}}$ encodes for disjoint supports, any matrix $\boldsymbol{U}$ with support $\hat{\boldsymbol{Z}}$ automatically satisfies the orthogonality constraint. We can obtain $\boldsymbol{U}$ solution of (13) by solving for each PC independently. For each $t \in [r]$, we consider the submatrix of $\boldsymbol{\Sigma}$ over the indices $\{i : \hat{Z}_{i,t} = 1\}$, extract its leading eigenvector via SVD, and pad it with zeros to construct $\boldsymbol{U}_t$. Although the restriction to disjoint support is not without loss of optimality, we will observe numerically in Section 5 that disjoint solutions are not particularly suboptimal for Problem (3) when k and r are small relative to p—an observation already made by Asteris et al. (2015).

Algorithm 1 A disjoint greedy rounding method of the semidefinite relaxation

---

Require: Covariance matrix $\boldsymbol{\Sigma}$, rank parameter r, sparsity parameter k or k_t

Compute UB the objective value of (9) or (12)

Compute $\boldsymbol{Z}^{\star}$ solution of (9) or (12) with the constraint $\sum_{t=1}^{r} Z_{i,t} \leq 1, \ \forall i \in [p]$

Construct $\hat{\boldsymbol{Z}} \in \{0,1\}^{p \times r}$ solution of

$$\max_{\boldsymbol{Z} \in \{0,1\}^{p \times r}} \langle \boldsymbol{Z}, \boldsymbol{Z}^{\star} \rangle \text{ s.t. } \sum_{i=1}^{p} Z_{i,t} \geq 1, \ \forall t \in [r],$$
$$\sum_{i \in [p], t \in [r]} Z_{i,t} \leq k \text{ or } \sum_{i \in [p]} Z_{i,t} \leq k_t, \ \forall t \in [r],$$
$$\sum_{t=1}^{r} Z_{i,t} \leq 1, \ \forall i \in [p].$$

Compute $\boldsymbol{U}$ solution of (13) via SVD

return $UB, \hat{\boldsymbol{Z}}, \boldsymbol{U}$.

---

3. Lagrangian Relaxation Approach

In this section, we investigate an approach for obtaining upper and lower bounds based on the theory of penalty methods and Lagrangian relaxations (e.g., Fisher 1981, Bertsekas 1996, Nocedal and Wright 2006, Chapter 17), which argues that if a non-convex problem is decomposable as a sum of easier (but still non-convex) subproblems with a coupling constraint, a good strategy is often to penalize the coupling constraint in the objective and iteratively solve the non-convex subproblems with different multipliers on the coupling constraint. For example, Lu and Zhang (2012) proposes an augmented Lagrangian method for solving an ℓ_1 relaxation of the sparse PCA, where the sparsity constraints are replaced by an ℓ_1 penalty in the objective.

Unlike the other two approaches we investigate, the Lagrangian only applies to the case where a separate sparsity budget k_t is imposed on each PC in Problem (3).

3.1. Lagrangian Relaxation and Upper Bound

We consider the case where a separate sparsity budget k_t is imposed on each PC so that the orthogonality constraints are the only coupling constraints. Namely we consider

$$\max_{\boldsymbol{U}_t \in \mathbb{R}^p, t \in [r]} \quad \sum_{t \in [r]} \langle \boldsymbol{U}_t \boldsymbol{U}_t^\top, \boldsymbol{\Sigma} \rangle \quad \text{s.t.} \quad \langle \boldsymbol{U}_t, \boldsymbol{U}_{t'} \rangle = \delta_{t,t'}, \ \forall t, t' \in [r],$$
$$\|\boldsymbol{U}_t\|_0 \le k_t, \ \forall t \in [r]. \tag{14}$$

We introduce a penalty parameter $\lambda_{t,t'} > 0$ for each orthogonality constraint $\langle \boldsymbol{U}_t, \boldsymbol{U}_{t'} \rangle = 0$ for $t \neq t'$. Observing that $(\langle \boldsymbol{U}_t, \boldsymbol{U}_{t'} \rangle)^2 = \langle \boldsymbol{U}_t \boldsymbol{U}_t^\top, \boldsymbol{U}_{t'} \boldsymbol{U}_{t'}^\top \rangle$, we obtain the following relaxation:

$$\max_{\boldsymbol{U}_t \in \mathbb{R}^p, t \in [r]} \quad \sum_{t \in [r]} \langle \boldsymbol{U}_t \boldsymbol{U}_t^\top, \boldsymbol{\Sigma} \rangle - \sum_{t,t' \in [r]: t \neq t'} \lambda_{t,t'} \langle \boldsymbol{U}_t \boldsymbol{U}_t^\top, \boldsymbol{U}_{t'} \boldsymbol{U}_{t'}^\top \rangle \quad \text{s.t.} \quad \langle \boldsymbol{U}_t, \boldsymbol{U}_t \rangle = 1, \ \forall t \in [r],$$
$$\|\boldsymbol{U}_t\|_0 \le k_t, \ \forall t \in [r]. \tag{15}$$

Note that the penalty in the objective of Problem (15) can be interpreted as a quadratic penalty on the orthogonality constraints $\langle \boldsymbol{U}_t, \boldsymbol{U}_{t'} \rangle = 0, t \neq t'$ or, equivalently, as the Lagrangian of (14) where the orthogonality constraints are reformulated as $\langle \boldsymbol{U}_t, \boldsymbol{U}_{t'} \rangle^2 = 0, t \neq t'$. For a fixed value of the parameters $\boldsymbol{\lambda}$, Problem (15) provides an upper bound on the objective value of (14) by the Lagrangian duality theorem. Optimizing for all PCs $\boldsymbol{U}_t$ simultaneously in Problem (15) is challenging due to the non-convex objective. For a given index t, however, optimizing for $\boldsymbol{U}_t$ (with all other $\boldsymbol{U}_{t'}$, $t' \neq t$, and $\boldsymbol{\lambda}$ fixed) is equivalent to finding the leading k_t-sparse PC of the matrix $\boldsymbol{\Sigma} - \sum_{t' \neq t} \lambda_{t,t'} \boldsymbol{U}_{t'} \boldsymbol{U}_{t'}^\top$, for which efficient algorithms have been developed recently (see Section 1.2)[1]. We use this recent technology to derive an upper bound on (14). Namely, for $\lambda = 0$, we solve (15) by solving r single-PC sparse PCA problems, or equivalently

$$\max_{\boldsymbol{U} \in \mathbb{R}^{p \times r}} \quad \sum_{t \in [r]} \boldsymbol{U}_t^\top \boldsymbol{\Sigma} \boldsymbol{U}_t \quad \text{s.t.} \quad \|\boldsymbol{U}_t\|_2 = 1, \ \|\boldsymbol{U}_t\|_0 \le k_t, \forall t \in [r] \tag{16}$$

using a certifiably optimal method (Berk and Bertsimas 2019, in our implementation). Note that $\boldsymbol{U}_t$ denotes the tth column of $\boldsymbol{U}$ and the overall objective value provides an upper bound on (14). Actually, (16) provides an r-factor approximation of (14), as formally stated below.

PROPOSITION 3. *The objective value of* (16) *is at most r times the objective value of the original sparse PCA problem* (14).

Proof of Proposition 3 We already know that (16) provides an upper bound on (14), (14) $\le$ (16) in short. Furthermore, (16) is bounded above by

$$r \times \max_{t \in [r]} \left\{ \max_{\boldsymbol{u} \in \mathbb{R}^p} \quad \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} \quad \text{s.t.} \quad \|\boldsymbol{u}\|_2 = 1, \ \|\boldsymbol{u}\|_0 \le k_t \right\} \quad \le r \times (14). \qquad \square$$

3.2. A Heuristic Based on Iterative Deflation

When solving (15) for a fixed $\boldsymbol{\lambda}$, the returned PCs may not be feasible for the original Problem (3) because they can violate the orthogonality constraints. A strategy to obtain a feasible solution is

thus to solve a sequence of problems of the form (15) with an increasing penalty parameter value, as we propose in Algorithm 2.

At each iteration, Algorithm 2 optimizes for each $\boldsymbol{U}_t$ in (15) sequentially (instead of simultaneously). Consequently, Algorithm 2 is not guaranteed to converge to a near-optimal solution. However, when the updates of each column $\boldsymbol{U}_t$ are ϵ-optimal (e.g., when using the methods of Berk and Bertsimas 2019, Bertsimas et al. 2022b), we show in Proposition EC.3 that the limiting solutions are feasible. For tractability considerations, we use the truncated power method of Yuan and Zhang (2013) (instead of exact methods) for which we cannot guarantee ϵ-optimality. However, we can still show convergence to some stationary solutions (see Proposition EC.4). Unfortunately, however, we cannot rule out convergence to infeasible solutions, which is a known limitation of the quadratic penalty method (see Nocedal and Wright 2006, Theorem 17.2 and discussion therein). In practice, we guarantee η-feasibility of the returned solution by recording all η-feasible solutions generated and returning the best one.

Algorithm 2 Lagrangian Alternating Maximization for Problem (3)

Require: Matrix $\boldsymbol{\Sigma}$, rank parameter r, sparsity parameters $k_1, \ldots, k_r$, number of iterations L, feasibility tolerance η

Compute UB the optimal value of (16)

Initialize $\boldsymbol{U}_t^{(0)} \leftarrow \boldsymbol{0}$

for $\ell = 1, \ldots, L$ **do**

    for $t = 1, \ldots, r$ **do**

        Compute $\boldsymbol{U}_t^{(\ell)}$ an approximate solution of

$$\max_{\boldsymbol{u} \in \mathbb{R}^p} \ \left\langle \boldsymbol{u}\boldsymbol{u}^\top, \boldsymbol{\Sigma} - \sum_{t' \in [r] : t' \neq t} \lambda_{t,t'} \boldsymbol{U}_{t'}^{(\ell-1)} \boldsymbol{U}_{t'}^{(\ell-1)\top} \right\rangle \text{ s.t. } \|\boldsymbol{u}\|_2 = 1, \ \|\boldsymbol{u}\|_0 \leq k_t.$$

    end for

    if $\sum_{t,t'} |\langle \boldsymbol{U}_t^{(\ell)}, \boldsymbol{U}_{t'}^{(\ell)} \rangle - \delta_{t,t'}| \leq \eta$ **then**

        Consider $\left\{ \boldsymbol{U}_t^{(\ell)}, t \in [r] \right\}$ feasible and record its objective value

    end if

    Update $\boldsymbol{\lambda}$ value.

end for

return UB and the best feasible solution $\{\boldsymbol{U}_t, t \in [r]\}$ found

---

In our block computations (column-wise) of $\boldsymbol{U}^{(\ell)}$, we use the previous values of $\boldsymbol{U}_{t'}^{(\ell-1)}$, even for the columns $t' < t$, which have already been updated (this is the so-called block-Jacobi update rule).

Instead, for memory efficiency, one could use $\boldsymbol{U}_{t'}^{(\ell)}$ for $t' < t$ and $\boldsymbol{U}_{t'}^{(\ell-1)}$ for $t' > t$ (known as Gauss-Seidel updates; see Remark EC.3). In our implementation, we consider a feasibility tolerance (e.g., 10^{-4}) and return the best solution among those satisfying the orthogonality constraints up to this tolerance. Note that our definition of constraint violation, $\sum_{t,t'} |\langle \boldsymbol{U}_t, \boldsymbol{U}_{t'} \rangle - \delta_{t,t'}|$, includes the constraints $\|\boldsymbol{U}_t\|_2 = 1$, so that the initialization (all-zero solution) is correctly considered as infeasible. For the termination criterion, we set a limit on the total number of iterations. We can also terminate the algorithm when it stalls, i.e., when the difference in objective value between two consecutive feasible solutions falls below a certain threshold ϵ, or impose a time limit. Regarding the update of the penalty parameter $\boldsymbol{\lambda}$, we increase it progressively as is standard in the Lagrangian relaxation literature (Fisher 1981, Bertsekas 1996, Nocedal and Wright 2006, Chapter 17). Typically, one can take increments proportional to the gradient of the Lagrangian with respect to $\boldsymbol{\lambda}$, $\langle \boldsymbol{U}_t, \boldsymbol{U}_{t'} \rangle^2$, or as a scaling factor between the two concurrent objectives, $\sum_{t \in [r]} \langle \boldsymbol{U}_t \boldsymbol{U}_t^\top, \boldsymbol{\Sigma} \rangle \Big/ \sum_{t,t' \in [r] : t \neq t'} \langle \boldsymbol{U}_t, \boldsymbol{U}_{t'} \rangle^2$. In our implementation, we use the former increment (i.e., gradient-based) during the first iterations, and the latter (which is typically larger) at later stages, the intuition being that we slowly increase $\boldsymbol{\lambda}$ in the beginning to enable exploration and switch to being more aggressive at later iterations to ensure that we are generating feasible solutions. Precisely, we use parameters of the form $\lambda_{t,t'} = w_t \tilde{\lambda}$, where w_t is equal to the objective value obtained by the PC $\boldsymbol{U}_t$ found at the first iteration ($\ell = 1$), and update $\tilde{\lambda}$ according to the following rule

$$\text{initialization:} \quad \tilde{\lambda} = 0,$$

$$\text{iteration } \ell = 1, \ldots, \lceil 0.15L \rceil - 1 \quad \tilde{\lambda} \leftarrow \tilde{\lambda} + \alpha \times \sum_{t \neq t'} \langle \boldsymbol{U}_t, \boldsymbol{U}_{t'} \rangle^2,$$

$$\text{iteration } \ell = \lceil 0.15L \rceil, \ldots, L \quad \tilde{\lambda} \leftarrow \tilde{\lambda} + \alpha \times \sum_{t \in [r]} \langle \boldsymbol{U}_t \boldsymbol{U}_t^\top, \boldsymbol{\Sigma} \rangle \Big/ \sum_{t,t' \in [r] : t \neq t'} \langle \boldsymbol{U}_t, \boldsymbol{U}_{t'} \rangle^2,$$

with α a fixed stepsize parameter.

4. Combinatorial Approach

The Gershgorin circle theorem (cf. Horn and Johnson 1985, Chapter 6) bounds the largest eigenvalue of a matrix $\boldsymbol{\Sigma} \in \mathcal{S}_+^p$ via the combinatorial function:

$$\lambda_{\max}(\boldsymbol{\Sigma}) \leq \max_{i \in [p]} \sum_{j \in [p]} |\Sigma_{i,j}|. \tag{17}$$

For sparse PCA with a single PC, several authors (Berk and Bertsimas 2019, Bertsimas et al. 2022b) have leveraged this result to derive an upper bound on (2) that depends on the support of the sparse PC. Motivated by their observations, we now bound the objective value of (3) as a function of the support matrix $\boldsymbol{Z}$, and leverage this bound to generate upper bounds and feasible solutions to Problem (3).

4.1. Generalizing Gershgorin Circle Theorem

One naive upper bound is to apply the Gershgorin circle theorem to each PC separately and bound the objective of (3) via the sum of the largest eigenvalues of each sub-matrix of $\boldsymbol{\Sigma}$ induced by a column of $\boldsymbol{Z}$. However, this approach is too conservative for non-disjoint PCs, since it does not take into account any information about the overlap between the support of each PC. Indeed, with fully overlapping support, this approach bounds the sum of the r largest eigenvalues of the relevant submatrix by r times its largest eigenvalue. Instead, the valid inequalities we derive in this section rely on a new and non-trivial bound on the variance collectively explained by r orthogonal PCs. Incidentally, our result leads to the following bound on the sum of the r largest eigenvalues of a semidefinite matrix $\boldsymbol{\Sigma}$:

$$\sum_{t\in[r]}\lambda_t(\boldsymbol{\Sigma}) \leq \max_{\boldsymbol{\mu}\in\{0,1\}^p:\boldsymbol{e}^\top\boldsymbol{\mu}\leq r}\sum_{i,j\in[p]}\mu_i|\Sigma_{i,j}|, \tag{18}$$

which strictly generalizes (17) and could be of independent interest.[2]

Formally, we derive the following generalization of the circle theorem bound, which holds with multiple PCs and a fixed but arbitrary support pattern $\boldsymbol{Z}\in\{0,1\}^{p\times r}$ satisfying $\sum_{i\in[p]}Z_{i,t}\geq 1\ \forall t\in[r]$. Subsequently, we derive a mixed-integer linear representation of this bound:

THEOREM 4. *For any feasible support pattern $\boldsymbol{Z}\in\{0,1\}^{p\times r}:\sum_{i\in[p]}Z_{i,t}\geq 1\ \forall t\in[r]$, an upper bound on the objective value attained by any matrix $\boldsymbol{U}$ such that $U_{i,t}=0$ if $Z_{i,t}=0\ \forall i\in[p],\ \forall t\in[r]$ in Problem (3) is given by:*

$$\max_{\substack{\boldsymbol{\mu}\in\{0,1\}^{p\times r}:\\ \sum_{i\in[p]}\mu_{i,t}=1,\forall t\in[r],\\ \sum_{t\in[r]}\mu_{i,t}\leq 1\forall i\in[p]}}\sum_{i,j\in[p]}\sum_{t\in[r]}\mu_{i,t}Z_{i,t}Z_{j,t}|\Sigma_{i,j}|. \tag{19}$$

REMARK 4. Taking $\boldsymbol{Z}=\boldsymbol{E}$ in Theorem 4 yields (18). If $r=1$, this bound is equivalent to the circle theorem, and if $r=p$ it is equivalent to the (known) fact that $\sum_{t\in[p]}\lambda_t(\boldsymbol{\Sigma})=\operatorname{tr}(\boldsymbol{\Sigma})\leq\sum_{i,j\in[p]}|\Sigma_{i,j}|$. More generally, it shows that the eigenvalues of a positive semidefinite matrix are majorized by the absolute column sums (see Marshall and Olkin 1979, for a general theory of majorization).

REMARK 5. If $\boldsymbol{\Sigma}$ is a diagonally dominant matrix, i.e., $\Sigma_{i,i}\geq\sum_{j\neq i}|\Sigma_{i,j}|\ \forall i$, then for any (i,t) with $Z_{i,t}=1$, $\sum_{j\in[p]}Z_{j,t}|\Sigma_{i,j}|=\Sigma_{i,i}+\sum_{j\neq i}Z_{j,t}|\Sigma_{i,j}|\leq 2\Sigma_{i,i}$. Hence, (19) is lower than the maximum of $2\times\sum_t\Sigma_{i_t,i_t}$ over distinct indices i_t such that $Z_{i_t,t}=1$. Observe that $\sum_t\Sigma_{i_t,i_t}=\langle\boldsymbol{U}\boldsymbol{U}^\top,\boldsymbol{\Sigma}\rangle$ for the matrix $\boldsymbol{U}$ whose columns are the canonical basis vectors $\boldsymbol{e}_{i_t}$. Consequently, (19) provides a 2-factor approximation of the optimal objective value in this case.

REMARK 6. For a fixed $\boldsymbol{Z}$, Problem (19) is a matching problem, where each PC $t\in[r]$ has to be matched to exactly one coordinate $i\in[p]$, and each coordinate can be matched to at most one PC. The Hopcroft-Karp algorithm (Hopcroft and Karp 1973) solves it in $O(rp\sqrt{r+p})$ time.

When the supports of each PC are disjoint, (19)'s upper bound is equivalent to applying the circle theorem to each PC separately. Alternatively, with fully overlapping support, it reduces to bounding the variance explained by the largest r column sums of the submatrix selected by $\boldsymbol{Z}$. In the case with partially overlapping support, it systematically interpolates between these bounds.

Proof of Theorem 4 Fix $\boldsymbol{Z} \in \{0,1\}^{p \times r}$ in Problem (3). Then, it follows directly from Theorem 1 that an upper bound on the objective value attained by any orthogonal matrix $\boldsymbol{U}$ such that $U_{i,t} = 0$ if $Z_{i,t} = 0$ is given by the following maximization problem:

$$\max_{\boldsymbol{Y}^t \in \mathcal{S}_+^p \ \forall t \in [r]} \sum_{t \in [r]} \langle \boldsymbol{Y}^t, \boldsymbol{\Sigma} \rangle \ \text{s.t.} \ \mathrm{tr}(\boldsymbol{Y}^t) = 1 \ \forall t \in [r], \sum_{t \in [r]} \boldsymbol{Y}^t \preceq \mathbb{I}, Y_{i,j}^t = 0 \ \text{if} \ Z_{i,t} = 0 \ \forall i \in [p], t \in [r].$$

To obtain a non-trivial mixed-integer linear representable upper bound as a function of the support pattern $\boldsymbol{Z}$, we now relax this problem. First, we observe that $Y_{i,j}^t = 0$ if $Z_{i,t} = 0$ and therefore we can replace $\boldsymbol{\Sigma}$ in the objective with $\mathrm{Diag}(\boldsymbol{Z}_t)\boldsymbol{\Sigma}\mathrm{Diag}(\boldsymbol{Z}_t)$ without loss of generality, where $\mathrm{Diag}(\boldsymbol{Z}_t)$ is a diagonal matrix with on-diagonal entries specified by the tth column of $\boldsymbol{Z}$. Further relaxing the problem by omitting the logical constraints then gives the following semidefinite upper bound:

$$\max_{\boldsymbol{Y}^t \in \mathcal{S}_+^p \ \forall t \in [r]} \sum_{t \in [r]} \langle \boldsymbol{Y}^t, \mathrm{Diag}(\boldsymbol{Z}_t)\boldsymbol{\Sigma}\mathrm{Diag}(\boldsymbol{Z}_t) \rangle \ \text{s.t.} \ \mathrm{tr}(\boldsymbol{Y}^t) = 1 \ \forall t \in [r], \sum_{t \in [r]} \boldsymbol{Y}^t \preceq \mathbb{I}.$$

Moreover, strong duality holds between this problem and its dual problem, namely:

$$\min_{\boldsymbol{U} \in \mathcal{S}_+^p, \boldsymbol{s} \in \mathbb{R}^r} \ \mathrm{tr}(\boldsymbol{U}) + \boldsymbol{e}^\top \boldsymbol{s} \quad \text{s.t.} \quad \boldsymbol{U} + s_t \mathbb{I} \succeq \mathrm{Diag}(\boldsymbol{Z}_t)\boldsymbol{\Sigma}\mathrm{Diag}(\boldsymbol{Z}_t) \ \forall t \in [r].$$

To obtain a linear, rather than semidefinite, upper bound from this problem, we restrict $\boldsymbol{U}$ to be a diagonal matrix and $\boldsymbol{U} + s_t \mathbb{I} - \mathrm{Diag}(\boldsymbol{Z}_t)\boldsymbol{\Sigma}\mathrm{Diag}(\boldsymbol{Z}_t)$ to be contained within the cone of diagonally dominant matrices, which is an inner approximation of the positive semidefinite cone (see also Barker and Carlson 1975, Ahmadi et al. 2017, for detailed studies of this inner approximation). This gives the following upper bound:

$$\min_{\boldsymbol{u} \in \mathbb{R}_+^p, \boldsymbol{s}} \ \boldsymbol{e}^\top \boldsymbol{u} + \boldsymbol{e}^\top \boldsymbol{s} \ \text{s.t.} \ u_i + s_t \geq \sum_{j \in [p]} Z_{i,t} Z_{j,t} |\Sigma_{i,j}| \ \forall i \in [p], \forall t \in [r].$$

Finally, we invoke strong duality and use the fact that some (binary) extreme point in the dual problem must be dual-optimal, to verify that the above problem attains the same value as:

$$\max_{\substack{\boldsymbol{\mu} \in \{0,1\}^{p \times r}: \\ \sum_{i \in [p]} \mu_{i,t} = 1 \ \forall t \in [r], \\ \sum_{t \in [r]} \mu_{i,t} \leq 1 \ \forall i \in [p]}} \sum_{i,j \in [p]} \sum_{t \in [r]} \mu_{i,t} Z_{i,t} Z_{j,t} |\Sigma_{i,j}|. \qquad \square$$

Observe that the proof of Theorem 4 involves invoking strong duality and taking a finitely generated inner approximation of the positive semidefinite cone. This is quite different from existing proofs of the Gershgorin circle theorem, which usually leverage properties of eigenvectors and therefore cannot easily be generalized.[3] Thus, our proof technique could also be useful in other contexts, e.g., in sparse canonical correlation analysis (Witten et al. 2009).

4.2. A Mixed-Integer Linear Relaxation

Since Theorem 4 provides an upper bound on the objective value (i.e., the fraction of variance explained) for a given support $\boldsymbol{Z}$, we can obtain an upper bound on (7) by optimizing (19) over all possible supports, $\boldsymbol{Z}\{0,1\}^{p \times r}$:

$$\max_{\boldsymbol{Z}\{0,1\}^{p \times r}} (19) \quad \text{s.t.} \quad \sum_{i,t} Z_{i,t} \leq k \text{ or } \sum_i Z_{i,t} \leq k_t, \forall t \in [r], \tag{20}$$

depending on the nature of the sparsity budget (global or per-component). Let us observe that Problem (20) can be reformulated as a mixed-integer linear optimization problem by introducing auxiliary variables $\rho_{i,t} \; \forall i \in [p], t \in [r]$ to model the column sum $\sum_{j \in [p]} Z_{j,t}|\Sigma_{i,j}|$ if $\mu_{i,t} = 1$ and equal 0 if $\mu_{i,t} = 0$ in (19). This allows us to represent Theorem 4's upper bound via the system:

$$
\begin{aligned}
\theta &= \sum_{i,t} \rho_{i,t}, \\
\rho_{i,t} &= \begin{cases} \sum_{j \in [p]} Z_{j,t}|\Sigma_{i,j}| & \text{if } \mu_{i,t} = 1 \\ 0 & \text{if } \mu_{i,t} = 0 \end{cases} \quad \forall i \in [p], t \in [r], \\
\sum_{i \in [p]} \mu_{i,t} &= 1 \qquad\qquad\qquad\qquad\quad \forall t \in [r], \\
\sum_{t \in [r]} \mu_{i,t} &\leq 1 \qquad\qquad\qquad\qquad\quad \forall i \in [p], \\
\mu_{i,t} &\leq Z_{i,t} \qquad\qquad\qquad\qquad\; \forall i \in [p], t \in [r], \\
\mu_{i,t} &\in \{0,1\} \qquad\qquad\qquad\qquad \forall i \in [p], t \in [r],
\end{aligned}
\tag{21}
$$

in the variables $(\theta, \boldsymbol{\mu}, \boldsymbol{\rho})$. We omit the term $Z_{i,t}$ from the partial sum $\sum_{j \in [p]} Z_{j,t}|\Sigma_{i,j}|$ by imposing the constraint $\mu_{i,t} \leq Z_{i,t}$ because if $Z_{i,t} = 0$ then the (i,t)th column sum is zero and can be omitted from the bound without loss of generality.

4.3. Feasible Solution with Disjoint Supports

Unfortunately, as explained in Section 2.4, the solution to the mixed-integer linear relaxation (20), $\boldsymbol{Z}^\star \in \{0,1\}^{p \times r}$, cannot be used to generate a feasible solution $\boldsymbol{U}$ directly, because estimating the optimal $\boldsymbol{U}$ for a given $\boldsymbol{Z}$ in non-trivial. To alleviate this issue, as in Algorithm 1, we restrict our attention to feasible solutions with disjoint supports.

In Algorithm 3, we first solve Problem (20), with the additional constraint that $\boldsymbol{Z}$ encodes for disjoint supports, and estimate the optimal PCs with sparsity pattern $\hat{\boldsymbol{Z}}$. This gives both a feasible solution $\boldsymbol{U}$ and a high-quality upper bound, after solving two MIOs and an eigenproblem.

5. Numerical Results

In this section, we evaluate the algorithmic strategies derived in the previous three sections for solving Problem (3), implemented in `Julia` 1.9 using `JuMP.jl` 1.12.0, `Gurobi` version 10.0.0 to solve

---

Algorithm 3 A disjoint-support solution from the combinatorial upper bound rounding (19)

Require: Covariance matrix $\boldsymbol{\Sigma}$, rank parameter r, sparsity parameter k

Compute UB the optimal value of (20)

Construct $\hat{\boldsymbol{Z}} \in \{0,1\}^{p \times r}$ solution of

$$\max_{\boldsymbol{Z} \in \{0,1\}^{p \times r}} \theta \text{ s.t. } \sum_{i,t} Z_{i,t} \leq k \text{ or } \sum_i Z_{i,t} \leq k_t, \forall t \in [r],$$

$$\sum_{t=1}^r Z_{i,t} \leq 1, \ \forall i \in [p],$$

$$\theta \text{ satisfying (21)}.$$

Compute $\boldsymbol{U}$ solution of (13) via SVD

return $UB, \hat{\boldsymbol{Z}}, \boldsymbol{U}$.

---

all non-convex quadratically constrained problems, and `Mosek` 10.1.11 to solve all conic relaxations. For the sake of conciseness, we defer full details of our experimental setup to Section EC.9.1. Moreover, for the purpose of averaging results across datasets with different p's, we report the proportion of variance explained whenever we report an objective value. For a correlation matrix, this corresponds to dividing by p, the number of features. We make our code available on `GitHub` at github.com/ryancorywright/MultipleComponentsSoftware.

Description of Data Sources: We perform experiments on eleven datasets from the UCI database in Sections 5.1-5.2 and 5.4, and experiments on synthetic data in Section 5.3. Of the eleven datasets (described in detail in Section EC.9.2), six datasets are overdetermined (meaning $n > p$), while five datasets are underdetermined (meaning $p > n$). Because some of our algorithms can only return PCs with disjoint supports, we sometimes report results over instances where $\sum_t k_t > p$ and where $\sum_t k_t \leq p$ separately. Given the values of (k_t, r) we consider, datasets with $p \geq 60$ are systematically in the second category.

5.1. Performance of Upper Bounds

In this section, we compare the tightness and scalability of the upper bounds obtained from our three approaches. First, we consider two semidefinite relaxations (namely, (9) with (10)-(11), hereafter "Extended-Ineq", and (12), hereafter "Perm-Ineq"). This allows us to determine the merits of specifying k_t versus only specifying k, in terms of whether it leads to tighter relaxations. Second, the Lagrangian upper bound obtained from solving (16) with the exact method of Berk and Bertsimas (2019), hereafter "Lagrangian". Finally, the bound attained by maximizing our combinatorial bound (19) in Theorem 4, hereafter "Combinatorial". We also considered running `Gurobi`'s non-convex branch-and-bound solver directly on this dataset (see Section EC.8). However, experiments

Rank (r)	Sparsity (k, k_t)	Extended-Ineq		Perm-Ineq		Lagrangian		Combinatorial		
		UB	T(s)	UB	T(s)	UB	T(s)	UB	Nodes	T(s)
2	4, -	0.297	20.28	-	-	-	-	0.301	248	0.048
2	4, (1, 3)	**0.267**	20.58	**0.267**	1.97	**0.267**	0.22	0.277	1	0.007
2	4, (2, 2)	**0.295**	0.37	**0.295**	0.47	0.301	0.03	0.301	1	0.007
2	6,	**0.384**	0.78	-	-	-	-	0.396	195	0.046
2	6, (1, 5)	**0.339**	0.44	**0.339**	0.25	**0.339**	0.02	0.360	1	0.006
2	6, (2, 4)	**0.371**	0.47	**0.371**	0.57	0.376	0.02	0.394	1	0.005
2	6, (3, 3)	0.361	0.42	**0.360**	0.45	0.381	0.03	0.396	1	0.009
2	8, -	**0.451**	0.75	-	-	-	-	0.482	182	0.046
2	8, (1, 7)	**0.384**	0.46	**0.384**	0.40	**0.384**	0.02	0.415	1	0.006
2	8, (2, 6)	**0.435**	0.43	**0.435**	0.48	0.440	0.03	0.465	1	0.005
2	8, (3, 5)	0.420	0.52	**0.418**	0.65	0.452	0.02	0.478	1	0.005
2	8, (4, 4)	0.412	0.48	**0.408**	0.54	0.452	0.02	0.482	1	0.007
2	10, -	**0.490**	0.72	-	-	-	-	0.559	213	0.045
2	10, (1, 9)	**0.395**	0.43	**0.395**	0.3	**0.395**	0.02	0.465	1	0.006
2	10, (2, 8)	**0.457**	0.53	**0.457**	0.5	0.463	0.02	0.516	1	0.007
2	10, (3, 7)	0.461	0.41	**0.459**	0.5	0.498	0.03	0.537	1	0.005
2	10, (4, 6)	0.458	0.44	**0.455**	0.6	0.516	0.02	0.553	1	0.005
2	10, (5, 5)	0.453	0.55	**0.449**	0.45	0.524	0.02	0.559	1	0.008
3	6, -	**0.443**	25.76	-	-	-	-	0.445	1872	0.147
3	6, (1, 1, 4)	**0.380**	42.92	**0.380**	5.02	**0.380**	0.05	0.398	1	0.008
3	6, (1, 2, 3)	**0.412**	0.94	**0.412**	1.54	0.418	0.04	0.427	1	0.007
3	6, (2, 2, 2)	**0.435**	0.68	**0.435**	2.65	0.451	0.03	0.445	1	0.009
3	9, -	**0.570**	2.10	-	-	-	-	0.588	1759	0.118
3	9, (1, 1, 7)	**0.461**	0.80	**0.461**	1.21	**0.461**	0.03	0.492	1	0.008
3	9, (1, 2, 6)	**0.512**	0.84	**0.512**	0.59	0.517	0.03	0.542	1	0.007
3	9, (1, 3, 5)	0.497	0.71	**0.495**	0.80	0.529	0.04	0.555	1	0.008
3	9, (1, 4, 4)	0.489	0.65	**0.485**	0.54	0.529	0.03	0.559	1	0.010
3	9, (2, 2, 5)	**0.539**	0.78	**0.539**	0.73	0.563	0.03	0.578	1	0.010
3	9, (2, 3, 4)	0.532	0.77	**0.531**	0.98	0.567	0.04	0.586	1	0.007
3	9, (3, 3, 3)	0.520	0.88	**0.512**	0.58	0.571	0.04	0.588	1	0.012

Table 2   Performance of upper bounds on the pitprops dataset ($p = 13$), as we vary the overall sparsity (k), the number of PCs (r), and the allocation of a sparsity budget to the different PCs. We use a 7,200-second time limit. We denote the best-performing solution (least upper bound) in bold. Note that all results are normalized by dividing by the trace of $\boldsymbol{\Sigma}$, i.e., p, the number of features, to report results in terms of the proportion of variance explained.

on the same dataset show that its upper bound does not scale as well as any of our relaxations when the total sparsity of the PCs is 12 or more (see Tables EC.2-EC.3). Thus, we do not report Gurobi's performance in terms of generating upper bounds in the main paper.

Benchmarking on Pitprops Data We first compare the bounds generated by each method—in terms of proportion of correlation explained—on the pitprops dataset ($n = 180$, $p = 13$) as we vary $r \in \{2, 3\}$ and k, in Table 2. We consider both imposing an overall sparsity budget alone (denoted by "$k, -$") and imposing a separate budget for each PC, denoted by "$k, (k_1, \ldots, k_t)$". We denote the performance of methods that require $\{k_t\}_{t \in [r]}$ by a dash when only k is provided, to indicate they are not applicable.

On the instances presented in Table 2, we observe that the semidefinite relaxations often terminate in less than a second and "Perm-Ineq" is uniformly the strongest relaxation. In particular, when individual sparsity budgets k_t are given, Perm-Ineq provides uniformly and sometimes significantly tighter bounds than Extended-Ineq. However, the Lagrangian bound is often only weaker at the third decimal point and takes one order of magnitude less time to compute, suggesting it may scale better. Finally, the combinatorial upper bound is worse than the bounds from all other methods considered for each instance.

We remind the reader that Perm-Ineq and the Lagrangian bound cannot compute a bound if we do not specify component-specific sparsity budgets k_t. Nonetheless, when only an overall sparsity k is imposed, Extended-Ineq and the combinatorial bound are much weaker than the worst-case bound over all possible allocations $\{k_t\}_{t\in[r]}$ with $\sum_{t\in[r]} k_t = k$. So, time permitting, we recommend computing the upper bound by solving the relaxations for all possible allocations of k_t and taking the worst-case bound. For instance, for $r = 2, k = 10$ with Perm-Ineq, this approach would give an upper bound of 0.459, which is 6.3% better than Extended-Ineq's bound with k alone. Accordingly, in the rest of the paper, we only consider instances of Problem (3) where we know both k and k_t and consider Perm-Ineq and its second-order cone relaxations, but not Extended-Ineq.

Benchmarking on Larger-Scale Datasets We now investigate the scalability of the SDP relaxation Perm-Ineq ("PSD"), its second-order cone relaxation as described in Section EC.4.2 ("SOC" as in Equation (EC.6)), and the aforementioned combinatorial and Lagrangian bounds on larger UCI datasets in Table 3. Note that for the micromass dataset, we only include sets of second-order cone constraints with fewer than $O(p^2)$ members in the SOC relaxation, to avoid excessively memory-intensive problems.

We observe that the PSD relaxation uniformly dominates all other upper bounds and can be solved within a few minutes for $p \approx 100$ but quickly requires a prohibitive amount of memory and time at higher dimensions. The SOC relaxation scales up to $p \approx 1000$ in hours but provides a weaker upper bound, within 1–20% of PSD. On the other hand, the Lagrangian relaxation scales to $p \approx 1000$ in seconds, and outperforms the SOC relaxation's bound on every instance where $\sum_t k_t \leq p$. Finally, the combinatorial bound is uniformly worse than the Lagrangian bound in terms of both runtime and the quality of the bound. Thus, in practice, we recommend using the SDP bound if it can be computed (say $p \leq 300$), and the Lagrangian bound otherwise.

5.2. Performance of Feasible Methods

In this section, we numerically evaluate the quality of our three algorithms in terms of their ability to recover approximately orthogonal and high-quality principal components on real-world datasets. We first validate that our methods are capable of recovering feasible and near-optimal solutions to

Dataset	Dim. (p)	Rank (r)	Sparsity (k, k_t)	PSD		SOC		Lagrangian		Combinatorial	
				UB	T(s)	UB	T(s)	UB	T(s)	UB	T(s)
Pitprops	13	2	10, (5, 5)	0.449	17.81	0.524	0.22	0.524	1.44	0.559	2.12
		2	20, (10, 10)	0.507	0.59	0.672	0.23	0.642	0.02	0.803	0.01
		3	15, (5, 5, 5)	0.616	0.87	0.761	0.42	0.786	0.05	0.827	0.01
		3	30, (10, 10, 10)	0.652	0.72	1.007	0.46	0.963	0.04	1.198	0.01
Wine	13	2	10, (5, 5)	0.458	0.57	0.529	0.25	0.529	0.03	0.579	0.01
		2	20, (10, 10)	0.554	0.55	0.722	0.22	0.707	0.03	0.876	0.01
		3	15, (5, 5, 5)	0.632	0.79	0.762	0.52	0.794	0.05	0.853	0.01
		3	30, (10, 10, 10)	0.665	0.74	1.083	0.57	1.060	0.03	1.296	0.01
Ionosphere	34	2	10, (5, 5)	0.209	8.22	0.221	1.45	0.221	0.10	0.228	0.03
		2	20, (10, 10)	0.305	8.35	0.363	1.53	0.361	0.04	0.401	0.02
		2	40, (20, 20)	0.378	9.84	0.504	1.85	0.500	0.05	0.618	0.02
		3	15, (5, 5, 5)	0.297	12.74	0.331	4.39	0.331	0.05	0.340	0.04
		3	30, (10, 10, 10)	0.411	11.93	0.545	4.49	0.542	0.07	0.597	0.04
		3	60, (20, 20, 20)	0.464	13.38	0.757	5.95	0.749	0.08	0.920	0.03
Communities	101	2	10, (5,5)	0.095	373.2	0.096	29.02	0.096	0.21	0.097	1.33
		2	20, (10, 10)	0.169	307.7	0.175	44.83	0.175	0.29	0.180	0.89
		2	40, (20, 20)	0.263	373.9	0.286	40.13	0.284	0.52	0.320	0.72
		3	15, (5, 5, 5)	0.141	441.6	0.144	80.30	0.144	0.26	0.146	0.25
		3	30, (10, 10, 10)	0.245	564.2	0.262	65.25	0.262	0.27	0.270	0.24
		3	60, (20, 20, 20)	0.378	447.3	0.429	54.71	0.425	0.30	0.476	0.26
Arrhythmia	274	2	10, (5, 5)	-	-	0.031	54.71	0.031	0.60	0.032	10.83
		2	20, (10, 10)	-	-	0.055	456.0	0.055	0.31	0.060	7.69
		2	40, (20, 20)	-	-	0.086	445.8	0.084	0.39	0.105	5.49
		3	15, (5, 5, 5)	-	-	0.047	671.8	0.046	0.43	0.049	30.80
		3	30, (10, 10, 10)	-	-	0.083	812.2	0.083	0.57	0.089	11.32
		3	60, (20, 20, 20)	-	-	0.129	803.7	0.126	0.64	0.155	8.54
Micromass	1300	2	10, (5, 5)	-	-	0.008	1089	0.008	2.98	0.008	854.6
		2	20, (10, 10)	-	-	0.015	13620	0.014	4.86	0.014	178.3
		2	40, (20, 20)	-	-	0.027	9213	0.023	4.03	0.025	154.5
		3	15, (5, 5, 5)	-	-	0.012	7953	0.011	5.40	0.012	> 7200
		3	30, (10, 10, 10)	-	-	0.023	19640	0.021	3.51	0.021	1092
		3	60, (20, 20, 20)	-	-	0.043	18630	0.034	4.39	0.038	649.0

Table 3 Performance of conic and combinatorial bounds across UCI datasets. All bounds are normalized by dividing by $p = \operatorname{tr}(\boldsymbol{\Sigma})$, the number of features, to report in terms of proportion of correlation explained. The notation "-" denotes that an instance could not be solved using the provided memory budget, namely 32 GB for instances where $p \leq 101$, 100 GB for instances where $p \in [102, 250]$, 370 GB for instances where $p > 250$.

small-scale sparse PCA problems with multiple PCs. We then compare our algorithms with five state-of-the-art techniques on eleven UCI datasets. Since Algorithm 2 and most of the benchmarked algorithms from the literature require that the sparsity of each PC is specified separately, we consider this formulation in this section (and fix $k_t = k/r$ for concision).

Benchmarking on Pitprops Data: We first investigate the performance of Algorithms 1–3 on the `pitprops` dataset because Lu and Zhang (2012) used this dataset to extensively benchmark several algorithms for sparse PCA with $r = 6$ PCs. All six methods they benchmarked required an overall sparsity of around 45–60 to explain less than 70% of the correlation. The best performing method could explain 69.55% of the correlation with an overall sparsity of 46 (Lu and Zhang 2012, Table

11). They concluded that "there do not exist six highly sparse, nearly orthogonal and uncorrelated PCs while explaining most of variance" (note that the pitprops dataset is normalized, and hence variance should be read as correlation here).

As reported in Table EC.1, with $r = 6$ PCs and $k_t = 2$, hence an overall sparsity of 12, solutions returned by Algorithms 2–3 explain 74%–75% of the correlation, and our SDP upper bound from Algorithm 1 proves that it is not possible to explain more than 75% of the correlation in this case. Moreover, Algorithm 2 even provides a solution that explains 81% of the correlation with an overall sparsity of $6 \times 4 = 24$. In other words, the solutions that were previously believed to be unattainable at this sparsity level are achievable with our methods, and can be computed within seconds on this instance.

Benchmarking on Larger-Scale Datasets We now investigate the performance of Algorithms 1–3 on eleven UCI datasets summarized in Table EC.4, whose dimensions range from $p = 13$ (`pitprops`) to $p = 1300$ (`micromass`). We compare with four state-of-the-art methods. Namely,

- The branch-and-bound method of Berk and Bertsimas (2019) for optimally computing one sparse PC, combined with the deflation scheme of Mackey (2008) to obtain multiple PCs, implemented in `Julia` and made available at `github.com/lauren897/Optimal-SPCA`. According to Berk and Bertsimas (2019), this method outperformed four others across three UCI datasets ($r = 3, k = 5$).
- The deflation method of Hein and Bühler (2010), using the custom deflation method developed in Bühler (2014), implemented in `Matlab` and made publicly available at `github.com/tbuehler/sparsePCA`, using default parameters. This approach was found by Berk and Bertsimas (2019, Table 9) to be second-best of the methods in their comparison.
- The `Lasso`-inspired method of Zou et al. (2006), using the `spca` function in the `elasticnet` package version 1.3, using default parameters. This approach is perhaps the most commonly used one in practice, since it is distributed via the ubiquitous `elasticnet` package.
- The covariance thresholding method of Deshpande and Montanari (2014b), building upon the works of Krauthgamer et al. (2015), which relies on applying a soft-thresholding operation first on the entries of the covariance matrix and then on its r leading eigenvectors. We implemented this method natively in `Julia` and release it as part of our codebase.

Finally, we also compare with Gurobi's non-convex branch-and-bound solver as a benchmark, as described in Section EC.8. We do not report upper bounds for this benchmark, because they are uniformly above 1, as shown for the pitprops dataset in Tables EC.2–EC.3, and sometimes no upper bound can be computed within the time limit, giving an average upper bound of $+\infty$.

Algorithm 1 involves solving a convex relaxation from Section 2. Based on the scalability results presented in the previous section, we use different convex relaxations depending on the dimensionality of the problem. Namely, for Algorithm 1, we use the full semidefinite relaxation (12) if $p \leq 101$, and the SOC relaxation (EC.6) otherwise.

We report summary results in Tables 4-5 (see also Tables EC.5–EC.10 in Section EC.9.4 for instance-wise results). In particular, we report the average objective value attained by each method, the average upper bound attained by each method (where applicable), and the average gap between the feasible solution and the best upper bound found by *any* of our three methods. Our results are broken down by whether $\sum_t k_t \leq p$ or $\sum_t k_t > p$, since Algorithms 1 and 3 are incapable of returning PCs that have a collective sparsity larger than p. Finally, when the returned solution violates the orthogonality condition, its objective value is not necessarily a valid bound on the objective of (3) and the average reported gap is thus an optimistic estimate.

Method	Obj.	UB	Rel. gap* (%)	Viol.	T(s)
Algorithm 1	0.135	0.168	17.99%	0	2220
Algorithm 2	0.144	0.177	8.82%	0	70.66
Algorithm 3	0.151	0.188	3.11%	0	126.9
Branch-and-bound	0.137	-	14.73%	0	6945
Berk and Bertsimas (2019)	0.152	-	2.63%	0.012	29.81
Deshpande and Montanari (2014b)	0.144	-	7.73%	0.063	14.09
Hein and Bühler (2010)	0.124	-	25.75%	0.016	0.23
Zou et al. (2006)	0.033	-	81.68%	1.340	5.66

Table 4   Average performance of methods across the 11 UCI datasets described in Table EC.4 with $k \in \{5, 10, 20\}, r \in \{2, 3\} : k \leq p$, conditioning on instances where $kr \leq p$ (in particular, kr ranges from 10 to 60 in our experiments). *Optimality gaps computed using the best UB from Algorithms 1–3. Methods that violate orthogonality can achieve a negative gap.

Method	Obj.	UB	Rel. gap* (%)	Viol.	T(s)
Algorithm 1	0.422	0.532	21.49%	0	7.84
Algorithm 2	0.519	0.733	2.82%	0	6.71
Algorithm 3	0.477	0.878	10.43%	0	0.76
Branch-and-bound	0.485	-	10.35%	0	> 7200
Berk and Bertsimas (2019)	0.519	-	2.78%	0.146	0.59
Deshpande and Montanari (2014b)	0.510	-	4.41%	0.279	0.01
Hein and Bühler (2010)	0.507	-	5.26%	0.066	0.032
Zou et al. (2006)	0.146	-	73.13%	2.168	0.76

Table 5   Average performance of methods across the 11 UCI datasets described in Table EC.4 with $k \in \{5, 10, 20\}, r \in \{2, 3\} : k \leq p$, conditioning on instances where $kr > p$ (in particular, this criterion excludes datasets with $p \geq 60$). *Optimality gaps computed using the best UB from Algorithms 1–3. Methods that violate orthogonality can achieve a negative gap.

First, we observe that our three methods and branch-and-bound are the only ones to return PCs that are systematically orthogonal (with an average orthogonality violation $< 10^{-3}$). Among them, we observe that Algorithm 2 performs the best on datasets where $\sum_{t \in [r]} k_t = kr > p$ and Algorithm 3 performs best on datasets where $kr \leq p$, with average relative optimality gaps of 2.78% and 3.11% on these respective instances, and average runtimes in the tens or hundreds of seconds

for both methods. On the other hand, Algorithm 1 uniformly provides the best upper bounds of all methods, although it is dominated by Algorithms 2–3 in terms of solution quality and requires two orders of magnitude more runtime to compute.

Of the remaining methods, the method of Berk and Bertsimas (2019) performs the next best, in terms of explaining a large proportion of the correlation (0.137 when $kr \leq p$ and 0.485 when $kr > p$) while not violating feasibility too significantly (0.012 when $kr \leq p$ and 0.146 when $kr > p$). However, its higher objective value than Algorithm 3 on instances where $kr \leq p$ is driven by instances where it achieves an objective value higher than Algorithm 1's upper bound by violating the orthogonality constraint. Runtimes of Algorithms 2-3 and of the method of Berk and Bertsimas (2019) are also of similar order of magnitude, although the method of Berk and Bertsimas (2019) is faster on average. Finally, the methods of Hein and Bühler (2010), Deshpande and Montanari (2014b), Zou et al. (2006) repeatedly violate the orthogonality constraint and explain less correlation than Algorithm 2 on average.

Finally, we remark that no one method performs best on every instance. Algorithm 2 appears to perform the best overall on instances where $kr > p$ and Algorithm 3 on instances where $kr \leq p$. However, some other methods (e.g. Berk and Bertsimas 2019) perform best on some instances. These results suggest that both k and the amount of overlap between the optimal PCs impact the performance of each method, and motivate a comparison on synthetic data, where we control the ground truth, in the next section.

5.3. Statistical Recovery on Synthetic Data

To evaluate the support recovery ability of each method, we now compare the performance of Algorithms 1, 2, and 3 against the same four methods from the literature on synthetic data. We use a spiked Wishart model with multiple spikes, as in Deshpande and Montanari (2014a), Ding et al. (2024), to generate data. Namely, we consider an underlying covariance matrix of the form $\boldsymbol{\Sigma} = \boldsymbol{I}_p + \beta \boldsymbol{x}_1 \boldsymbol{x}_1^\top + \beta \boldsymbol{x}_2 \boldsymbol{x}_2^\top$, where $\beta = 1$ is the signal-to-noise ratio. The vectors $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \{-1, 0, 1\}^p$ are random k_{true}-sparse orthogonal vectors. We also control the proportion of overlap between the supports of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, $q \in [0, 1]$ ($q = 0$ corresponds to disjoint support while $q = 1$ corresponds to row sparsity). In our experiments, we take $p = 50$, $k_{true} = 20$ and vary $q \in \{0.1, 0.5, 0.9\}$. Finally, we sample n observations from a multivariate centered normal distribution with covariance matrix $\boldsymbol{\Sigma}$ and construct the empirical covariance matrix $\hat{\boldsymbol{\Sigma}}$. We investigate the performance of different methods as n increases (so that the empirical covariance matrix converges to the underlying truth, $\boldsymbol{\Sigma}$). We impose a limit of 100 iterations for Algorithm 2.

For each algorithm, we compute the fraction of variance explained and the feasibility violation (i.e., the inner product between the two PCs computed). We average these performance metrics

over 20 random instances and report them in Figure 1. The method of Zou et al. (2006) is clearly dominated by all other methods since it explains a significantly lower fraction of the variance, while returning the least orthogonal vectors. In terms of objective value (left panel), we observe that Algorithm 2, Berk and Bertsimas (2019), Hein and Bühler (2010) perform almost identically, followed closely by covariance thresholding. They all explain a larger fraction of the variance than Algorithms 1 and 3. However, we observe on the right panel that Algorithms 1, 2, and 3 are the only methods to return orthogonal PCs, across all values of n. In addition, the gap between the methods (and especially the gap between the four best performing methods) seems to shrink as q increases, i.e., when the overlap between the support increases.

Since the data are synthetically generated, we can also evaluate the ability to recover the true support. For two k_{true}-sparse candidate PCs $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, we measure how well $\mathrm{supp}(\boldsymbol{u}_1) \cup \mathrm{supp}(\boldsymbol{u}_2)$ recovers the support of $\mathrm{supp}(\boldsymbol{x}_1) \cup \mathrm{supp}(\boldsymbol{x}_2)$ in terms of accuracy and false detection rate:

$$A := \frac{|S \cap S^\star|}{|S^\star|},$$

with $S = \mathrm{supp}(\boldsymbol{u}_1) \cup \mathrm{supp}(\boldsymbol{u}_2)$ and $S^\star = \mathrm{supp}(\boldsymbol{x}_1) \cup \mathrm{supp}(\boldsymbol{x}_2)$. This definition of support recovery corresponds to the one used in statistical studies for sparse PCA with multiple PCs (e.g., Deshpande and Montanari 2014b). Figure 2 (left panel) reports the value of A for the different algorithms as n increases. Since we do not explicitly control for the overlap between the returned PCs (except for Algorithms 1 and 3), methods might differ in the size of the support they return, $|S|$. Hence, to allow for a fair comparison, we also report $|S|$ in the right panel of Figure 2. Regarding A, we observe that Algorithm 1 detects a noticeably higher fraction of the true features than other methods, which is not surprising given the fact that it returns disjoint supports. For the remaining methods, their relative performance is aligned with their performance in terms of fraction of variance explained (Figure 1, left panel).
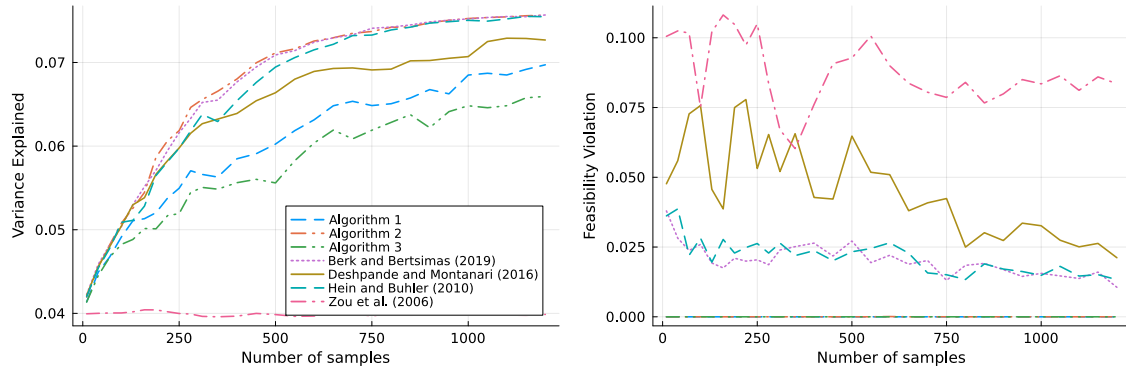
(a) Support overlap $q = 0.1$
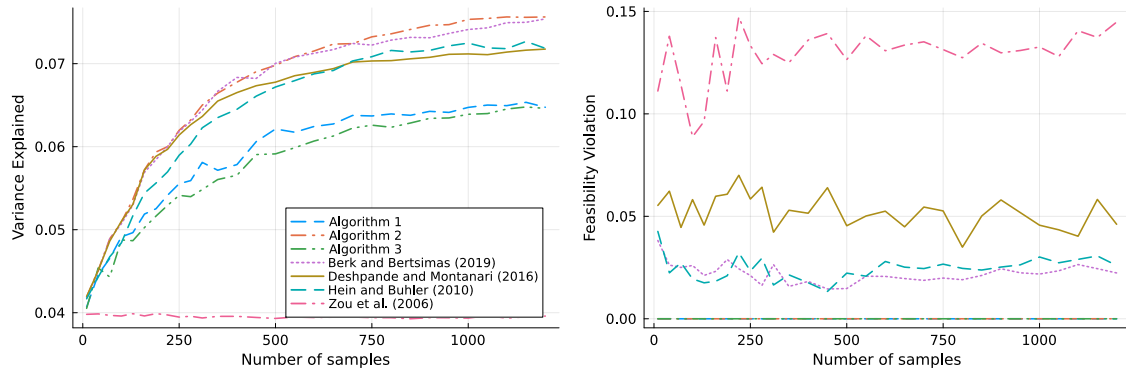
(b) Support overlap $q = 0.5$
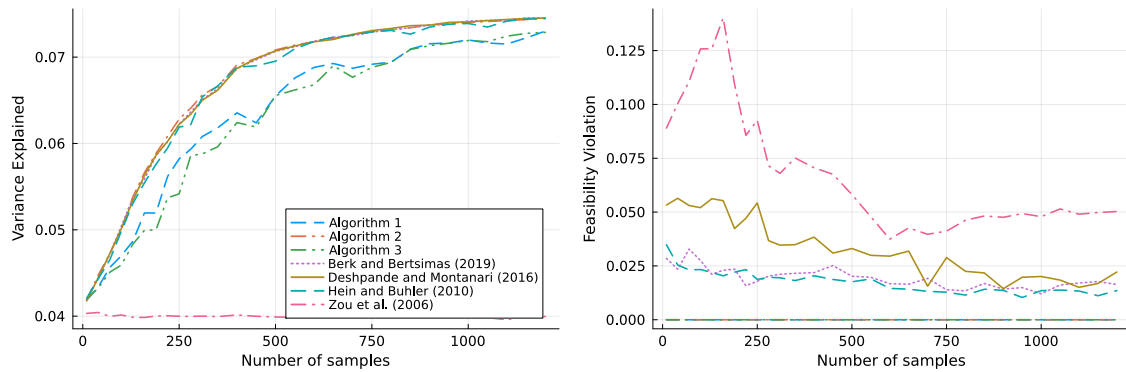
(c) Support overlap $q = 0.9$

Figure 1   Variance explained (left panel) and feasibility violation (right panel) on synthetic instances of sparse PCA with two 20-sparse PCs with partially overlapping support. Results are averaged over 20 replications.

(a) Support overlap $q = 0.1$

(b) Support overlap $q = 0.5$

(c) Support overlap $q = 0.9$

Figure 2   Accuracy (left panel) and joint support size (right panel) for the recovery of $\mathrm{supp}(\boldsymbol{x}_1) \cup \mathrm{supp}(\boldsymbol{x}_2)$, on synthetic instances of sparse PCA with two 20-sparse PCs with partially overlapping support. Results are averaged over 20 replications.
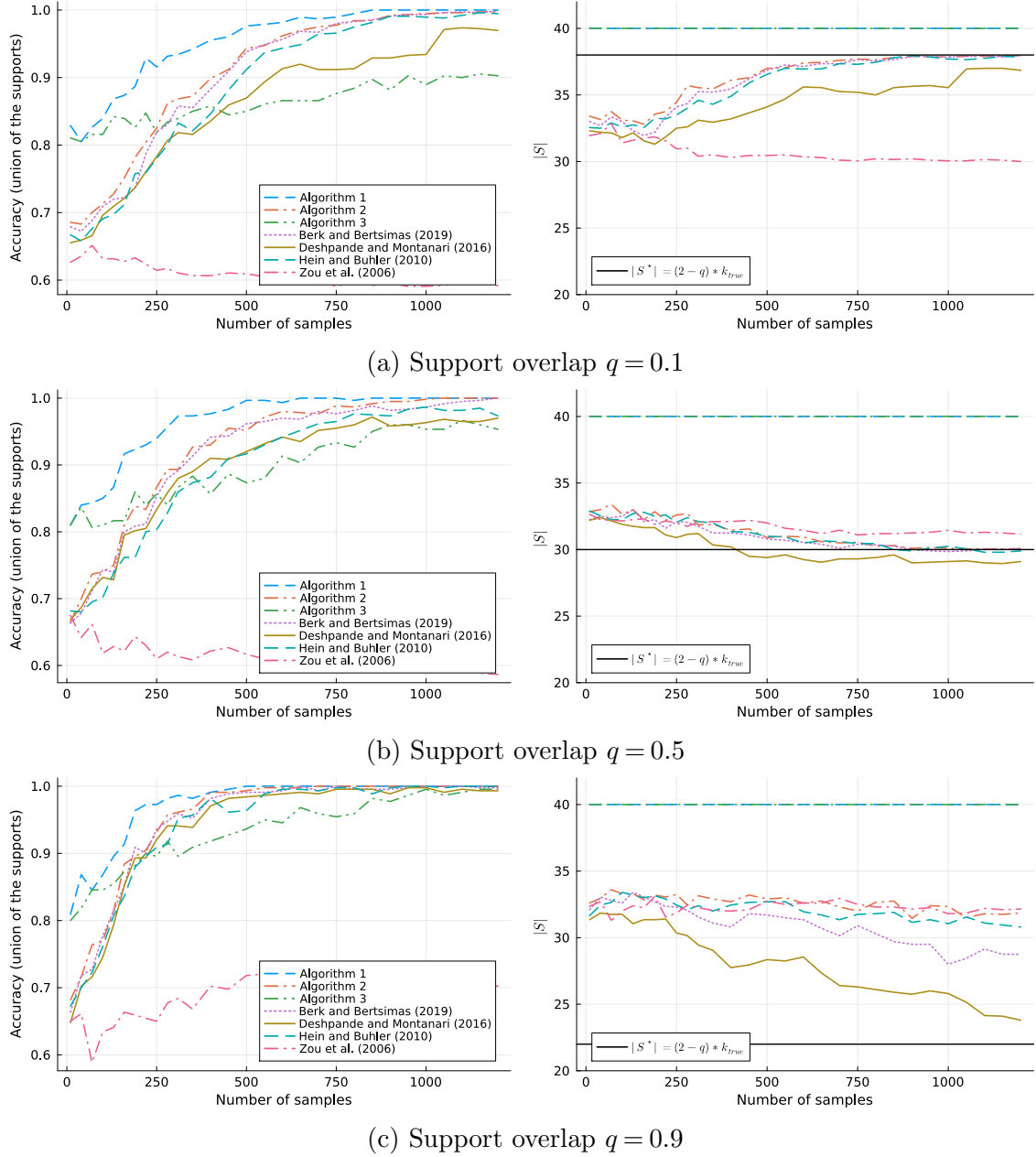
5.4. Specifying the Sparsity Pattern: The Benefits of Asymmetry

While Problem (3) only requires a bound on the total sparsity, thus allowing flexibility on how this budget is allocated across PCs, the worst-case semidefinite upper bound over all sparsity patterns $(k_1, \ldots, k_r) : \sum_{t \in [r]} k_t = k$ is often significantly tighter than the semidefinite relaxation of (3) with a sparsity budget of k alone, as demonstrated in Section 5.1. Moreover, Algorithm 2, which as demonstrated in Sections 5.2–5.3 is currently the best performing method for obtaining feasible solutions to Problem (3), requires that $(k_1, \ldots, k_t)$ are individually specified. Collectively, these observations suggest that it may be necessary to enumerate all allocations of the sparsity budget k, which could be expensive. In our experiments, as is often done in practice, we restricted our search to symmetric allocations. In this section, we revisit the symmetry assumption, investigate when it is justified, and study the relative benefits of asymmetric sparsity budget allocations in terms of obtaining equally sparse sets of PCs that explain more variance.

We consider the `pitprops`, `ionosphere`, `geographical`, and `communities` UCI datasets with a fixed number of PCs $r = 3$ and a given overall sparsity budget $k \in \{15, 30\}$. Accordingly, in Figures 3 and EC.1 of Section EC.9.5, we depict the relationship between the proportion of correlation explained in the data for each possible allocation of the sparsity budget $(k_1, k_2, k_3) : k_1 + k_2 + k_3 = k, p \geq k_1 \geq k_2 \geq k_3 \geq 1$, as computed by Algorithm 2 with a limit of 200 iterations and the same setup as in Section 5.2 (and the corresponding upper bound computed by Algorithm 1), against the relative asymmetry in the sparsity budget, as measured by

$$\frac{\mathrm{KL}((k_1, k_2, k_3) \| (k/3, k/3, k/3))}{\max_{p \geq k_1 \geq k_2 \geq k_3 : k_1 + k_2 + k_3 = k} \mathrm{KL}((k_1, k_2, k_3) \| (k/3, k/3, k/3))},$$

where $KL(p \| q) := \sum_i p_i \log(p_i / q_i)$ denotes the KL divergence.

We observe a general trend that more symmetric sparsity budget allocations tend to explain more of the correlation in the data (both in terms of actual correlation explained by a solution from Algorithm 2 and in terms of the upper bound). This suggests that, when time is a concern, requiring that all PCs are equally sparse is a reasonable approach.

Table 6 compares the quality of Algorithm 2's solution (a) when all PCs have the sparsity budget of k/r and (b) the maximum possible correlation explained over all feasible allocations of the sparsity budget k (computed by enumerating all possible allocations of the sparsity budget-18 such allocations for $k = 15, p \geq k$ and 74 allocations for $k = 30, p \geq 30$), together with the upper bound on the proportion of correlation explained obtained in each case. We observe that in several instances a perfectly symmetric allocation of the sparsity budget yields the highest quality solution, and in all instances a perfectly symmetric allocation is within 7% in the worst-case (and within 1.75% on average) of the best solution. In the enumerated case, we also compute the optimality gap between the worst-case upper bound over all sparsity budget allocations, and the best solution found, and
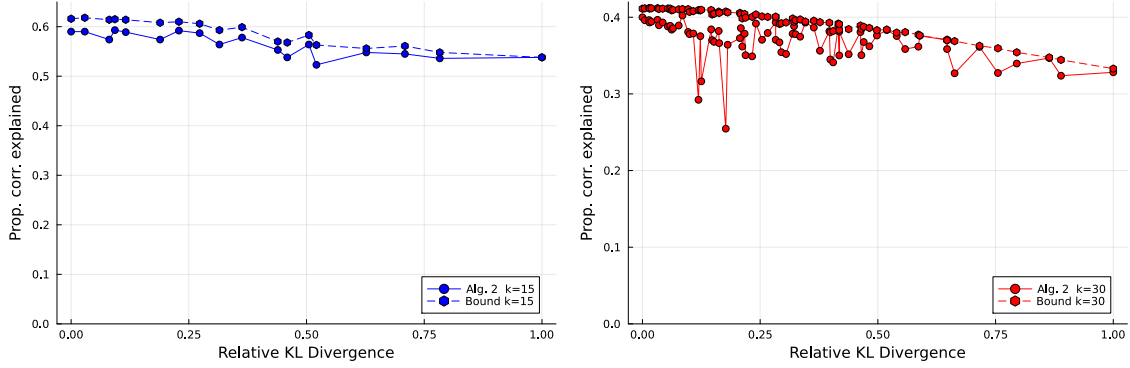
(a) Pitprops data ($k = 15$, $r = 3$)

(b) Ionosphere data ($k = 30$, $r = 3$)

Figure 3   Asymmetry of sparsity budget allocation (the higher the relative KL divergence, the further away (k_1, k_2, k_3) is from a symmetric allocation) vs. proportion of correlation explained on the pitprops (left panel, $k = 15$, $r = 3$) and ionosphere (right panel, $k = 30$, $r = 3$) datasets. For the proportion of correlation explained, we report both an upper bound (obtained from solving our semidefinite relaxation) and a lower bound (obtained from the solution of Algorithm 2). Note that the normalizing constant for the KL divergence is different for each dataset, as the value of k is different.

observe that on average it is less than 1% over the instances considered. Note that this is a different gap to the one reported in Section 5.2, where the upper bound is computed after assuming that all PCs are equally sparse.

Dataset	p	r	k	Symmetric			Enumerated					Improvement (%)
			k	UB	Obj.	Viol.	UB	k_t	Obj.	Rel. gap (%)	Viol.	
Pitprops	13	3	15	0.616	0.590	0.000	0.618	(6, 6, 3)	0.593	4.06%	0.000	0.47%
			30	0.652	0.650	0.000	0.652	(10, 10, 10)	0.650	0.19%	0.000	0%
Ionosphere	34	3	15	0.297	0.286	0.000	0.299	(7, 6, 2)	0.299	0%	0.001	4.48%
			30	0.411	0.400	0.000	0.412	(15, 8, 7)	0.402	2.34%	0.000	0.60%
Geographical	68	3	15	0.221	0.221	0.000	0.221	(5, 5, 5)	0.221	0%	0.000	0.00%
			30	0.410	0.389	0.000	0.420	(12, 12, 6)	0.415	1.07%	0.000	6.29%
Communities	101	3	15	0.141	0.141	0.000	0.142	(6, 5, 4)	0.142	0.02%	0.000	0.48%
			30	0.246	0.243	0.000	0.247	(11, 10, 9)	0.247	0.02%	0.000	1.71%

Table 6   Comparison of symmetric and enumerated solution computed by Algorithm 2 for a given sparsity budget k_{total} and given number of PCs ($r = 3$). We report the largest upper bound over all sparsity budget allocations as our enumerated upper bound, and report the relative optimality gap between the best solution and the worst-case bound. On average, considering asymmetric sparsity budget allocations improves the proportion of correlation explained by 1.65%.

In summary, more symmetric allocations of the sparsity budget tend to perform better on average. Therefore, for a given sparsity budget k, a reasonable strategy could be to (a) run Algorithm 2 to compute a perfectly symmetric allocation, (b) compute an upper bound using (12) across all possible allocations, and (c) run Algorithm 2 only on the asymmetric allocations for which the upper bound from (b) allows for a significant potential improvement upon the symmetric solution.

5.5. Summary and Guidelines From Numerical Experiments

In summary, our main findings from our numerical experiments are as follows:

- As reflected in Section 5.1, the semidefinite relaxation (12) provides the tightest upper bound and can currently be computed within minutes for $p \leq 300$. At larger scales, the Lagrangian bound we develop in Algorithm 2 is competitive when $kr \ll p$ and can typically be computed in seconds even when p is in the thousands.

- In terms of solution quality, Algorithms 2–3 substantially outperform existing methods, obtaining higher-quality solutions that satisfy the orthogonality constraints (Section 5.2). Moreover, both algorithms require 100 seconds or less to run, on average. Therefore, they should be considered as viable and accurate approaches for sparse PCA problems with multiple PCs.

- As demonstrated in Section 5.2, Algorithm 3 gives certifiably near-optimal sparse PCs when $kr \leq p$ (3.11% bound gap across UCI instances on average), while Algorithm 2 gives certifiably near-optimal sparse PCs when $kr > p$ (2.82% bound gap across UCI instances on average). Thus, using the techniques in the paper, it is possible to compute sparse PCs that are near-optimal in a practical amount of time, even when p is in the thousands.

- However, we should note that these optimality gaps are computed using the tightest (i.e., typically Algorithm 1's) upper bound. Using its own upper bound instead, Algorithm 3 (resp. Algorithm 2) returns solutions with an average bound gap of 11.32% (resp. 28.21%) on datasets where $kr \leq p$ (resp. $kr > p$). So, while Algorithms 2–3 find these solutions in minutes when $p = 1000$s, it currently takes hours to compute the conic bound and certify their near optimality. This observation emphasizes the benefit of combining these different approaches together, and motivates future research to improve the scalability of the conic bounds.

- If practitioners have an overall sparsity budget but are agnostic about the sparsity of each column, a reasonable strategy is to require that all columns are equally sparse (i.e., set $k_t = k/r$), as shown in Section 5.4. In our experiments, considering asymmetrically sparse sets of PCs increases the amount of correlation explained by around 2% on average, at the cost of an order-of-magnitude increase in total runtime.

6. Conclusion

In this paper, we studied the problem of selecting a set of mutually orthogonal sparse principal components and proposed three algorithms which, collectively, for the first time, allow this problem to be solved to certifiable near-optimality with 100s or 1000s of features in minutes or hours. We propose a semidefinite relaxation (Section 2) which generates high-quality upper bounds on the amount of variance explainable by any set of sparse and mutually orthogonal components, and round this relaxation to obtain a feasible set of sparse PCs in Algorithm 1. Further, we propose a

Lagrangian alternating maximization scheme (Algorithm 2 in Section 3) which gives an alternative upper bound and high-quality feasible solutions. Finally, we derived a new combinatorial upper bound on sparse PCA with multiple PCs, and used this bound to design a new relax-and-round scheme (Algorithm 3 in Section 4) which also gives a valid upper bound and high-quality feasible solutions. Across suites of numerical experiments (Section 5), we demonstrate that the best solution obtained by any of our methods substantially outperforms methods from the literature in terms of obtaining sparse and mutually orthogonal PCs that explain most of the variance in a dataset, and nearly matches the tightest upper bound obtained by any of our methods. All in all, we solve sparse PCA with multiple components to near-optimality for $p = 1000$s and $r > 1$ for the first time.

Acknowledgments

Endnotes

1. Most sparse PCA algorithms for $r = 1$ require the input covariance matrix to be semidefinite. Accordingly, since $\text{tr}(\boldsymbol{U}_t \boldsymbol{U}_t^\top) = 1$, we can add a constant term $\lambda_{\text{offset}} \text{tr}(\boldsymbol{U}_t \boldsymbol{U}_t^\top)$ to the objective without impacting the optimal solution. Thus, we can pick $\lambda_{\text{offset}} > 0$ to be sufficiently large that the matrix $\boldsymbol{\Sigma} - \sum_{t' \neq t} \lambda_{t,t'} \boldsymbol{U}_{t'} \boldsymbol{U}_{t'}^\top + \lambda_{\text{offset}} \mathbb{I}$ and use these algorithms off-the-shelf.

2. As suggested by an anonymous reviewer, (18) could be derived from linear algebra principles directly, instead of as a corollary of Theorem 4. Denoting $r_i := \sum_{j \in [p]} |\Sigma_{i,j}|$, the matrix $\text{Diag}(\boldsymbol{r}) - \boldsymbol{\Sigma}$ is diagonally dominant, hence positive semidefinite. Therefore, we have $\text{Diag}(\boldsymbol{r}) \succeq \boldsymbol{\Sigma}$. In particular, the ordered vector of eigenvalues of $\text{Diag}(\boldsymbol{r})$ majorizes that of $\boldsymbol{\Sigma}$ (section L.1 of Marshall and Olkin 1979), which leads precisely to (18).

3. Actually, one could derive an alternative proof of Theorem 4 that relies on linear algebra principles, in much the same way as the alternative proof of (19). We detail this alternative proof technique in Section EC.7.

References

Ahmadi AA, Dash S, Hall G (2017) Optimization over structured subsets of positive semidefinite matrices via column generation. *Discrete Optimization* 24:129–151.

Alizadeh F, Goldfarb D (2003) Second-order cone programming. *Mathematical Programming* 95(1):3–51.

Amini AA, Wainwright MJ (2008) High-dimensional analysis of semidefinite relaxations for sparse principal components. *2008 IEEE International Symposium on Information Theory*, 2454–2458 (IEEE).

Asteris M, Papailiopoulos D, Kyrillidis A, Dimakis AG (2015) Sparse PCA via bipartite matchings. *Advances in Neural Information Processing Systems* 28.

Atamtürk A, Gomez A (2025) Rank-one convexification for sparse regression. *Journal of Machine Learning Research* 26(35):1–50.

Avellaneda M, Lee JH (2010) Statistical arbitrage in the US equities market. *Quantitative Finance* 10(7):761–782.

Barker G, Carlson D (1975) Cones of diagonally dominant matrices. *Pacific Journal of Mathematics* 57(1):15–32.

Behdin K, Mazumder R (2025) Sparse PCA: A new scalable estimator based on integer programming. *Annals of Statistics* .

Benidis K, Sun Y, Babu P, Palomar DP (2016) Orthogonal sparse PCA and covariance estimation via procrustes reformulation. *IEEE Transactions on Signal Processing* 64(23):6211–6226.

Berk L, Bertsimas D (2019) Certifiably optimal sparse principal component analysis. *Mathematical Programming Computation* 11(3):381–420.

Berthet Q, Rigollet P (2013) Optimal detection of sparse principal components in high dimension. *The Annals of Statistics* 41(4):1780–1815.

Bertsekas DP (1996) *Constrained Optimization and Lagrange Multiplier Methods* (Athena Scientific).

Bertsimas D, Cory-Wright R (2020) On polyhedral and second-order cone decompositions of semidefinite optimization problems. *Operations Research Letters* 48(1):78–85.

Bertsimas D, Cory-Wright R, Pauphilet J (2021) A unified approach to mixed-integer optimization problems with logical constraints. *SIAM Journal on Optimization* 31(3):2340–2367.

Bertsimas D, Cory-Wright R, Pauphilet J (2022a) Mixed-projection conic optimization: A new paradigm for modeling rank constraints. *Operations Research* 70(6):3321–3344.

Bertsimas D, Cory-Wright R, Pauphilet J (2022b) Solving large-scale sparse PCA to certifiable (near) optimality. *Journal of Machine Learning Research* 23(13):1–35.

Bertsimas D, Kitane DL (2023) Sparse PCA: A geometric approach. *Journal of Machine Learning Research* 24:32–1.

Bienstock D, Pia AD, Hildebrand R (2023) Complexity, exactness, and rationality in polynomial optimization. *Mathematical Programming* 197(2):661–692.

Boutsidis C, Drineas P, Magdon-Ismail M (2011) Sparse features for PCA-like linear regression. *Advances in Neural Information Processing Systems* 24.

Bresler G, Park SM, Persu M (2018) Sparse PCA from sparse linear regression. *Advances in Neural Information Processing Systems* 31.

Bühler T (2014) *A flexible framework for solving constrained ratio problems in machine learning.* Ph.D. thesis, Saarland University.

d'Aspremont A, Bach F, El Ghaoui L (2008) Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research* 9(7).

d'Aspremont A, El Ghaoui L, Jordan MI, Lanckriet GR (2007) A direct formulation for sparse PCA using semidefinite programming. *SIAM Review* 49(3):434–448.

Del Pia A (2023) Sparse PCA on fixed-rank matrices. *Mathematical Programming* 198(1):139–157.

Deshpande Y, Montanari A (2014a) Information-theoretically optimal sparse PCA. *2014 IEEE International Symposium on Information Theory*, 2197–2201 (IEEE).

Deshpande Y, Montanari A (2014b) Sparse PCA via covariance thresholding. *Advances in Neural Information Processing Systems* 27.

Dey SS, Mazumder R, Wang G (2022a) Using ℓ_1-relaxation and integer programming to obtain dual bounds for sparse PCA. *Operations Research* 70(3):1914–1932.

Dey SS, Molinaro M, Wang G (2022b) Solving sparse principal component analysis with global support. *Mathematical Programming* 199:421–459.

Ding Y, Kunisky D, Wein AS, Bandeira AS (2024) Subexponential-time algorithms for sparse PCA. *Foundations of Computational Mathematics* 24:865–914.

Eckart C, Young G (1936) The approximation of one matrix by another of lower rank. *Psychometrika* 1(3):211–218.

Fan J, Liao Y, Wang W (2016) Projected principal component analysis in factor models. *Annals of Statistics* 44(1):219.

Fisher ML (1981) The lagrangian relaxation method for solving integer programming problems. *Management Science* 27(1):1–18.

Gally T, Pfetsch ME (2016) Computing restricted isometry constants via mixed-integer semidefinite programming. *Optimization Online* .

Garey MR, Johnson DS (1979) *Computers and Intractability: A Guide to the Theory of NP-completeness* (W.H.Freeman & Co Ltd).

Günlük O, Linderoth J (2010) Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Mathematical Programming* 124(1):183–205.

Gupta SD, Van Parys BP, Ryu EK (2023) Branch-and-bound performance estimation programming: A unified methodology for constructing optimal optimization methods. *Mathematical Programming* .

Hein M, Bühler T (2010) An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA. *Advances in Neural Information Processing Systems* 23.

Hopcroft JE, Karp RM (1973) An nˆ5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing* 2(4):225–231.

Horn RA, Johnson CR (1985) *Matrix analysis* (Cambridge University Press, New York).

Hotelling H (1933) Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24(6):417.

Jeffers JN (1967) Two case studies in the application of principal component analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 16(3):225–236.

Johnstone IM, Lu AY (2009) On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association* 104(486):682–693.

Jolliffe IT, Trendafilov NT, Uddin M (2003) A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics* 12(3):531–547.

Journée M, Nesterov Y, Richtárik P, Sepulchre R (2010) Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research* 11(2).

Kim J, Tawarmalani M, Richard JPP (2022) Convexification of permutation-invariant sets and an application to sparse principal component analysis. *Mathematics of Operations Research* 47(4):2547–2584.

Krauthgamer R, Nadler B, Vilenchik D (2015) Do semidefinite relaxations solve sparse pca up to the information limit? *The Annals of Statistics* 43(3):1300–1322.

Li Y, Xie W (2024) Beyond symmetry: Best submatrix selection for the sparse truncated SVD. *Mathematical Programming* 208:1–50.

Li Y, Xie W (2025) Exact and approximation algorithms for sparse PCA. *INFORMS Journal on Computing* 37(3):582–602.

Lu Z, Zhang Y (2012) An augmented Lagrangian approach for sparse principal component analysis. *Mathematical Programming* 135(1):149–193.

Lubin M, Vielma JP, Zadik I (2022) Mixed-integer convex representability. *Mathematics of Operations Research* 47(1):720–749.

Mackey L (2008) Deflation methods for sparse PCA. *Advances in Neural Information Processing Systems* 21.

Marshall AW, Olkin I (1979) *Inequalities: Theory of Majorization and its Applications* (Academic, New York).

Naikal N, Yang AY, Sastry SS (2011) Informative feature selection for object recognition via sparse PCA. *2011 International Conference on Computer Vision*, 818–825 (IEEE).

Nocedal J, Wright SJ (2006) *Numerical Optimization* (Springer).

Overton ML, Womersley RS (1992) On the sum of the largest eigenvalues of a symmetric matrix. *SIAM Journal on Matrix Analysis and Applications* 13(1):41–45.

Pearson K (1901) Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2(11):559–572.

Probel CJ, Tropp JA (2011) Large-scale PCA with sparsity constraints. Technical report, California Institute of Technology.

Ramana MV (1997) An exact duality theory for semidefinite programming and its complexity implications. *Mathematical Programming* 77(1):129–162.

Reuther A, Kepner J, Byun C, Samsi S, Arcand W, Bestor D, Bergeron B, Gadepally V, Houle M, Hubbell M, Jones M, Klein A, Milechin L, Mullen J, Prout A, Rosa A, Yee C, Michaleas P (2018) Interactive supercomputing on 40,000 cores for machine learning and data analysis. *2018 IEEE High Performance extreme Computing Conference (HPEC)*, 1–6 (IEEE).

Rudin C, Chen C, Chen Z, Huang H, Semenova L, Zhong C (2022) Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys* 16:1–85.

Tan KM, Petersen A, Witten D (2014) Classification of RNA-seq data. *Statistical Analysis of Next Generation Sequencing Data*, 219–246 (Springer).

Tropp JA, Yurtsever A, Udell M, Cevher V (2017) Practical sketching algorithms for low-rank matrix approximation. *SIAM Journal on Matrix Analysis and Applications* 38(4):1454–1485.

Udell M, Horn C, Zadeh R, Boyd S (2016) Generalized low rank models. *Foundations and Trends® in Machine Learning* 9(1):1–118.

Vu VQ, Cho J, Lei J, Rohe K (2013) Fantope projection and selection: A near-optimal convex relaxation of sparse pca. *Advances in Neural Information Processing Systems* 26.

Vu VQ, Lei J (2013) Minimax sparse principal subspace estimation in high dimensions. *The Annals of Statistics* 41(6):2905–2947.

Wang J, Dey SS, Xie Y (2023) Variable selection for kernel two-sample tests. *arXiv preprint arXiv:2302.07415* .

Wei L, Gómez A, Küçükyavuz S (2022) Ideal formulations for constrained convex optimization problems with indicator variables. *Mathematical Programming* 192(1):57–88.

Witten DM, Tibshirani R, Hastie T (2009) A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics* 10(3):515–534.

Yuan XT, Zhang T (2013) Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research* 14(4).

Zou H, Hastie T, Tibshirani R (2006) Sparse principal component analysis. *Journal of Computational and Graphical Statistics* 15(2):265–286.

Supplementary Material

EC.1. Proof of Proposition 1

Proof of Proposition 1   We decompose each matrix $\boldsymbol{Y}^t$ into $\boldsymbol{Y}^t = \boldsymbol{u}_t \boldsymbol{u}_t^\top$ with $\|\boldsymbol{u}_t\|^2 = \mathrm{tr}(\boldsymbol{Y}^t) = 1$. Hence, for any pair (t, t'), $\langle \boldsymbol{Y}^t, \boldsymbol{Y}^{t'} \rangle = \boldsymbol{u}_t^\top \boldsymbol{Y}^{t'} \boldsymbol{u}_t = (\boldsymbol{u}_t^\top \boldsymbol{u}_{t'})^2 \geq 0$.

($\Rightarrow$) If $\boldsymbol{Y} := \sum_{t' \in [r]} \boldsymbol{Y}^{t'} \preceq \mathbb{I}$, then, for any $t \in [r]$, $\boldsymbol{u}_t^\top \boldsymbol{Y} \boldsymbol{u}_t \leq \|\boldsymbol{u}_t\|^2 = 1$. However, $\boldsymbol{u}_t^\top \boldsymbol{Y} \boldsymbol{u}_t = 1 + \sum_{t' \neq t} \langle \boldsymbol{Y}^t, \boldsymbol{Y}^{t'} \rangle$. Hence, for all $t' \neq t$, we must have $\langle \boldsymbol{Y}^t, \boldsymbol{Y}^{t'} \rangle = 0$.

($\Leftarrow$) If $\langle \boldsymbol{Y}^t, \boldsymbol{Y}^{t'} \rangle = 0$ for all $t' \neq t$, then $\{\boldsymbol{u}_t\}_{t \in [r]}$ is an orthonormal family that can be completed to form an orthonormal basis $\{\boldsymbol{u}_t\}_{t \in [p]}$. For any $t \in [r], t' \in [r]$, $\boldsymbol{u}_t^\top \boldsymbol{Y}^{t'} \boldsymbol{u}_t = 1$ if $t = t'$, 0 otherwise so for any $t \in [r]$, $\boldsymbol{u}_t^\top \boldsymbol{Y} \boldsymbol{u}_t \leq \|\boldsymbol{u}_t\|^2$ and $\boldsymbol{Y} = \sum_{t' \in [r]} \boldsymbol{Y}^{t'} \preceq \mathbb{I}$. $\square$

EC.2. Proof of Theorem 2

In this section, we derive the valid inequalities introduced in Theorem 2 from first principles. We proceed in two ways. First, we derive inequalities which hold for each $\boldsymbol{Y}^t$ separately. Second, we observe that these inequalities can be generalized to also apply for $\boldsymbol{Y} = \sum_{t \in [r]} \boldsymbol{Y}^t$, and also derive new inequalities which reflect the interaction of the sparsity and rank constraints.

We repeatedly reference two results on the convex hulls of convex quadratic functions under logical constraints that were established by Wei et al. (2022), building upon the work of Günlük and Linderoth (2010), Atamtürk and Gomez (2025):

LEMMA EC.1. *(Theorem 3 of Wei et al. 2022) The convex closure of the set*

$$\mathcal{S} = \left\{ (\boldsymbol{x}, \boldsymbol{z}, t) \in \mathbb{R}^n \times \{0,1\}^n \times \mathbb{R} : \quad t \geq \sum_{i=1}^n x_i^2, \ \boldsymbol{e}^\top \boldsymbol{z} \leq k, \ x_i = 0 \text{ if } z_i = 0, \ \forall i \in [n] \right\}$$

is given by

$$\mathcal{S}^c = \left\{ (\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\theta}, t) \in \mathbb{R}^n \times [0,1]^n \times \mathbb{R}^n \times \mathbb{R} : \quad t \geq \sum_{i=1}^n \theta_i, \ \boldsymbol{e}^\top \boldsymbol{z} \leq k, \ \theta_i z_i \geq x_i^2 \ \forall i \in [n] \right\}.$$

The above result is sometimes known as a perspective reformulation, since we strengthen the quadratic constraint $t \geq \sum_i x_i^2$ by replacing x_i^2 with its perspective $z_i(x_i/z_i)^2$.

LEMMA EC.2. *(Proposition 4 of Wei et al. 2022) Let $k \geq 2$. Then, the convex closure of the set*

$$\mathcal{T} = \left\{ (\boldsymbol{x}, \boldsymbol{z}, t) \in \mathbb{R}^n \times \{0,1\}^n \times \mathbb{R} : \quad t \geq (\boldsymbol{e}^\top \boldsymbol{x})^2, \ \boldsymbol{e}^\top \boldsymbol{z} \leq k, \ x_i = 0 \text{ if } z_i = 0, \ \forall i \in [n] \right\},$$

is given by

$$\mathcal{T}^c = \left\{ (\boldsymbol{x}, \boldsymbol{z}, t) \in \mathbb{R}^n \times [0,1]^n \times \mathbb{R} : \quad t \cdot \min(1, \boldsymbol{e}^\top \boldsymbol{z}) \geq (\boldsymbol{e}^\top \boldsymbol{x})^2, \ \boldsymbol{e}^\top \boldsymbol{z} \leq k \right\}.$$

Lemma EC.2 is extremely useful when a single continuous variable depends upon multiple indicator variables, as occurs in certain substructures of our reformulations of Problem (3).

Rank-One Valid Inequalities First, inspired by Bertsimas and Cory-Wright (2020), we observe that in a feasible solution to Problem (8) the 2×2 minors of $\boldsymbol{Y}^t$ are certainly non-negative, i.e.,

$$(Y_{i,j}^t)^2 \leq Y_{i,i}^t Y_{j,j}^t, \ \forall i, j \in [p].$$

These constraints are implied by $\boldsymbol{Y}^t \succeq \boldsymbol{0}$ and hence redundant in and of themselves. However, we can sum over all such constraints $i \in [p]$ and use $\mathrm{tr}(\boldsymbol{Y}^t) = 1$, to obtain the constraint

$$\sum_{i \in [p]} (Y_{i,j}^t)^2 \leq Y_{j,j}^t, \ \forall j \in [p].$$

This constraint is a sum of redundant constraints and hence redundant. However, we can strengthen it, by noting that it is a separable convex quadratic inequality under logical constraints. Indeed, by Lemma EC.1, its convex closure under the logical constraints $Y_{i,j}^t = 0$ if $Z_{i,t} = 0$ is given by:

$$\sum_{j=1}^{p} (Y_{i,j}^t)^2 \leq Y_{i,i}^t Z_{i,t}, \ \forall i \in [p], t \in [r]. \tag{EC.1}$$

Rank-r Valid Inequalities In the same spirit as in the rank one case, we can obtain strong valid inequalities by summing the 2×2 minors of $Y_{i,i} = \sum_{t=1}^{r} Y_{i,i}^t$. Indeed, since $\boldsymbol{Y}$ is positive semidefinite, summing its 2×2 minors implies that:

$$\sum_{j=1}^{p} (Y_{i,j})^2 \leq r Y_{i,i}.$$

Moreover, since $Y_{i,j} = \sum_{t=1}^{r} \boldsymbol{Y}_{i,j}^t$ is a rank-one quadratic under logical constraints $Y_{i,j}^t = 0$ if $Z_{i,t} = 0$, invoking Lemma EC.2 reveals that the convex closure of this quadratic constraint under these logical constraints is given by the strengthened inequality:

$$\sum_{j=1}^{p} (Y_{i,j})^2 \leq r Y_{i,i} \min \left(1, \sum_{t=1}^{r} Z_{i,t} \right), \ \forall i \in [p]. \tag{EC.2}$$

Second, in any feasible solution we have:

$$|Y_{i,j}| \leq \sum_{t=1}^{r} |Y_{i,j}^t| = \sum_{t=1}^{r} |U_{i,t}||U_{j,t}|.$$

Let us denote by k_t the sparsity of the tth column of $\boldsymbol{U}$. Then, it is well known that $\|\boldsymbol{U}_t\|_1 \leq \sqrt{k_t}$. Therefore:

$$\sum_{j=1}^{p} |Y_{i,j}| \leq \sum_{j=1}^{p} \left(\sum_{t=1}^{r} |U_{i,t}||U_{j,t}| \right) \leq \sum_{t=1}^{r} \sqrt{k_t} |U_{i,t}|.$$

Next, squaring both sides and invoking the Cauchy-Schwarz inequality reveals that

$$\left(\sum_{j=1}^{p} |Y_{i,j}| \right)^2 \leq \left(\sum_{t=1}^{r} U_{i,t}^2 \right) \left(\sum_{t=1}^{r} k_t \right) = k Y_{i,i}.$$

Finally, noting that the expression $\left(\sum_{j=1}^{p} |Y_{i,j}| \right)^2 \leq k Y_{i,i}$ is a convex quadratic under logical constraints $Y_{i,j}^t = 0$ if $Z_{i,t} = 0$ and invoking Lemma EC.2 to obtain its convex closure yields the

strengthened second-order cone inequality

$$\left(\sum_{j=1}^{p} |Y_{i,j}|\right)^2 \le k Y_{i,i} \min\left(1, \sum_{t\in[r]} Z_{i,t}\right), \ \forall i \in [p], t \in [r]. \tag{EC.3}$$

Third, in the same spirit, the 2×2 minors of $\boldsymbol{Y} \preceq \mathrm{Diag}\left(\min\left(\boldsymbol{e}, \sum_{t\in[r]} \boldsymbol{Z}_t\right)\right)$ are

$$\left(\min\left(1, \sum_{t\in[r]} Z_{i,t}\right) - Y_{i,i}\right)\left(\min\left(1, \sum_{t\in[r]} Z_{j,t}\right) - Y_{j,j}\right) \ge \left(\min\left(1, \sum_{t\in[r]} Z_{i,t}\right)\delta_{i,j} - Y_{i,j}\right)^2,$$

where $\delta_{i,j} = \mathbf{1}\{i = j\}$ is an indicator denoting whether $i = j$. Summing these constraints over all indices $i \neq j$ and using $k - r + 1$ as an upper bound on $\sum_{i\in[p]:i\neq j}\sum_{t\in[r]} Z_{i,t} - Y_{i,i}$ then yields

$$(k - r + 1)\left(\min\left(1, \sum_{t\in[r]} Z_{j,t}\right) - Y_{j,j}\right) \ge \sum_{i\in[p]:i\neq j} Y_{i,j}^2, \ \forall j \in [p].$$

Finally, we recognize the right hand side as a sum of rank-one quadratic terms $(\sum_{t=1}^{r} Y_{i,j}^t)^2$ under logical constraints $Y_{i,j}^t = 0$ if $Z_{j,t} = 0$ and invoke Lemma EC.2 to obtain the convex closure, giving:

$$(k - r + 1)\min\left(1, \sum_{t\in[r]} Z_{j,t}\right)\left(\min\left(1, \sum_{t\in[r]} Z_{j,t}\right) - Y_{j,j}\right) \ge \sum_{i\in[p]:i\neq j} Y_{i,j}^2 \ \forall j \in [p]. \tag{EC.4}$$

The result then follows by introducing a vector $\boldsymbol{w}$ such that w_i models $\min(1, \sum_{t\in[r]} Z_{i,t})$ via $\boldsymbol{w} \in [0,1]^p$, $\boldsymbol{w} \le \boldsymbol{Z}\boldsymbol{e}$, and noting that we can replace $\boldsymbol{Y} \preceq \mathrm{Diag}(\min(\boldsymbol{e}, \boldsymbol{Z}\boldsymbol{e})$ with $\boldsymbol{Y} \preceq \mathrm{Diag}(\boldsymbol{w})$.

EC.3. Proof of Proposition 2

Proof of Proposition 2 First, let us observe that if $\sum_{i\in[p]} Z_{i,t} \le k$ and $\boldsymbol{Y}^t = \boldsymbol{U}_t\boldsymbol{U}_t^\top$ is a rank-one matrix such that $\|\boldsymbol{U}\|_2 = 1$ then we have $\|\boldsymbol{U}\|_1 \le \sqrt{k_t}$ by norm equivalence. Therefore

$$\sum_{j=1}^{p} |Y_{i,j}^t| \le \sum_{j=1}^{p} |U_{i,t}||U_{j,t}| \le \sqrt{k_t}|U_{i,t}|.$$

Squaring both sides of this inequality then yields

$$\left(\sum_{j=1}^{p} |Y_{i,j}^t|\right)^2 \le k_t Y_{i,i}^t,$$

and combining Lemma EC.1 with this inequality yields (10).

Second, in the same spirit, since $\boldsymbol{U}_t\boldsymbol{U}_t^\top$ is only supported on indices where $\boldsymbol{Z}_t$ is non-zero, we have that $\boldsymbol{Y}^t \preceq \mathrm{Diag}(\boldsymbol{Z}_t)$. This constraint implies the following 2×2 minors are non-negative

$$(Z_{i,t} - Y_{i,i}^t)(Z_{j,t} - Y_{j,j}^t) \ge (\delta_{i,j} - Y_{i,j}^t)^2, \ \forall i, j \in [p],$$

where $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise. Summing these inequalities over indices $i \neq j$ and setting $k_t - 1$ as a valid upper bound on $\sum_{i\in[p]:i\neq j} Z_{i,t} - Y_{i,i}^t$ whenever $Z_{j,t} = 1$ (as $Y_{i,j}^t = 0$ if $Z_{j,t} = 0$) gives

$$(k_t - 1)(Z_{j,t} - Y_{j,j}^t) \ge \sum_{i\in[p]:i\neq j} Y_{i,j}^{t\ 2}, \ \forall j \in [p].$$

Finally, using Lemma EC.1 to take the convex closure of this inequality under the logical constraints $Y_{i,j}^t = 0$ if $Z_{j,t} = 0$ gives Equation (11). $\qquad\square$

EC.4. Complete Formulations for the Semidefinite and Second-Order Cone Relaxations

In this section, we provide the complete formulation for our SDP relaxation (12) as well as its second-order cone approximation.

EC.4.1. Semidefinite Relaxation

In Section 2.3, we proposed a semidefinite relaxation, (12), in the case where a sparsity budget for each PC, k_t, is provided. In particular, this relaxation involves valid inequalities that Kim et al. (2022) have derived in the single-PC case. To the best of our knowledge, their formulation leads to the strongest known relaxation for sparse PCA with $r = 1$ which can be solved in polynomial time. We note however that invoking a fixed but sufficiently large level of the sum-of-squares hierarchy may give tighter relaxations, although we do not write these relaxations down as they involve very large semidefinite constraints and are therefore intractable in practice (see also Dey et al. (2022a) for an NP-hard relaxation that uses the ℓ_1 norm).

In (12), we concisely denoted $(\boldsymbol{Y}^t, \boldsymbol{Z}_t) \in \mathcal{T}(k_t)$ the set of valid inequalities involved in Kim et al. (2022)'s "T-relaxation". We now elicit the constraints involved in the set $\mathcal{T}(k_t)$ and provide the complete formulation of the SDP relaxation (12). For each $t \in [r]$, we introduce an additional variable $\boldsymbol{F}^t$ to capture the entry-wise absolute value of $\boldsymbol{Y}^t$, and an additional matrix $\boldsymbol{G}^t$ which contains a sorted version of $\boldsymbol{F}^t$. We obtain:

$$
\max_{\substack{\boldsymbol{Z} \in [0,1]^{p \times r}: \\ \langle \boldsymbol{E}, \boldsymbol{Z} \rangle \leq k, \\ \boldsymbol{w} \in [0,1]^p}} \quad \max_{\substack{\boldsymbol{Y} \in \mathcal{S}_+^p, \boldsymbol{Y}^t, \boldsymbol{F}^t, \boldsymbol{G}^t \in \mathcal{S}_+^p, \\ \boldsymbol{T}^t \in \mathbb{R}_+^{p \times p}, \\ \boldsymbol{r}^{t,D} \in \mathbb{R}^{p-1}, \boldsymbol{t}^{t,D} \in \mathbb{R}_+^{p \times p-1}}} \quad \langle \boldsymbol{Y}, \boldsymbol{\Sigma} \rangle \tag{EC.5}
$$

$$
\text{s.t.} \quad \boldsymbol{Y} \preceq \mathrm{Diag}(\boldsymbol{w}), \ \boldsymbol{Y} = \sum_{t=1}^r \boldsymbol{Y}^t, \ \boldsymbol{w} \leq \boldsymbol{Z}\boldsymbol{e},
$$

$$
\sum_{j=1}^p Y_{i,j}^2 \leq r Y_{i,i} w_i \qquad\qquad \forall i \in [p],
$$

$$
\left(\sum_{j=1}^p |Y_{i,j}| \right)^2 \leq k Y_{i,i} w_i \qquad\qquad \forall i \in [p],
$$

$$
\sum_{i \in [p]: i \neq j} Y_{i,j}^2 \leq (k - r + 1) w_j (w_j - Y_{j,j}) \qquad\qquad \forall j \in [p],
$$

$$
\pm \boldsymbol{Y}^t \leq \boldsymbol{F}^t \qquad\qquad \forall t \in [r],
$$

$$
G_{i,1}^t \geq G_{i,2}^t \geq \ldots \geq G_{i,k_t}^t \qquad\qquad \forall i \in [k_t], \ \forall t \in [r],
$$

$$
G_{i,j}^t = 0 \qquad\qquad \forall i > k_t \text{ or } j > k_t, \ \forall t \in [r],
$$

$$\text{tr}(\boldsymbol{Y}^t) = \text{tr}(\boldsymbol{G}^t) = \text{tr}(\boldsymbol{F}^t) = 1 \qquad \forall t \in [r],$$

$$\langle \boldsymbol{E}, \boldsymbol{G}^t \rangle = \langle \boldsymbol{E}, \boldsymbol{F}^t \rangle \qquad \forall t \in [r],$$

$$\sum_{i=1}^{j} G_{i,i}^t \geq j r_j^{D,t} + \sum_{j=1}^{n} t_{i,j}^{t,D} \qquad \forall j \in [p-1], \ t \in [r],$$

$$Y_{i,i}^t \leq r_j^{t,D} + t_{i,j}^{t,D} \qquad \forall i \in [p], j \in [p-1], t \in [r],$$

$$(F_{i,j}^t)^2 \leq T_{i,j}^t T_{j,i}^t, \ T_{i,i}^t = F_{i,i}^t \qquad \forall i \in [p], j \in [i-1], \ t \in [r],$$

$$\sum_{j \in [p]} T_{i,j}^t = Z_{i,t}, \ \sum_{i \in [p]} T_{i,j}^t = k_t F_{j,j}^t \qquad \forall i \in [p], \forall j \in [p], t \in [r],$$

$$0 \leq T_{i,j}^t \leq F_{i,j}^t \qquad \forall i, j \in [p], t \in [r].$$

The additional variables $\boldsymbol{r}^{t,D}$ and $\boldsymbol{t}^{t,D}$ are introduced to enforce coupling constraints between the diagonal entries of $\boldsymbol{F}^t$ and $\boldsymbol{G}^t$ (Kim et al. 2022, eq. 44), while $\boldsymbol{T}^t$ allows to couple $\boldsymbol{F}^t$ with the binary variables $\boldsymbol{Z}$ (Kim et al. 2022, eq. 50). In contrast to Kim et al. (2022), we explicitly require that each $\boldsymbol{F}^t$ is positive semidefinite (rather than that its 2×2 minors are), in order to obtain a stronger relaxation; we consider the 2×2 minors when developing a more tractable relaxation in the next section.

EC.4.2. A Second-Order Cone Relaxation for Large-Scale Instances

Unfortunately, (12) cannot scale beyond $p = 100$, at least with current technology, due to the presence of multiple semidefinite matrices and constraints.

We now develop a more tractable, albeit less tight, version of the relaxation of (12) which scales to $p > 100$ features. Namely, we replace all semidefinite constraints of the form $\boldsymbol{X} \in \mathcal{S}_+^p$ with the non-negativity of their 2×2 minors, $X_{i,i} X_{j,j} \geq X_{i,j}^2 \ \forall i, j \in [p]$, as presented by Bertsimas and Cory-Wright (2020) and references therein. This gives the following second-order cone relaxation of (12):

$$\max_{\substack{\boldsymbol{Z} \in [0,1]^{p \times r}: \\ \langle \boldsymbol{E}, \boldsymbol{Z} \rangle \leq k, \boldsymbol{w} \in [0,1]^p}} \quad \max_{\substack{\boldsymbol{Y} \in \mathcal{S}^p, \boldsymbol{Y}^t, \boldsymbol{F}_t, \boldsymbol{G}_t \in \mathcal{S}^p, \\ \boldsymbol{T}_t \in \mathbb{R}_+^{p \times p} \ \forall t \in [r], \\ \boldsymbol{r}^{t,D} \in \mathbb{R}^{p-1}, \boldsymbol{t}^{t,D} \in \mathbb{R}_+^{p \times p-1}}} \quad \langle \boldsymbol{Y}, \boldsymbol{\Sigma} \rangle \qquad (\text{EC.6})$$

$$\text{s.t.} \quad \boldsymbol{Y} = \sum_{t=1}^{r} \boldsymbol{Y}^t, \ \text{tr}(\boldsymbol{Y}^t) = 1, \boldsymbol{w} \leq \boldsymbol{Z}\boldsymbol{e} \qquad \forall t \in [r],$$

$$Y_{i,j}^{t^2} \leq Y_{i,i}^t Y_{j,j}^t \qquad \forall i, j \in [p], \forall t \in [r],$$

$$(\delta_{i,j} - Y_{i,j})^2 \leq (w_i - Y_{i,i})(w_j - Y_{j,j}) \qquad \forall i, j \in [p], \forall t \in [r],$$

$$\sum_{j=1}^{p} Y_{i,j}^{2} \leq r Y_{i,i} w_i \qquad \forall i \in [p]$$

$$\left(\sum_{j=1}^{p} |Y_{i,j}| \right)^2 \leq k Y_{i,i} w_i, \ \pm \boldsymbol{Y}^t \preceq \boldsymbol{F}_t \qquad \forall i \in [p], \ \forall t \in [r],$$

$$\sum_{i\in[p]:i\neq j} Y_{i,j}^2 \leq (k-r+1)w_j(w_j - Y_{j,j}) \qquad \forall j \in [p],$$

$$Y_{i,i}^t \leq t_{i,j}^{t,D} + r_j^{t,D} \qquad \forall i \in [p], j \in [p-1], t \in [r],$$

$$\mathrm{tr}(\boldsymbol{Y}^t) = \mathrm{tr}(\boldsymbol{G}^t) = \mathrm{tr}(\boldsymbol{F}_t) = 1 \qquad \forall t \in [r],$$

$$\langle \boldsymbol{E}, \boldsymbol{G}_t - \boldsymbol{F}_t \rangle = 0 \qquad \forall t \in [r],$$

$${G_{i,j}^t}^2 \leq G_{i,i}^t G_{j,j}^t \qquad \forall i,j \in [p], \forall t \in [r],$$

$$G_{i,1}^t \geq G_{i,2}^t \geq \ldots \geq G_{i,k_t}^t \qquad \forall i \in [k_t], \ \forall t \in [r],$$

$$G_{i,j}^t = 0 \qquad \forall i > k_t \text{ or } j > k_t \ \forall t \in [r],$$

$$\sum_{i=1}^{j} G_{i,i}^t \geq j r_j^{D,t} + \sum_{j=1}^{n} t_{i,j}^{t,D} \qquad \forall j \in [p-1], \ t \in [r],$$

$${F_{i,j}^t}^2 \leq T_{i,j}^t T_{j,i}^t, \ T_{i,i}^t = F_{i,i}^t \qquad \forall i \in [p], j \in [i-1], \ t \in [r],$$

$$\sum_{j\in[p]} T_{i,j}^t = Z_{i,t}, \ \sum_{i\in[p]} T_{i,j}^t = k_t F_{j,j}^t \qquad \forall i \in [p], \forall j \in [p], t \in [r],$$

$$0 \leq T_{i,j}^t \leq F_{i,j}^t, {F_{i,j}^t}^2 \leq F_{i,i}^t F_{j,j}^t \qquad \forall i,j \in [p], \forall t \in [r].$$

EC.5. Proof of Theorem 3

We prove that Problem (13) is NP-hard via a reduction from a variant of the exact 3-set cover problem.

Restricted Exact Cover by 3-Sets Problem (X3C-R) Consider a set X with $|X| = 3q$ and a family $\mathcal{C} = \{S_1, \ldots, S_m\}$ of 3-element subsets of X of size $m := |\mathcal{C}|$. Further, assume that for any pair of sets $S, S' \in \mathcal{C}$, $|S \cap S'| \leq 1$. Then, X3C-R corresponds to asking whether there exists an exact cover of X, namely a subfamily $\mathcal{C}' \subseteq \mathcal{C}$ of size q, whose sets are pairwise disjoint and whose union is X.

Problem X3C-R is NP-complete, as we prove in Proposition EC.2. Leveraging that result, we now prove the following proposition, which implies Theorem 3.

PROPOSITION EC.1. *For any instance of X3C-R, we can construct integers (p,r), a binary mask $\hat{\boldsymbol{Z}} \in \{0,1\}^{p \times r}$, a symmetric matrix $\boldsymbol{\Sigma} \in \mathcal{S}^p$, and a scalar τ, such that X3C-R is equivalent to finding a feasible solution to (13) with objective value at least τ. Moreover, $p = |X| + |\mathcal{C}| = 3q + m$, $r = |\mathcal{C}|$, $\tau = 2q$ are all polynomial in the size of X3C-R.*

REMARK EC.1. Given the orthogonality constraints in (13), we can replace the matrix $\boldsymbol{\Sigma}$ in Problem (13) by $\boldsymbol{\Sigma} + \beta \boldsymbol{I}_p$ for any $\beta > 0$ and obtain the same optimal solution (it only shifts the objective value by βr). In particular, if $\boldsymbol{\Sigma} \in \mathcal{S}^p$ in Problem (13), we can assume $\boldsymbol{\Sigma} \succeq \boldsymbol{0}$ without loss of generality.

Proof of Proposition EC.1 Consider an instance of X3C-R and let us construct an instance of (13).

Let us first define the coordinate space: We create one coordinate for each element $x \in X$, and one "dummy" coordinate d_t for each 3-set $S_t \in \mathcal{C}$. Hence, $p = |X| + |\mathcal{C}| = 3q + m$.

We consider one PC per 3-set. Hence, $r = |\mathcal{C}| = m$. Each column $t \in [r]$ is allowed to be nonzero only on the coordinates corresponding to $x \in S_t$ and on the dummy d_t. In other words,

$$\hat{Z}_{i,t} = 1, \quad \text{if } i \in S_t \cup \{d_t\},$$

$$\hat{Z}_{i,t} = 0, \quad \text{otherwise.}$$

Note that $r \leq m$ and Problem (13) is always feasible (a feasible solution is obtained by setting $\boldsymbol{u}_t$ equal to 1 on its dummy coordinate d_t and 0 elsewhere).

Finally, the matrix $\boldsymbol{\Sigma}$ is non-zero on the block indexed by $X \times X$ only, i.e.,

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}, \quad \text{with} \quad A_{xy} = \begin{cases} 1, & \text{if there exists } S \in \mathcal{C} \text{ with } x, y \in S, \\ 0, & \text{otherwise.} \end{cases}$$

Consider one column $t \in [r]$. It is associated with one 3-set S_t. On this set, the submatrix of $\boldsymbol{A}$ is equal to

$$\boldsymbol{A}_{S_t, S_t} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} = \boldsymbol{e}\boldsymbol{e}^\top - \boldsymbol{I}_3.$$

So, for any non-zero vector $\boldsymbol{z} \in \mathbb{R}^3$, we have by Cauchy-Schwarz

$$\boldsymbol{z}^\top \boldsymbol{A}_{S_t, S_t} \boldsymbol{z} = (\boldsymbol{e}^\top \boldsymbol{z})^2 - \|\boldsymbol{z}\|^2 \leq (\|\boldsymbol{z}\|_0 - 1)\|\boldsymbol{z}\|^2, \tag{EC.7}$$

with $\|z\|_0 = |\{j : z_j \neq 0\}|$.

Set $\tau = 2q$. Let us now show that there exists a feasible solution to (13) with objective value at least τ iff there is an affirmative answer to X3C-R.

Consider a solution to X3C-R, $\mathcal{C}'$. For each set $S_t \in \mathcal{C}'$, define $\boldsymbol{u}_t$ as equal to $\frac{1}{\sqrt{3}}(1, 1, 1)$ on the coordinates indexed by S_t and 0 on the dummy coordinate. Alternatively, for each set $S_t \notin \mathcal{C}'$, set $\boldsymbol{u}_t$ equal to $\boldsymbol{0}$ on the coordinates indexed by S_t and 1 on the dummy coordinate. By construction, because each element $x \in X$ is covered by exactly one 3-set of $\mathcal{C}'$, for any $t \neq t'$, the columns $\boldsymbol{u}_t$ and $\boldsymbol{u}_{t'}$ have disjoint supports and $\boldsymbol{u}_t^\top \boldsymbol{u}_{t'} = \delta_{t,t'}$. Second,

$$\sum_{t \in [r]} \boldsymbol{u}_t^\top \boldsymbol{\Sigma} \boldsymbol{u}_t = \frac{1}{3} \sum_{S_t \in \mathcal{C}'} \boldsymbol{e}^\top \boldsymbol{A}_{S_t, S_t} \boldsymbol{e} = 2q = \tau.$$

Conversely, assume there exists a solution to (13) with objective value at least τ. For any $x \in X$, there exists at most one column $\boldsymbol{u}_t$ such that $u_{t,x} \neq 0$. Indeed, by contradiction, assume there exist two distinct columns t, t' such that $u_{t,x} \neq 0$ and $u_{t',x} \neq 0$. Then, $x \in S_t \cap S_{t'}$ so we must have $\{x\} = S_t \cap S_{t'}$ by assumption. In this case, the constraint $\boldsymbol{u}_t^\top \boldsymbol{u}_{t'} = 0$ reads $u_{t,x} u_{t',x} = 0$ and forces one of the coefficients $u_{t,x}$ or $u_{t',x}$ to be equal to 0, which is a contradiction. We say that a column of $\boldsymbol{U}$, $\boldsymbol{u}_t$, is s-heavy ($s \in \{0, 1, 2, 3\}$) if s of its coordinates indexed by X are non-zeros. Let n_s

denote the number of s-heavy columns. From our earlier reasoning, each $x \in X$ is covered by at most one column of $\boldsymbol{U}$ so we must have $3n_3 + 2n_2 \leq 3n_3 + 2n_2 + n_1 \leq 3q$. The solution $\boldsymbol{U}$ achieves an objective value at least equal to $\tau = 2q$ so (EC.7) implies $2q = \tau \leq 2n_3 + n_2$. However,

$$2n_3 + n_2 = \frac{2}{3}(3n_3 + \frac{3}{2}n_2) \leq \frac{2}{3}(3n_3 + 2n_2) \leq 2q,$$

so $2q = 2n_3 + n_2$ and the inequalities above are tight, i.e., $n_2 = 0$ and $n_3 = q$. Consequently, $n_1 = 0$ as well. In conclusion, define $\mathcal{C}' = \{S_t : \boldsymbol{u}_t \text{ is 3-heavy}\}$. For any $x \in X$, there exists at most one column $\boldsymbol{u}_t$ such that $u_{t,x} \neq 0$, so the sets in $\mathcal{C}'$ are pairwise disjoint. Furthermore, $|\mathcal{C}'| = n_3 = q = |X|/3$ so $\mathcal{C}'$ must cover the entire set X. $\qquad\square$

REMARK EC.2. Observe that the proof of Proposition EC.1 shows that there exists a solution to X3C-R iff there exists a feasible solution to Problem (13) with objective value $\tau - \eta$ with $\tau = 2q > 2p/3$ being the optimal value and $\eta \in [0,1)$. Hence, it shows that it is NP-hard to approximate Problem (13) within a factor $1 - \epsilon$ with $\epsilon = 1/\tau < 3/(2p)$.

EC.5.1. X3C-R is NP-Complete

To the best of our knowledge, there is no existing proof of the complexity of X3C-R. Problem X3C-R is a restriction of the classical exact cover by 3-sets problem (X3C), which is known to be NP-complete (Garey and Johnson 1979, chapter A3.1).

Exact Cover by 3-Sets Problem (X3C) Consider a set X with $|X| = 3q$ and a family $\mathcal{C} = \{S_1, \ldots, S_m\}$ of 3-element subsets of X, of size $m := |\mathcal{C}|$. Does there exist a subfamily $\mathcal{C}' \subseteq \mathcal{C}$ of size q whose sets are pairwise disjoint and whose union is X?

Formally, we have the following result:

PROPOSITION EC.2. *Any instance of X3C can be reduced to an instance of X3C-R in polynomial time, and hence Problem X3C-R is NP-hard.*

Any instance of X3C-R is definitionally an instance of X3C, so Proposition EC.2 implies that X3C-R is NP-complete.

Proof of Proposition EC.2   We consider an instance $(X, \mathcal{C})$ of X3C. Define the set

$$Y = X \cup \{z_{S,t} : S \in \mathcal{C}, \ t \in \{1, \ldots, 6\}\}.$$

So, $|Y| = |X| + 6|\mathcal{C}| = 3q + 6m$.

For any 3-set $C = \{x_1, x_2, x_3\} \in \mathcal{C}$ (the indexing is arbitrary and fixed), define the following sets

$$
\begin{aligned}
D_{S,1} &= \{x_1 & & z_{S,1} & & z_{S,4} & & \}, \\
D_{S,2} &= \{ & x_2 & & z_{S,2} & & z_{S,5} & \}, \\
D_{S,3} &= \{ & & x_3 & & z_{S,3} & & z_{S,6} \}, \\
D_{S,4} &= \{ & & z_{S,1} \ z_{S,2} \ z_{S,3} & & & & \}, \\
D_{S,5} &= \{ & & & & z_{S,4} \ z_{S,5} \ z_{S,6} & & \},
\end{aligned}
$$

and define $\mathcal{D} = \{D_{S,t} : S \in \mathcal{C}, \ t \in \{1, \ldots, 5\}\}$. Hence, $|\mathcal{D}| = 5|\mathcal{C}| = 5m$. By construction, for any pair of sets $D, D' \in \mathcal{D}$, we must have $|D \cap D'| \leq 1$. In other words, $(Y, \mathcal{D})$ is a valid X3C-R instance.

Let us now show that there exists an exact cover of $(X, \mathcal{C})$ iff there exists an exact cover of $(Y, \mathcal{D})$.

Let $\mathcal{C}'$ be an exact cover of $(X, \mathcal{C})$. Define $\mathcal{D}' = \{D_{S,1}, D_{S,2}, D_{S,3} : S \in \mathcal{C}'\} \cup \{D_{S,4}, D_{S,5} : S \in \mathcal{C} \setminus \mathcal{C}'\}$. We have $|\mathcal{D}'| = 3|\mathcal{C}'| + 2|\mathcal{C} \setminus \mathcal{C}'| = |\mathcal{C}'| + 2|\mathcal{C}| = q + 2m = |Y|/3$. Because $\mathcal{C}'$ is a cover, for any $x \in X$, there exists $S \in \mathcal{C}'$ such that $x \in S$. So there exists $t \in \{1, 2, 3\}$ such that $x \in D_{S,t}$ and $D_{S,t} \in \mathcal{D}'$. For any $z_{S,t}, S \in \mathcal{C}, t \in \{1, \ldots, 6\}$, either $S \in \mathcal{C}'$ and there must exists $t' \in \{1, 2, 3\}$ such that $z_{S,t} \in D_{S,t'}$ and $D_{S,t'} \in \mathcal{D}'$, or $S \notin \mathcal{C}'$ and there must exists $t' \in \{4, 5\}$ such that $z_{S,t} \in D_{S,t'}$ and $D_{S,t'} \in \mathcal{D}'$. Altogether, we showed that any element of Y belongs to an element of $\mathcal{D}'$ (i.e., $\mathcal{D}'$ is a cover). Because elements of $\mathcal{D}'$ are 3-sets and $|\mathcal{D}'| = |Y|/3$, $\mathcal{D}'$ is an exact cover.

Let $\mathcal{D}'$ be an exact cover of $(Y, \mathcal{D})$ (in particular, $|\mathcal{D}'| = q + 2m$). Define $\mathcal{C}' = \{S \in \mathcal{C} : \exists t \in \{1, 2, 3\}, D_{S,t} \in \mathcal{D}'\}$. For any $x \in X$, $x \in Y$ and $\mathcal{D}'$ is a cover, so there exist $S \in \mathcal{C}, t \in \{1, 2, 3\}$ such that $x \in D_{S,t}$ and $D_{S,t} \in \mathcal{C}'$. Hence, $\mathcal{C}'$ covers X. Let us consider a set $S \in \mathcal{C}$ such that $D_{S,1} \in \mathcal{D}'$, we must have $D_{S,2}, D_{S,3} \in \mathcal{D}'$. Indeed, we must have $D_{S,4}, D_{S,5} \notin \mathcal{D}'$, because they both overlap with $D_{S,1}$ and elements of $\mathcal{D}'$ are pairwise disjoint. The variable $z_{S,2}$ is covered by $\mathcal{D}'$ so necessarily, we must have $D_{S,2} \in \mathcal{D}'$. Similarly for $z_{S,3}$. Hence, $3|\mathcal{C}'| + 2|\mathcal{C} \setminus \mathcal{C}'| = |\mathcal{C}'| + 2m$. Therefore, $|\mathcal{C}'| = q = |X|/3$ and $\mathcal{C}'$ is an exact cover of X. $\qquad \square$

EC.6. Asymptotic Feasibility of the Solutions Returned by Algorithm 2

In this section, we analyze the asymptotic feasibility of solutions generated by Algorithm 2 and show how they depend on the method used to solve each subproblem. First, we prove that Algorithm 2 generates a subsequence of solutions that are asymptotically feasible, provided that each subproblem in Algorithm 2 is solved to near-optimality (Section EC.6.1). We acknowledge that this assumption may not hold for the truncated power method of Yuan and Zhang (2013) that we use in our implementation (Section EC.6.2). Secondly, under less restrictive assumptions that are satisfied by the truncated power method, we show convergence to stationary points of the function $\boldsymbol{U} \mapsto \sum_{t' > t}(\boldsymbol{U}_t^\top \boldsymbol{U}_{t'})^2$ over the feasible region. Unfortunately, this result cannot rule out convergence to infeasible solutions, which is a known limitation of the quadratic penalty method, even when subproblems are solved with increasing accuracy (see Nocedal and Wright 2006, Theorem 17.2).

We introduce some compact notation from Nocedal and Wright (2006, chapter 17). Letting $\mathbb{S}_k^p$ denote the set of k-sparse vectors from the p-dimensional unit sphere, we rewrite Problem (3) as a generic-equality constrained maximization problem of the form

$$\max_{\boldsymbol{U} \in \mathcal{U}} f(\boldsymbol{U}) \quad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{U}) = \boldsymbol{0},$$

where the feasible set $\mathcal{U} := \{\boldsymbol{U} \in \mathbb{R}^{p \times r} : \forall t \in [r], \ \boldsymbol{U}_t \in \mathbb{S}_{k_t}^p\}$ is non-convex, the objective function f is defined as $f(\boldsymbol{U}) = \sum_{t \in [r]} \boldsymbol{U}_t^\top \boldsymbol{\Sigma} \boldsymbol{U}_t$, and the constraint function $\boldsymbol{c} : \mathbb{R}^{n \times r} \to \mathbb{R}^{r(r-1)/2}$ is defined as $c_{t,t'}(\boldsymbol{U}) = \boldsymbol{U}_t^\top \boldsymbol{U}_{t'}$ for $t > t'$.

EC.6.1. Asymptotic Convergence of Algorithm 2 With Branch-and-Bound for Subproblems

Suppose that, at each iteration ℓ of Algorithm 2, the vector $\boldsymbol{U}_t^{(\ell)}$ is computed by solving

$$\max_{\boldsymbol{u} \in \mathbb{S}_{k_t}^p} \left\langle \boldsymbol{u}\boldsymbol{u}^\top, \boldsymbol{\Sigma} - \sum_{t' \in [r]: t' \neq t} \lambda_{t,t'}^{(\ell)} \boldsymbol{U}_{t'}^{(\ell-1)} \boldsymbol{U}_{t'}^{(\ell-1)\top} \right\rangle. \tag{EC.8}$$

Under this assumption, we now establish the asymptotic feasibility of a subsequence of the solutions generated by Algorithm 2, provided the subproblems (EC.8) are solved to finite (but not necessarily improving) accuracy, and $k_t \geq r$ for each $t \in [r]$.

REMARK EC.3. The subproblem (EC.8) for updating column t at iteration ℓ corresponds to a block-Jacobi update rule where the penalty on orthogonality violation is expressed using the value of $\boldsymbol{U}$ at the previous iteration $\ell - 1$. In particular, it requires storing both $\boldsymbol{U}^{(\ell)}$ and $\boldsymbol{U}^{(\ell-1)}$ and does not use the updated values of $\boldsymbol{U}_{t'}^{(\ell)}$ for $t' < t$. Instead, we can implement our algorithm (and adapt our analysis) using Gauss-Seidel updates instead, which would involve subproblems of the form

$$\max_{\boldsymbol{u} \in \mathbb{S}_{k_t}^p} \left\langle \boldsymbol{u}\boldsymbol{u}^\top, \boldsymbol{\Sigma} - \sum_{t' < t} \lambda_{t,t'}^{(\ell)} \boldsymbol{U}_{t'}^{(\ell)} \boldsymbol{U}_{t'}^{(\ell)\top} - \sum_{t' > t} \lambda_{t,t'}^{(\ell)} \boldsymbol{U}_{t'}^{(\ell-1)} \boldsymbol{U}_{t'}^{(\ell-1)\top} \right\rangle.$$

PROPOSITION EC.3. *Assume that $k_t \geq r$, for all $t \in [r]$. Consider a sequence of penalty parameters $\{\lambda_{t,t'}^{(\ell)}\}$ with $\mu_\ell := \min_{t,t' \in [r]} \lambda_{t,t'}^{(\ell)} \to +\infty$ and a sequence of iterates $\{\boldsymbol{U}^{(\ell)}\}$ such that, for any $\ell \geq 1$, any $t \in [r]$, and any sparse vector $\boldsymbol{u} \in \mathbb{S}_{k_t}^p$,*

$$\boldsymbol{U}_t^{(\ell)\top} \boldsymbol{\Sigma} \boldsymbol{U}_t^{(\ell)} - \sum_{t' \neq t} \lambda_{t,t'}^{(\ell)} (\boldsymbol{U}_t^{(\ell)\top} \boldsymbol{U}_{t'}^{(\ell-1)})^2 \geq \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} - \sum_{t' \neq t} \lambda_{t,t'}^{(\ell)} (\boldsymbol{u}^\top \boldsymbol{U}_{t'}^{(\ell-1)})^2 - \epsilon_\ell, \tag{EC.9}$$

with $\epsilon_\ell / \mu_\ell \to 0$. Then, for any convergent subsequence of $\{\boldsymbol{U}^\ell\}$, its limit point $\bar{\boldsymbol{U}}$ is feasible.

REMARK EC.4. Because $\boldsymbol{U}^{(\ell)} \in \mathcal{U}$, which is compact, a converging subsequence exists by the Bolzano-Weierstrass theorem.

When using a provably optimal method, like the branch-and-bound algorithm of Berk and Bertsimas (2019), for solving the subproblems (EC.8), we can set the suboptimality gap ϵ_ℓ explicitly as the termination criterion and ensure the condition of Proposition EC.3 holds (e.g., by keeping ϵ_ℓ constant). However, empirically, solutions from the truncated power method (TPM) are very close to the optimal solution, but we cannot guarantee that TPM solves (EC.8) to ϵ_ℓ-optimality for some controllable $\epsilon_\ell > 0$.

Proof of Proposition EC.3 Consider the ℓth iteration and fix $t \in [r]$. Since $k_t \geq r$, we can construct $\boldsymbol{u} \in \mathbb{S}_{k_t}^p$ such that $\boldsymbol{u}^\top \boldsymbol{U}_{t'}^{(\ell-1)} = 0, \forall t' \neq t$ (e.g., select r coordinates, the restriction of the $r-1$ vectors $\boldsymbol{U}_{t'}^{(\ell-1)}, t' \neq t$ to these r coordinates cannot span the entire space $\mathbb{R}^r$ so we can find a unit vector orthogonal to all of them, which we can view as an r-sparse unit vector in $\mathbb{R}^p$ by padding with zeros). For this vector, we have, by assumption

$$\boldsymbol{U}_t^{(\ell)\top} \boldsymbol{\Sigma} \boldsymbol{U}_t^{(\ell)} - \sum_{t' \neq t} \lambda_{t,t'}^{(\ell)} (\boldsymbol{U}_t^{(\ell)\top} \boldsymbol{U}_{t'}^{(\ell-1)})^2 \geq \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} - \epsilon_\ell,$$

leading to

$$\sum_{t' \neq t} (\boldsymbol{U}_t^{(\ell)\top} \boldsymbol{U}_{t'}^{(\ell-1)})^2 \leq \frac{1}{\mu_\ell} \left(\boldsymbol{U}_t^{(\ell)\top} \boldsymbol{\Sigma} \boldsymbol{U}_t^{(\ell)} - \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} \right) + \frac{\epsilon_\ell}{\mu_\ell} \leq \frac{\lambda_{\max}(\boldsymbol{\Sigma}) + \epsilon_\ell}{\mu_\ell}.$$

Taking limits when $\ell \to \infty$, we get $\sum_{t' \neq t} (\bar{\boldsymbol{U}}_t^\top \bar{\boldsymbol{U}}_{t'})^2 = 0$, hence $\bar{\boldsymbol{U}}_t^\top \bar{\boldsymbol{U}}_{t'} = 0, \ \forall t \neq t'$. □

EC.6.2. Asymptotic Stationarity of Algorithm 2 With Truncated Power Method for Subproblems

In this section, we analyze the convergence of Algorithm 2 when the subproblems (EC.8) are solved using the truncated power method of Yuan and Zhang (2013). In particular, we leverage the following property, which follows immediately from the description of the method in Yuan and Zhang (2013):

LEMMA EC.3. *Consider a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{p \times p}$, $k \leq p$, and the sparse quadratic problem*

$$\max_{\boldsymbol{x} \in \mathbb{R}^p} \quad \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x} \ s.t. \ \boldsymbol{x} \in \mathbb{S}_k^p, \tag{EC.10}$$

The truncated power method of Yuan and Zhang (2013) applied to (EC.10) converges to a sparse eigenvector of $\boldsymbol{A}$, i.e., a vector $\bar{\boldsymbol{x}} \in \mathbb{R}^p : \|\boldsymbol{x}\|_2 = 1$, such that the support of $\bar{\boldsymbol{x}}$, $S := \{j : \bar{x}_j \neq 0\}$ is of size at most k and such that there exists $\lambda \in \mathbb{R}$ such that $\boldsymbol{A}_{S,S} \bar{\boldsymbol{x}}_S = \lambda \bar{\boldsymbol{x}}_S$.

For any $\boldsymbol{x} \in \mathbb{S}_k^p$, we denote $\mathcal{T}_{\boldsymbol{x}}(\mathbb{S}_k^p)$ the tangent cone to $\mathbb{S}_k^p$ at $\boldsymbol{x}$, and $\text{Proj}_{\boldsymbol{x}}$ the projection onto $\mathcal{T}_{\boldsymbol{x}}(\mathbb{S}_k^p)$. In particular, we have

$$\mathcal{T}_{\boldsymbol{x}}(\mathbb{S}_k^p) = \{\boldsymbol{y} \in \mathbb{R}^p : \text{supp}(\boldsymbol{y}) \subseteq \text{supp}(\boldsymbol{x}), \ \boldsymbol{y}^\top \boldsymbol{x} = 0\}.$$

With these notations, the equation $\boldsymbol{A}_{S,S} \bar{\boldsymbol{x}}_S = \lambda \bar{\boldsymbol{x}}_S$ in Lemma EC.3 implies

$$\boldsymbol{y}^\top \boldsymbol{A} \bar{\boldsymbol{x}} = 0, \quad \forall \boldsymbol{y} \in \mathcal{T}_{\boldsymbol{x}}(\mathbb{S}_k^p), \quad \text{and} \quad \|\text{Proj}_{\boldsymbol{x}}(\boldsymbol{A} \bar{\boldsymbol{x}})\| = \boldsymbol{0}.$$

At iteration ℓ of Algorithm 2, the vector $\boldsymbol{U}_t^\ell$ is computed by solving a problem of the form (EC.10) with $\boldsymbol{A} = \boldsymbol{\Sigma} - \sum_{t' \neq t} \lambda_{t,t'}^{(\ell)} \boldsymbol{U}_{t'}^{(\ell-1)} \boldsymbol{U}_{t'}^{(\ell-1)\top}$ using the truncated power method. Hence, Lemma EC.3 yields

$$\left\| \text{Proj}_{\boldsymbol{U}_t^{(\ell)}} \left(\boldsymbol{\Sigma} \boldsymbol{U}_t^{(\ell)} - \sum_{t' \neq t} \lambda_{t,t'}^{(\ell)} (\boldsymbol{U}_t^{(\ell)\top} \boldsymbol{U}_{t'}^{(\ell-1)}) \boldsymbol{U}_{t'}^{(\ell-1)} \right) \right\| = 0.$$

Thanks to this condition, we can show that the sequence of iterates generated by Algorithm 2 converges to a stationary point of the quadratic penalty function. Unfortunately, stationarity does not imply feasibility and Algorithm 2 can converge to an infeasible, yet stationary, point. This behavior is a known limitation of the quadratic penalty method, which occurs even when subproblems are solved to increasing accuracy (see Nocedal and Wright 2006, Theorem 17.2).

Formally, we have the following result:

PROPOSITION EC.4. *Consider a sequence of homogeneous penalty parameters $\{\lambda_{t,t'}^{(\ell)}\} = \{\mu_\ell\}$ with $\mu_\ell \to +\infty$ and a sequence of iterates $\{U^{(\ell)}\}$ such that*

$$\max_{t \in [r]} \left\| \mathrm{Proj}_{U_t^{(\ell)}} \left(\Sigma U_t^{(\ell)} - \mu_\ell \sum_{t' \neq t} (U_t^{(\ell)\top} U_{t'}^{(\ell-1)}) U_{t'}^{(\ell-1)} \right) \right\| \leq \tau_\ell,$$

with $\tau_\ell/\mu_\ell \to 0$. Then, for any converging subsequence of $\{U^\ell\}$ such that the sequence $\{\mathrm{supp}(U_t^{(\ell)})\}, t \in [r]$ converges, its limit point $\bar{U}$ is a stationary point of the function $U \mapsto \|c(U)\|^2$ on $\mathcal{U}$, namely

$$\mathrm{Proj}_{\bar{u}_t} \left(\sum_{t' \neq t} (\bar{u}_t^\top \bar{u}_{t'}) \bar{u}_{t'} \right) = \mathbf{0}.$$

Proposition EC.4 asserts that limit points are stationary points of $U \mapsto \|c(\bar{U})\|^2$ over the feasible set, meaning that for each column t of a limit point $\bar{U}$, the directional derivative of $\|c(\bar{U})\|^2$ vanishes along every direction in $\mathcal{T}_{\bar{U}_t}(\mathbb{S}_{k_t}^p)$. To see that stationarity does not imply feasibility, observe, for example, that with $r = 2$, Proposition EC.4 does not rule out convergence to a matrix $U = [u_1, u_1] \in \mathbb{R}^{p \times 2}$ with $u_1 \in \mathbb{S}_k^p$. Provided the truncated power method is run until convergence at each iteration, Algorithm 2 satisfies the assumptions of Proposition EC.4 with $\tau_\ell = 0$. Nonetheless, Proposition EC.4 shows convergence to a stationary point even when this condition is approximately satisfied.

Proof of Proposition EC.4   We follow the proof steps of Nocedal and Wright (2006, Theorem 17.2). Denoting $w_t^{(\ell)} := \Sigma U_t^{(\ell)} - \mu_\ell \sum_{t' \neq t} (U_t^{(\ell)\top} U_{t'}^{(\ell-1)}) U_{t'}^{(\ell-1)}$, we have

$$\mathrm{Proj}_{U_t^{(\ell)}} \left(\sum_{t' \neq t} (U_t^{(\ell)\top} U_{t'}^{(\ell-1)}) U_{t'}^{(\ell-1)} \right) = \frac{1}{\mu_\ell} \mathrm{Proj}_{U_t^{(\ell)}} \left(\Sigma U_t^{(\ell)} - w_t^{(\ell)} \right)$$

and our assumption leads to

$$\left\| \mathrm{Proj}_{U_t^{(\ell)}} \left(\sum_{t' \neq t} (U_t^{(\ell)\top} U_{t'}^{(\ell-1)}) U_{t'}^{(\ell-1)} \right) \right\| \leq \frac{1}{\mu_\ell} \left(\| \mathrm{Proj}_{U_t^{(\ell)}} \left(\Sigma U_t^{(\ell)} \right) \| + \| \mathrm{Proj}_{U_t^{(\ell)}} \left(w_t^{(\ell)} \right) \| \right)$$

$$\leq \frac{1}{\mu_\ell} \left(\lambda_{\max}(\Sigma) + \tau_\ell \right)$$

Let us consider a converging subsequence of $\{U^\ell\}$ such that the sequences $\{\mathrm{supp}(U_t^{(\ell)})\}$ for $t \in [r]$ converge. Since the sequences $\{\mathrm{supp}(U_t^{(\ell)})\}$ take discrete values and converge, there exists ℓ_0 such that, for any $\ell \geq \ell_0$, $t \in [r]$, $\mathrm{supp}(U_t^{(\ell)}) = \mathrm{supp}(\bar{U}_t)$. Hence, for any $\ell \geq \ell_0$, $\mathrm{Proj}_{U_t^{(\ell)}}(\cdot) \to \mathrm{Proj}_{\bar{U}_t}(\cdot)$ (for $\ell \geq \ell_0$, the projection operators only differ in the linear constraint defining the tangent cone) and taking the limit in the inequality above, for each $t \in [r]$,

$$\mathrm{Proj}_{\bar{U}_t} \left(\sum_{t' \neq t} (\bar{U}_t^\top \bar{U}_{t'}) \bar{U}_{t'} \right) = \mathbf{0}. \quad \square$$

REMARK EC.5. At the expense of more complicated notation, Proposition EC.4 can be extended to non-homogeneous penalty sequences $\{\lambda_{t,t'}^{(\ell)}\}$ such that $\mu_\ell := \min_{t,t'} \lambda_{t,t'}^{(\ell)} \to +\infty$ and such that, for

each pair (t, t'), there exists $\lambda_{t,t'} > 0$ s.t. $\lambda_{t,t'}^{(\ell)}/\mu_\ell \to \lambda_{t,t'}$ (i.e., all $\lambda_{t,t'}^{(\ell)}$s grow at the same rate), in which case we have convergence to a stationary point of the function $\boldsymbol{U} \mapsto \sum_{t>t'} \lambda_{t,t'} (\boldsymbol{U}_t^\top \boldsymbol{U}_{t'})^2$.

EC.7. Alternative Proof of Theorem 4

For a fixed $\boldsymbol{Z}$, the objective value in

$$\max_{\boldsymbol{U} \in \mathbb{R}^{p \times r}} \quad \langle \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{\Sigma} \rangle \quad \text{s.t.} \quad \boldsymbol{U}^\top \boldsymbol{U} = \mathbb{I}, \, U_{i,t} = 0 \text{ if } Z_{i,t} = 0, \, \forall i \in [p], \, \forall t \in [r],$$

is equal to

$$\langle \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{\Sigma} \rangle = \sum_{t \in [r]} \langle \boldsymbol{U}_t \boldsymbol{U}_t^\top, \boldsymbol{\Sigma} \rangle = \sum_{t \in [r]} \langle \boldsymbol{U}_t \boldsymbol{U}_t^\top, \mathrm{Diag}(\boldsymbol{Z}_t) \boldsymbol{\Sigma} \mathrm{Diag}(\boldsymbol{Z}_t) \rangle = \mathrm{vec}(\boldsymbol{U})^\top \boldsymbol{S} \, \mathrm{vec}(\boldsymbol{U}),$$

with

$$\boldsymbol{S} := \begin{pmatrix} \mathrm{Diag}(\boldsymbol{Z}_1) \boldsymbol{\Sigma} \mathrm{Diag}(\boldsymbol{Z}_1) & & \\ & \ddots & \\ & & \mathrm{Diag}(\boldsymbol{Z}_r) \boldsymbol{\Sigma} \mathrm{Diag}(\boldsymbol{Z}_r) \end{pmatrix},$$

and $\mathrm{vec}(\boldsymbol{U}) \in \mathbb{R}^{pr}$ the vector obtained by concatenating the columns of $\boldsymbol{U}$, $\boldsymbol{u}_t$, vertically. As in the proof of (18), we use the fact that $\boldsymbol{S} \preceq \mathrm{Diag}(\boldsymbol{s})$ with $\boldsymbol{s} \in \mathbb{R}^{pr}$ the vector of absolute row-sums of $\boldsymbol{S}$, i.e., $s_{(t-1) \times p + i} = \sum_{j \in [p]} Z_{i,t} Z_{j,t} |\Sigma_{i,j}|$ for $i \in [p]$, $t \in [r]$. This leads to

$$\langle \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{\Sigma} \rangle \leq \mathrm{vec}(\boldsymbol{U})^\top \mathrm{Diag}(\boldsymbol{s}) \, \mathrm{vec}(\boldsymbol{U}) = \sum_{t \in [r]} \sum_{i \in [p]} U_{i,t}^2 \left(\sum_{j \in [p]} Z_{i,t} Z_{j,t} |\Sigma_{i,j}| \right).$$

Hence, we can bound the objective value by

$$\max_{\boldsymbol{U} \in \mathbb{R}^{p \times r}} \quad \sum_{t \in [r]} \sum_{i \in [p]} U_{i,t}^2 \left(\sum_{j \in [p]} Z_{i,t} Z_{j,t} |\Sigma_{i,j}| \right) \quad \text{s.t.} \quad \boldsymbol{U}^\top \boldsymbol{U} = \mathbb{I}.$$

On one side, the orthogonality constraint implies $\|\boldsymbol{U}_t\|_2^2 = \sum_{i \in [p]} U_{i,t}^2 = 1$, for any $t \in [r]$. On the other side, $\boldsymbol{U}^\top \boldsymbol{U} = \mathbb{I} \implies \boldsymbol{U}\boldsymbol{U}^\top \preceq \mathbb{I} \implies \sum_{t \in [r]} U_{i,t}^2 \leq 1, \forall i \in [p]$. Hence, optimizing for $\mu_{i,t} := U_{i,t}^2$, we get a bound of the form

$$\max_{\boldsymbol{\mu} \geq 0} \quad \sum_{t \in [r]} \sum_{i \in [p]} \mu_{i,t} \left(\sum_{j \in [p]} Z_{i,t} Z_{j,t} |\Sigma_{i,j}| \right) \quad \text{s.t.} \sum_{i \in [p]} \mu_{i,t} = 1, \forall t \in [r]$$
$$\sum_{t \in [r]} \mu_{i,t} \leq 1, \forall i \in [p].$$

Finally, we recognize that some optimal solution $\boldsymbol{\mu}$ to the above problem is an extreme point (i.e., binary) by the linearity of the objective, giving the overall result.

EC.8. Algorithmic Benchmark: Non-Convex QCQP Solvers

The sparse PCA problem with multiple PCs, either in its original formulation (3) or its equivalent reformulation (7), can be seen as a non-convex mixed-integer quadratically constrained problem—for (7), the rank constraints can be encoded as non-convex quadratic constraints $(\boldsymbol{Y}^t)^2 = \boldsymbol{Y}^t$ or $\boldsymbol{Y}^t =$

$U_t U_t^\top$. However, current non-convex MIQCP solvers cannot handle SDP variables and constraints. Accordingly, we now discuss how to solve our sparse PCA problem exactly using commercial global optimization solvers via formulation (3), and evaluate this option numerically.

EC.8.1. Solving (3) With an Off-the-Shelf MIQCQP Solver

Since feasible solutions are extremely challenging for spatial branch-and-bound solvers to recover when quadratic equality constraints are imposed exactly, especially when these solutions are irrational or of exponential size (cf. Ramana 1997, Bienstock et al. 2023), we relax the constraint $U^\top U = \mathbb{I}$ to require that it is satisfied to within an elementwise tolerance of ϵ. This gives:

$$\max_{\substack{Z \in \{0,1\}^{p \times r}: U \in \mathbb{R}^{p \times r} \\ \langle E, Z \rangle \leq k}} \max_{U \in \mathbb{R}^{p \times r}} \langle UU^\top, \Sigma \rangle \tag{EC.11}$$

$$\text{s.t.} \quad \|U^\top U - \mathbb{I}\|_\infty \leq \epsilon,$$

$$U_{i,t} = 0 \text{ if } Z_{i,t} = 0, \qquad \forall i \in [p], \ t \in [r].$$

We set $\epsilon = 10^{-4}/r^2$ so that the total constraint violation does not exceed 10^{-4}. Problem (EC.11) is a non-convex quadratically constrained mixed-integer problem with pr continuous variables, pr binaries, and r^2 quadratic constraints.

In addition, we strengthen Problem (EC.11) with valid inequalities derived from the ℓ_1 relaxation of sparse PCA, as explored by Dey et al. (2022b,a). Indeed, if the sparsity of each PC, k_t, is specified a priori, we have the valid inequalities

$$\|U_t\|_1 \leq \sqrt{k_t}, \ \forall t \in [r]. \tag{EC.12}$$

Moreover, if k is specified but k_t is not, we instead impose the second-order cone inequalities

$$\|U_t\|_1^2 \leq \sum_{i \in [p]} Z_{i,t}, \ \forall t \in [r], \tag{EC.13}$$

which allows us to model $k_t = \sum_{j=1}^p Z_{j,t}$ in a tractable fashion.

In practice, this approach allows MIQCP solvers to solve Problem (7) to optimality for $pr < 100$ (Section 5.2) and obtain high-quality solutions at larger problem sizes. Note that we avoid mixing both sets of inequalities, as we observed in some preliminary numerical experiments that this sometimes induces numerical instability.

To further improve branch-and-bound, we consider two acceleration strategies:

- Using the solution generated by Algorithm 1 as a warm-start;
- Inform branching decisions by the combinatorial upper bound derived in Section 4.

As spatial branch-and-bound technology improves over time, we believe that it should be possible to solve Problem (EC.11) exactly at larger problem sizes. Indeed, recent works, e.g. Gupta et al. (2023), solve some quadratically constrained problems with up to 50 variables to optimality using

custom branch-and-bound solvers, and Gupta et al. (2023) reports that `Gurobi`'s off-the-shelf QCQP solver has achieved a machine-independent speedup factor of 67.5 in less than two years, which suggests that larger instances of (13)–(EC.11) may soon be in reach.

EC.8.2. Numerical Performance on Pitprops

In this section, we numerically evaluate the quality of our approaches on the `pitprops` dataset (Table EC.1) and an approach based on the currently available non-convex MIQCQP technology.

r	k_t	Alg. 1				Alg. 2			Alg. 3		
		UB	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)
2	2	0.295	**0.295**	0	20.64	**0.295**	0	7.54	**0.295**	0	0.02
2	4	0.408	0.378	0	20.80	0.400	0	4.94	**0.404**	0	0.52
2	6	0.477	0.437	0	20.62	0.452	0	6.77	0.443	0	0.62
2	8	0.501	0.375	0	20.93	**0.476**	0	7.83	0.446	0	0.64
2	10	0.507	0.463	0	21.01	**0.500**	0	6.36	0.464	0	0.72
3	2	0.435	**0.435**	0	20.81	0.424	0	9.98	**0.435**	0	0.03
3	4	0.572	0.525	0	21.03	0.551	0	7.24	**0.555**	0	1.01
3	6	0.641	0.463	0	23.03	**0.608**	0	10.21	0.569	0	0.92
3	8	0.652	0.580	0	20.94	**0.638**	0	8.88	0.569	0	1.06
3	10	0.652	0.392	0	20.81	**0.650**	0	11.43	0.569	0	1.39
4	2	0.554	**0.554**	0	21.22	**0.554**	0	11.05	**0.554**	0	0.97
4	4	0.704	0.470	0	21.07	0.657	0.003	13.81	**0.657**	0	3.25
4	6	0.737	0.537	0	22.22	**0.697**	0.002	12.14	0.644	0	2.07
4	8	0.737	0.553	0	22.71	**0.720**	0	11.84	0.644	0	3.48
4	10	0.737	0.508	0	21.03	**0.736**	0	11.22	0.644	0	3.16
5	2	0.657	0.455	0	20.87	0.647	0	12.23	**0.648**	0	1.57
5	4	0.795	0.586	0	21.27	**0.743**	0	15.60	0.709	0	8.12
5	6	0.807	0.538	0	23.47	**0.779**	0.016	13.95	0.713	0	16.54
5	8	0.807	0.563	0	21.02	**0.800**	0.004	17.94	0.713	0	24.78
5	10	0.807	0.525	0	21.97	**0.807**	0.001	15.01	0.713	0	7.39
6	2	0.749	0.581	0	21.45	0.746	0	12.61	**0.749**	0	4.87
6	4	0.866	0.576	0	21.23	0.807	0.035	23.08	**0.780**	0	54.2
6	6	0.870	0.617	0	23.45	**0.839**	0.044	15.63	0.780	0	12.54
6	8	0.870	0.628	0	21.69	**0.849**	0.011	17.20	0.780	0	8.45
6	10	0.870	0.664	0	21.12	**0.866**	0.006	25.91	0.780	0	86.88
Avg		0.668	0.508	0	21.46	**0.649**	0.005	12.42	0.610	0	9.85

Table EC.1 Performance of Algorithms 1–3 on the pitprops dataset ($p = 13$) using the experimental setup laid out in Section 5.2. We denote the best-performing solution (in terms of the proportion of variance explained minus the total orthogonality constraint violation) in bold. We report the semidefinite upper bound obtained from solving Problem (12) as part of our analysis of Algorithm 1 since the semidefinite upper bound is the tightest bound proposed in the paper, but do not report the upper bound from the other methods to avoid redundancy. Note that k_t denotes the sparsity of each individual component, meaning a set of r PCs have a collective sparsity budget of $k_t r$, and that all objective values are reported in terms of the proportion of variance explained by dividing by p, the number of features. Note that the relative optimality gap from Gurobi was less than 10^{-4} at termination for all results in this table.

Tables EC.2-EC.3 compare commercial spatial branch-and-bound with/without the combinatorial upper bound (19), and with/without a warmstart from Algorithm 1.

In comparison with the performance of Algorithms 2–3, we observe that (i) Gurobi's upper bounds are uniformly worse than the SDP relaxation for $r \geq 4$; (ii) on instances where $\sum_t k_t \leq p$ it explains a comparable amount of variance to Algorithms 2–3, although it explains significantly less variance than Algorithms 2 on instances where $\sum_t k_t > p$; (iii) It requires 3–4 orders of magnitude more time than any of our algorithms.

r	k_t	Branch-and-Bound						Branch-and-Bound with (19)					
		UB	Obj.	Viol.	Nodes	Gap (%)	T(s)	UB	Obj.	Viol.	Nodes	Gap (%)	T(s)
2	2	0.295	**0.295**	0	5100	0.00	10.9	0.295	**0.295**	0	3557	0.00	19.42
2	4	0.404	**0.404**	0	99800	0.01	59.49	0.404	**0.404**	0	109485	0.01	238.48
2	6	0.514	**0.456**	0	1405600	12.72	>600	0.521	0.452	0	1038059	15.20	>600
2	8	0.595	0.467	0	1266600	27.28	>600	0.604	0.465	0	1413722	29.70	>600
2	10	0.633	0.486	0	640900	30.13	>600	0.635	0.488	0	848164	30.03	>600
3	2	0.435	**0.435**	0	22400	0.01	17.92	0.435	**0.435**	0	17924	0.00	176.79
3	4	0.717	0.530	0	542700	35.43	>600	0.753	0.524	0	222894	43.68	>600
3	6	0.846	0.551	0	518100	53.62	>600	0.879	0.560	0	387403	56.96	>600
3	8	0.933	0.585	0	564200	59.45	>600	0.935	0.570	0	612382	64.06	>600
3	10	0.962	0.627	0	383100	53.51	>600	0.963	0.595	0	455231	61.95	>600
4	2	0.566	**0.554**	0	783600	2.02	>600	0.774	**0.554**	0	36000	39.64	>600
4	4	1.089	0.636	0	269900	71.24	>600	1.114	0.610	0	203231	82.70	>600
4	6	1.213	0.642	0	268300	88.92	>600	1.226	0.617	0	201271	98.73	>600
4	8	1.267	0.648	0	251800	95.45	>600	1.266	0.699	0	308274	81.18	>600
4	10	1.289	0.713	0	163200	80.67	>600	1.289	0.714	0	266105	80.62	>600
5	2	0.946	0.641	0	702700	47.60	>600	1.073	0.603	0	44795	77.98	>600
5	4	1.430	0.697	0	215200	105.20	>600	1.468	0.652	0	113951	125.31	>600
5	6	1.555	0.692	0	199400	124.65	>600	1.555	0.686	0	152744	126.63	>600
5	8	1.592	0.761	0	191100	109.14	>600	1.610	0.714	0	156767	125.47	>600
5	10	1.616	0.801	0	147500	101.68	>600	1.620	0.804	0	108311	101.51	>600
6	2	1.323	0.702	0	323700	88.34	>600	1.430	0.698	0	43880	104.84	>600
6	4	1.822	0.761	0	139900	139.43	>600	1.807	0.711	0	93572	154.35	>600
6	6	1.903	0.771	0	114000	146.82	>600	1.907	0.776	0	43754	145.87	>600
6	8	1.932	**0.846**	0	141700	128.42	>600	1.946	0.833	0	62657	133.63	>600
6	10	1.947	**0.868**	0	31200	124.29	>600	1.947	0.864	0	68429	125.36	>600
Avg		1.113	0.623	0	375700	69.04	533.28	1.138	0.613	0	280500	76.22	547.62

Table EC.2 Performance of branch-and-bound without warmstart on the pitprops dataset ($p = 13$) using the experimental setup laid out in Section 5.2, except we use a time limit of 600s for branch-and-bound. We report the performance of branch-and-bound with and without the upper bound developed in Section 5.1 separately. The column "UB" reports the upper bound obtained by the branch-and-bound scheme at the time limit. We use > 600 to denote an instance where branch-and-bound terminates at the 600s time limit. The column gap denotes the relative optimality gap reported by Gurobi at termination (in %). We denote the best-performing solution (in terms of the proportion of variance explained minus the total orthogonality constraint violation) in bold (cont.).

r	k_t	Branch-and-Bound (warm-start)						Branch-and-Bound with (19) (warm-start)					
		UB	Obj.	Viol.	Nodes	Gap (%)	T(s)	UB	Obj.	Viol.	Nodes	Gap (%)	T(s)
2	2	0.295	**0.295**	0	6400	0.00	10.90	0.295	**0.295**	0	143	0.00	19.42
	4	0.404	**0.404**	0	75200	0.01	59.49	0.404	**0.404**	0	88735	0.01	238.5
	6	0.521	0.453	0	1051700	15.10	>600	0.524	0.445	0	744000	17.89	>600
	8	0.598	0.463	0	826700	29.31	>600	0.604	0.462	0	1447055	30.73	>600
	10	0.633	0.489	0	686100	29.55	>600	0.636	0.487	0	687944	30.63	>600
3	2	0.435	**0.435**	0	48600	0.00	17.92	0.435	**0.435**	0	2490	0.00	176.8
	4	0.716	0.536	0	491800	33.59	>600	0.752	0.524	0	240207	43.42	>600
	6	0.863	0.560	0	470900	54.16	>600	0.855	0.567	0	598767	50.86	>600
	8	0.935	0.578	0	383000	61.89	>600	0.937	0.566	0	239381	65.68	>600
	10	0.963	0.603	0	297600	59.74	>600	0.963	0.595	0	355350	61.91	>600
4	2	0.554	**0.554**	0	604900	0.01	>600	0.554	**0.544**	0	29467	38.09	>600
	4	1.107	0.627	0	264300	76.52	>600	1.126	0.646	0	136767	74.26	>600
	6	1.208	0.633	0	250300	90.89	>600	1.220	0.625	0	247504	95.30	>600
	8	1.273	0.676	0	225600	88.26	>600	1.271	0.651	0	216736	95.07	>600
	10	1.289	0.702	0	210000	83.73	>600	1.291	0.712	0	165362	81.32	>600
5	2	0.907	**0.656**	0	279800	38.27	>600	1.132	0.616	0	28514	83.71	>600
	4	1.419	0.718	0	251000	97.50	>600	1.475	0.680	0	141932	116.99	>600
	6	1.549	0.699	0	225900	121.73	>600	1.555	0.677	0	229664	129.75	>600
	8	1.609	0.743	0	223700	116.69	>600	1.603	0.743	0	200306	115.90	>600
	10	1.619	0.800	0	126300	102.49	>600	1.621	0.794	0	135142	103.99	>600
6	2	1.285	**0.749**	0	253200	71.53	>600	1.647	**0.749**	0	11818	119.79	>600
	4	1.821	0.778	0	89900	133.89	>600	1.859	0.737	0	141476	152.33	>600
	6	1.891	0.780	0	185600	142.52	>600	1.906	0.768	0	126387	148.36	>600
	8	1.927	0.833	0	124300	131.24	>600	1.942	0.856	0	91093	126.86	>600
	10	1.948	0.867	0	43200	124.74	>600	1.948	0.866	0	82759	124.83	>600
Avg		1.111	0.625	0	307800	68.13	>600	1.150	0.618	0	255600	76.30	>600

Table EC.3   Performance of branch-and-bound with warmstart on the pitprops dataset ($p = 13$) using the experimental setup laid out in Section 5.2, except we use a time limit of 600s for branch-and-bound. We report the performance of branch-and-bound with and without the upper bound developed in Section 5.1 separately. We use >600 to denote an instance where branch-and-bound terminates at the 600s time limit. The column gap denotes the relative optimality gap reported by Gurobi at termination (in %). We denote the best-performing solution (in terms of the proportion of variance explained minus the orthogonality violation) in bold (cont.).

We observe that including the combinatorial upper bound developed in Section 4 within the branch-and-bound scheme does more harm than good, and that using Algorithm 1 as a warmstart marginally improves the performance.

Furthermore, we observe that the upper bound returned by branch-and-bound outperforms the semidefinite upper bound from Problem (12) for the smallest combinations of r and k, but rapidly becomes worse as k and r increases, to the extent that it is unable to provide an upper bound better than the trivial bound of 1 for the largest combinations of r and k. This suggests that the upper bound from branch-and-bound is not practically useful for larger problem instances.

EC.9. Supplementary Numerical Results on UCI Datasets

This section provides supplementary results supporting the numerical experiments performed in Section 5 on UCI datasets.

EC.9.1. Description of the Experimental Setup

All experiments were performed on MIT's supercloud cluster (Reuther et al. 2018), which hosts Intel Xeon Platinum 8260 processors and Intel Xeon Gold 6248 processors. For experiments where $p < 100$, we use Platinum processors with 32 GB RAM, for experiments where $p \in [100, 250]$, Platinum processors with 100 GB RAM; and for $p > 250$, Gold processors with 370 GB RAM.

We also implement some existing algorithmic strategies from the literature, to provide a baseline for the performance of our methods. To abide by software licensing restrictions, all existing strategies from the literature were benchmarked using a MacBook Pro laptop with a 2.9GHz 6-Core Intel i9 CPU, using 16 GB DDR4 RAM. Therefore, runtimes are not directly comparable across strategies.

EC.9.2. Description of the Data Sources

We perform experiments on eleven datasets from the frequently used UCI database in Sections 5.1-5.2 and 5.4. Of the eleven datasets, six datasets are overdetermined (meaning $n > p$), while five datasets are underdetermined (meaning $p > n$). Moreover, many existing works on sparse PCA report results on similar datasets. For instance, the pitprops dataset was also considered by Jolliffe et al. (2003), Zou et al. (2006), Journée et al. (2010) among others, and three of the datasets studied by Berk and Bertsimas (2019) are included within our suite of datasets. Thus, our experimental setup is broadly representative of both the underdetermined and the overdetermined regimes, as well as of the literature. For completeness, we summarize the datasets we benchmark on and their dimensionality in Table EC.4.

Dataset	p	n
Pitprops	13	180
Wine	13	178
Ionosphere	34	351
Lung (Lung cancer)	54	32
Geographical (Geographical Origin of Music)	68	1059
Communities (Communities and Crime)	101	1994
Arrhythmia	274	452
Voice (LSVT Voice Rehabilitation)	310	126
Gait (Gait Classification)	320	48
Gastro (Gastrointestinal Lesions in Regular Colonoscopy)	466	152
Micromass	1300	931

Table EC.4 Summary of the 11 datasets in our library, where n denotes the number of observations and p the number of features. For conciseness, the names of certain datasets are abbreviated throughout. For these datasets, we first state the abbreviation used, followed by their full names in brackets. Further, we report the dimensionality of each dataset after preprocessing, removing all features with missing values. All datasets can be found in the UCI database, except the pitprops dataset, which is due to Jeffers (1967) and distributed via the R package ElasticNet.

EC.9.3. Preliminary Experiments With Pitprops Dataset

We now provide instance-wise results for different variants of our methods on the `pitprops` dataset. In particular, we consider invoking the valid inequalities (21) derived in Section 4 to improve branch-and-bound further. When we do so, we also invoke a branching callback each time we expand a node to determine whether the subtree rooted at this node can improve upon the incumbent solution. This is justified by the fact that at each node, some variables $Z_{i,t}$ are fixed to 0, some to 1, and some are not fixed. Accordingly, we can compute an upper bound on any solution with the same fixed variables by relaxing the orthogonality constraint and applying the Gershgorin circle theorem to each component separately; see Bertsimas et al. (2022b, Section 2.4) for a discussion of this callback in the rank-one case. In particular, if the Gershgorin bound for a given subtree is weaker than an incumbent solution, then this subtree does not contain any optimal solutions, and we can prune it from our search tree.

EC.9.4. Instance-Wise Results on Larger UCI Datasets

Next, we provide an instance-by-instance account of the results summarized in Table 4–Table 5, in Tables EC.5-EC.10.

Dataset	p	r	k_t	Alg. 1				Alg. 2				Branch-and-bound		
				UB	Obj.	Viol.	T(s)	UB	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)
Pitprops	13	2	5	0.449	0.429	0	0.99	0.524	0.433	0	5.01	**0.439**	0	> 7200
		2	10	0.507	0.380	0	0.34	0.642	**0.500**	0	4.10	0.498	0	> 7200
		3	5	0.616	0.541	0	0.44	0.786	**0.582**	0	6.48	0.555	0	> 7200
		3	10	0.652	0.511	0	0.46	0.963	**0.650**	0	6.41	0.618	0	> 7200
Wine	13	2	5	0.458	0.401	0	0.35	0.529	0.446	0	2.19	**0.448**	0	1073
		2	10	0.554	0.508	0	0.31	0.707	**0.544**	0	4.03	0.537	0	> 7200
		3	5	0.632	0.446	0	0.48	0.794	**0.613**	0	3.56	0.576	0	> 7200
		3	10	0.665	0.528	0	0.46	1.060	**0.660**	0	6.47	0.640	0	> 7200
Ionosphere	34	2	5	0.209	0.203	0	5.97	0.221	**0.204**	0	3.18	0.202	0	> 7200
		2	10	0.305	0.265	0	8.72	0.361	**0.285**	0	3.95	**0.285**	0	> 7200
		2	20	0.378	0.286	0	23.31	0.500	**0.360**	0	7.64	0.344	0	> 7200
		3	5	0.297	0.287	0	31.04	0.331	0.279	0	6.72	**0.289**	0	> 7200
		3	10	0.411	0.305	0	39.00	0.542	0.397	0	11.89	0.375	0	> 7200
		3	20	0.464	0.390	0	10.26	0.749	**0.458**	0	12.17	0.383	0	> 7200
Lung	54	2	5	0.119	**0.119**	0	17.84	0.124	0.110	0	3.81	0.113	0	> 7200
		2	10	0.176	**0.175**	0	30.81	0.178	0.171	0	2.06	0.168	0	> 7200
		2	20	0.234	0.170	0	30.82	0.262	0.217	0	4.12	0.185	0	> 7200
		3	5	0.173	0.165	0	40.83	0.185	0.160	0	4.45	0.169	0	> 7200
		3	10	0.249	0.194	0	90.20	0.267	0.240	0	4.31	0.188	0	> 7200
		3	20	0.324	0.308	0	34.50	0.393	0.303	0	9.51	0.213	0	> 7200
Geography	68	2	5	0.147	0.145	0	99.45	0.147	**0.147**	0	2.99	0.145	0	> 7200
		2	10	0.294	0.290	0	107.27	0.294	**0.294**	0	1.81	0.292	0	> 7200
		2	20	0.433	0.393	0	1213.3	0.564	0.376	0	6.32	0.327	0	> 7200
		3	5	0.221	0.213	0	119.3	0.221	**0.221**	0	2.58	0.215	0	> 7200
		3	10	0.410	0.342	0	1453.19	0.441	0.348	0	5.39	0.355	0	> 7200
		3	20	0.529	0.457	0	1571.86	0.846	0.345	0	11.51	0.352	0	> 7200
Communities	101	2	5	0.095	**0.095**	0	484.0	0.096	0.078	0	3.73	**0.095**	0	> 7200
		2	10	0.169	**0.169**	0	1327	0.175	0.159	0	4.81	0.160	0	> 7200
		2	20	0.268	0.219	0	2438	0.284	0.244	0	6.26	0.198	0	> 7200
		3	5	0.141	**0.141**	0	979.7	0.144	0.119	0	7.63	**0.141**	0	> 7200
		3	10	0.246	0.242	0	3553	0.262	0.243	0	8.92	0.205	0	> 7200
		3	20	0.385	0.267	0	3231	0.425	**0.370**	0	8.28	0.300	0	> 7200
Arrhythmia	274	2	5	0.031	0.021	0	583.7	0.031	0.027	0	16.57	0.027	0	> 7200
		2	10	0.055	0.035	0	555.4	0.055	0.047	0	24.72	0.044	0	> 7200
		2	20	0.086	0.067	0	622.8	0.084	0.071	0.002	49.41	0.059	0	> 7200
		3	5	0.047	0.031	0	1423.0	0.046	0.039	0	27.99	0.044	0	> 7200
		3	10	0.083	0.044	0	1085.8	0.083	0.067	0	38.16	0.065	0	> 7200
		3	20	0.129	0.083	0	1059.7	0.126	0.105	0	28.96	0.068	0	> 7200

Table EC.5   Performance of Algorithms 1–2 and branch-and-bound on UCI datasets. k_t denotes the sparsity of each individual component, meaning a set of r PCs have a collective sparsity budget of $k_t r$. Note that all objective values are reported in terms of the proportion of correlation explained by dividing by p, the number of features.

Dataset	p	r	k_t	Alg. 1				Alg. 2				Branch-and-bound		
				UB	Obj.	Viol.	T(s)	UB	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)
Voice	310	2	5	0.032	0.024	0	375.0	0.032	**0.032**	0	21.14	**0.032**	0	> 7200
		2	10	0.064	**0.064**	0	741.2	0.064	0.063	0	21.99	**0.064**	0	> 7200
		2	20	0.127	**0.127**	0	630.6	0.127	0.124	0	20.97	0.109	0	> 7200
		3	5	0.048	**0.048**	0	758.3	0.048	0.047	0	29.94	**0.048**	0	> 7200
		3	10	0.096	0.079	0	797.1	0.096	0.093	0	34.52	**0.096**	0	> 7200
		3	20	0.191	0.125	0	721.5	0.191	0.183	0	31.06	0.155	0	> 7200
Gait	320	2	5	0.031	0.018	0	399.2	0.031	0.028	0	19.40	0.028	0	> 7200
		2	10	0.057	0.036	0	450.7	0.057	0.050	0	24.64	0.047	0	> 7200
		2	20	0.103	0.062	0	392.3	0.103	0.081	0	24.51	0.067	0	> 7200
		3	5	0.046	0.036	0	933.7	0.046	0.041	0	27.80	**0.045**	0	> 7200
		3	10	0.085	0.049	0	829.3	0.085	0.077	0	34.99	0.060	0	> 7200
		3	20	0.154	0.060	0	792.1	0.154	0.121	0	34.11	0.111	0	> 7200
Gastro	466	2	5	0.021	**0.021**	0	1633	0.021	**0.021**	0	452.9	**0.021**	0	> 7200
		2	10	0.043	**0.043**	0	1753	0.043	**0.043**	0	38.95	**0.043**	0	> 7200
		2	20	0.086	0.085	0	2166	0.086	0.085	0	535.4	**0.086**	0	> 7200
		3	5	0.032	**0.032**	0	2682	0.032	**0.032**	0	59.97	**0.032**	0	> 7200
		3	10	0.064	**0.064**	0	4307	0.064	**0.064**	0	73.39	**0.064**	0	> 7200
		3	20	0.129	0.126	0	5544	0.128	0.121	0	529.8	**0.128**	0	> 7200
Micromass	1300	2	5	0.008	0.005	0	1089	0.008	0.006	0	170.5	0.004	0	> 7200
		2	10	0.015	0.008	0	13620	0.014	0.011	0	158.6	0.011	0	6100
		2	20	0.027	0.018	0	9213	0.023	0.019	0	440.3	0.018	0	6826
		3	5	0.012	0.008	0	7953	0.011	0.010	0	238.0	0.006	0	> 7200
		3	10	0.023	0.009	0	19640	0.021	0.018	0	208.5	0.009	0	> 7200
		3	20	0.043	0.029	0	18630	0.034	0.030	0	205.9	0.010	0	> 7200
Avg				0.213	0.176	0.000	1899	0.257	0.199	0.000	61.38	0.187	0	> 7200

Table EC.6 Performance of Algorithms 1–2 and branch-and-bound on UCI datasets (cont). k_t denotes the sparsity of each individual component, meaning a set of r PCs has a collective sparsity budget of $k_t r$. Note that all objective values are reported in terms of the proportion of correlation explained by dividing by p, the number of features.

Dataset	p	r	k_t	Berk and Bertsimas (2019)			Hein and Bühler (2010)			Zou et al. (2006)		
				Obj.	Viol.	T(s)	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)
Pitprops	13	2	5	0.421	0.168	1.67	0.418	0	0.11	0.177	1.341	0.12
		2	10	0.502	0.008	0.14	0.502	0.008	0.01	0.139	1.827	0.22
		3	5	0.592	0.675	0.08	0.575	0.166	0.02	0.169	3.462	0.04
		3	10	0.648	0.073	0.07	0.647	0.084	0	0.181	3.771	0.36
Wine	13	2	5	**0.448**	0	0.04	0.422	0.004	0.01	0.127	0.315	0.04
		2	10	0.545	0.020	0.04	0.545	0.02	0	0.068	0.731	0.06
		3	5	0.610	0.019	0.06	0.559	0.092	0	0.225	2.830	0.05
		3	10	0.654	0.059	0.06	0.655	0.093	0	0.232	2.771	0.32
Ionosphere	34	2	5	**0.205**	0	0.08	0.153	0	0.08	0.078	0	0.02
		2	10	0.289	0	0.30	0.288	0	0.01	0.106	0	0.04
		2	20	0.369	0.058	4.45	**0.370**	0.010	0.17	0.147	0.305	0.12
		3	5	0.291	0	0.14	0.227	0	0.02	0.097	1.666	0.07
		3	10	0.392	0.109	0.38	0.365	0.255	0.01	0.100	1.909	0.12
		3	20	0.449	0.183	0.27	0.451	0.037	0.03	0.111	2.111	4.51
Lung	54	2	5	**0.119**	0	0.43	0.107	0	0.34	0.040	0.587	0.04
		2	10	**0.176**	0	0.10	0.170	0	0.03	0.044	0.639	0.12
		2	20	0.220	0.008	0.44	0.184	0	0.05	0.044	0.908	0.63
		3	5	**0.172**	0	0.10	0.149	0	0.03	0.061	2.755	0.15
		3	10	**0.243**	0	0.11	0.234	0.113	0.05	0.054	1.593	0.20
		3	20	0.300	0.219	0.16	0.261	0.081	0.04	0.044	1.703	1.20
Geography	68	2	5	**0.147**	0	0.09	0.097	0	0.01	0.034	1.793	0.41
		2	10	**0.294**	0	0.08	0.164	0	0	0.068	1.939	0.66
		2	20	0.395	0	5.95	0.316	0.135	0.04	0.062	1.754	0.53
		3	5	**0.221**	0	0.13	0.122	0	0.01	0.061	2.720	0.57
		3	10	**0.389**	0	0.18	0.192	0	0.01	0.054	4.021	0.90
		3	20	0.484	0.273	23.66	0.387	0.261	0.06	0.090	5.009	1.40
Communities	101	2	5	**0.095**	0	0.73	0.093	0	0	0.032	0.576	0.05
		2	10	**0.169**	0	1.68	0.154	0	0	0.029	0.605	0.18
		2	20	**0.258**	0	120	**0.258**	0	0.07	0.027	0.090	1.49
		3	5	**0.141**	0	1.18	0.129	0	0.01	0.050	1.854	0.29
		3	10	**0.245**	0	2.85	0.181	0	0.02	0.044	1.504	1.76
		3	20	0.361	0.058	180.1	0.350	0.064	0.02	0.043	1.869	5.27
Arrhythmia	274	2	5	**0.031**	0	2.81	0.012	0	0.02	0.007	1.799	0.71
		2	10	**0.052**	0	61.25	0.011	0	0.03	0.007	1.143	1.08
		2	20	**0.077**	0	120.0	0.043	0.005	0.06	0.006	1.140	4.62
		3	5	**0.046**	0	5.35	0.016	0	0.02	0.012	1.076	0.53
		3	10	**0.074**	0	121.6	0.018	0	0.05	0.012	0.876	3.82
		3	20	**0.109**	0	180.0	0.074	0.005	0.07	0.012	0.694	10.65

Table EC.7   Performance of the methods of Berk and Bertsimas (2019), Hein and Bühler (2010), and Zou et al. (2006) on UCI datasets.

Dataset	p	r	k_t	Berk and Bertsimas (2019)			Hein and Bühler (2010)			Zou et al. (2006)		
				Obj.	Viol.	T(s)	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)
Voice	310	2	5	**0.032**	0	1.04	**0.032**	0	0.05	0.006	0.874	0.71
		2	10	**0.064**	0	1.09	**0.064**	0	0.04	0.006	0.907	2.68
		2	20	**0.127**	0	0.66	**0.127**	0	0.02	0.006	1.017	16.02
		3	5	**0.048**	0	1.72	0.039	0	0.16	0.009	1.834	0.94
		3	10	**0.096**	0	1.56	0.069	0	0.08	0.012	0.242	12.8
		3	20	**0.190**	0	1.22	0.187	0	0.12	0.021	2.121	26.18
Gait	320	2	5	**0.030**	0	0.93	0.027	0	0.02	0.006	1.071	1.75
		2	10	**0.055**	0	0.61	0.051	0	0.05	0.004	1.054	1.27
		2	20	0.094	0	1.36	0.080	0	0.080	0.005	0.852	3.88
		3	5	**0.045**	0	1.02	0.041	0	0.06	0.01	1.16	4.81
		3	10	**0.082**	0	1.64	0.070	0	0.06	0.009	1.821	5
		3	20	**0.135**	0	1.94	0.095	0	0.15	0.008	1.477	9.49
Gastro	466	2	5	**0.021**	0	2.71	0.020	0	0.05	0.007	1.154	1.14
		2	10	**0.043**	0	1.42	0.039	0	0.08	0.007	0.178	1.62
		2	20	**0.086**	0	1.95	0.076	0	0.07	0.005	0.456	3.61
		3	5	**0.032**	0	2.79	0.029	0	0.05	0.006	2.303	1.59
		3	10	**0.064**	0	3.51	0.053	0	0.12	0.007	1.826	3.33
		3	20	**0.128**	0	2.81	0.085	0	0.24	0.008	1.139	46.05
Micromass	1300	2	5	**0.008**	0	45.4	0.004	0	0.77	0.002	0.014	18.05
		2	10	**0.014**	0	120.2	0.007	0	1.09	0.002	0.323	41.97
		2	20	**0.023**	0	120.2	0.012	0	1.34	0.002	0.361	2.64
		3	5	**0.011**	0	71.97	0.005	0	1.54	0.002	3.004	24.30
		3	10	**0.020**	0	180.3	0.008	0	1.80	0.002	2.301	31.89
		3	20	**0.034**	0	180.3	0.013	0	2.64	0.002	1.252	2.02
Avg				0.205	0.031	25.57	0.180	0.023	0.19	0.049	1.458	4.95

Table EC.8 Performance of the methods of Berk and Bertsimas (2019), Hein and Bühler (2010), and Zou et al. (2006) on UCI datasets (cont.).

Dataset	p	r	k_t	Deshpande and Montanari (2014b)			Algorithm 3			
				Obj.	Viol.	T(s)	UB	Obj.	Viol.	T(s)
Pitprops	13	2	5	0.422	0.226	1.00	0.559	0.422	0	0.53
		2	10	0.501	0.104	0.00	0.803	0.456	0	0.05
		3	5	0.592	0.661	0.00	0.827	0.568	0	0.13
		3	10	0.644	0.214	0.00	1.198	0.569	0	0.13
Wine	13	2	5	0.434	0.211	0.02	0.579	0.447	0	0.02
		2	10	0.545	0.021	0	0.876	0.508	0	0.06
		3	5	0.546	0.510	0	0.853	0.577	0	0.21
		3	10	0.656	0.228	0	1.296	0.580	0	0.23
Ionosphere	34	2	5	0.203	0	0.12	0.228	0.202	0	0.08
		2	10	**0.287**	0	0.02	0.401	**0.290**	0	0.14
		2	20	0.368	0.011	0.02	0.618	0.357	0	0.2
		3	5	0.276	0	0.02	0.340	**0.292**	0	0.28
		3	10	0.357	0.143	0.02	0.597	**0.398**	0	1.51
		3	20	0.447	0.166	0.02	0.920	0.408	0	2.32
Lung	54	2	5	0.117	0	0.45	0.139	0.118	0	0.71
		2	10	0.154	0.291	0.04	0.218	0.171	0	0.1
		2	20	0.217	0.495	0.03	0.345	0.216	0	0.12
		3	5	0.164	0	0.03	0.204	0.169	0	0.23
		3	10	0.200	0.558	0.03	0.326	0.225	0	0.56
		3	20	0.290	0.591	0.03	0.514	0.271	0	3.54
Geography	68	2	5	0.130	0	0.09	0.147	**0.147**	0	0.64
		2	10	0.294	0	0.07	0.294	**0.294**	0	0.81
		2	20	0.395	0	0.08	0.568	**0.396**	0	0.61
		3	5	0.184	0	0.08	0.221	**0.221**	0	2.79
		3	10	0.378	0	0.07	0.441	**0.389**	0	2.04
		3	20	0.462	0.073	0.08	0.852	0.479	0	2.56
Communities	101	2	5	0.089	0	0.19	0.097	**0.095**	0	2.3
		2	10	0.158	0	0.23	0.180	0.167	0	1.37
		2	20	0.249	0.145	0.19	0.320	**0.259**	0	1.94
		3	5	0.140	0	0.19	0.146	**0.141**	0	1.14
		3	10	0.208	0.138	0.20	0.270	0.242	0	1.61
		3	20	0.317	0.669	0.19	0.476	0.369	0	3.62
Arrhythmia	274	2	5	0.030	0	1.79	0.032	**0.031**	0	13.71
		2	10	0.051	0.113	1.84	0.060	0.051	0	11.26
		2	20	0.074	0.019	1.88	0.105	0.075	0	8.04
		3	5	0.039	0	1.97	0.049	**0.046**	0	699.5
		3	10	0.072	0	2.35	0.089	0.072	0	223.1
		3	20	0.103	0.243	2.23	0.155	0.107	0	105.5

Table EC.9   Performance of the method of Deshpande and Montanari (2014b) and Algorithm 3 on UCI datasets.

Dataset	p	r	k_t	Deshpande and Montanari (2014b)			Algorithm 3			
				Obj.	Viol.	T(s)	UB	Obj.	Viol.	T(s)
Voice	310	2	5	**0.032**	0	60.46	0.032	**0.032**	0	16.14
		2	10	0.063	0	54.59	0.064	**0.064**	0	14.22
		2	20	0.124	0	39.19	0.128	**0.127**	0	28.01
		3	5	0.047	0	1.42	0.048	**0.048**	0	64.25
		3	10	0.093	0	1.75	0.097	**0.096**	0	66.68
		3	20	0.184	0	1.42	0.192	**0.190**	0	93.89
Gait	320	2	5	**0.030**	0	1.45	0.031	**0.030**	0	12.51
		2	10	0.054	0	1.45	0.058	**0.055**	0	10.94
		2	20	0.089	0	1.42	0.110	**0.096**	0	12.27
		3	5	0.042	0	1.85	0.046	**0.045**	0	47.73
		3	10	0.078	0	1.65	0.087	**0.082**	0	29.65
		3	20	0.128	0	3.04	0.164	**0.135**	0	46.5
Gastro	466	2	5	**0.021**	0	6.59	0.021	**0.021**	0	35.41
		2	10	0.042	0	4.33	0.043	**0.043**	0	30.51
		2	20	0.083	0	3.67	0.086	**0.086**	0	54.39
		3	5	0.031	0	3.26	0.032	**0.032**	0	97.05
		3	10	0.061	0	3.12	0.064	**0.064**	0	115.2
		3	20	0.117	0	3.11	0.129	**0.128**	0	153.1
Micromass	1300	2	5	0.007	0	75.38	0.008	**0.008**	0	982.8
		2	10	0.012	0	147.9	0.014	0.013	0	615.4
		2	20	0.022	0	98.34	0.025	**0.022**	0	481.8
		3	5	0.010	0	71.05	0.012	0.011	0	1023.8
		3	10	0.018	0	71.88	0.021	0.020	0	1015.1
		3	20	0.033	0	73.00	0.038	**0.033**	0	593.0
Avg				0.197	0.094	12.04	0.289	0.199	0	108.6

Table EC.10 Performance of the method of Deshpande and Montanari (2014b) and Algorithm 3 on UCI datasets (cont.)

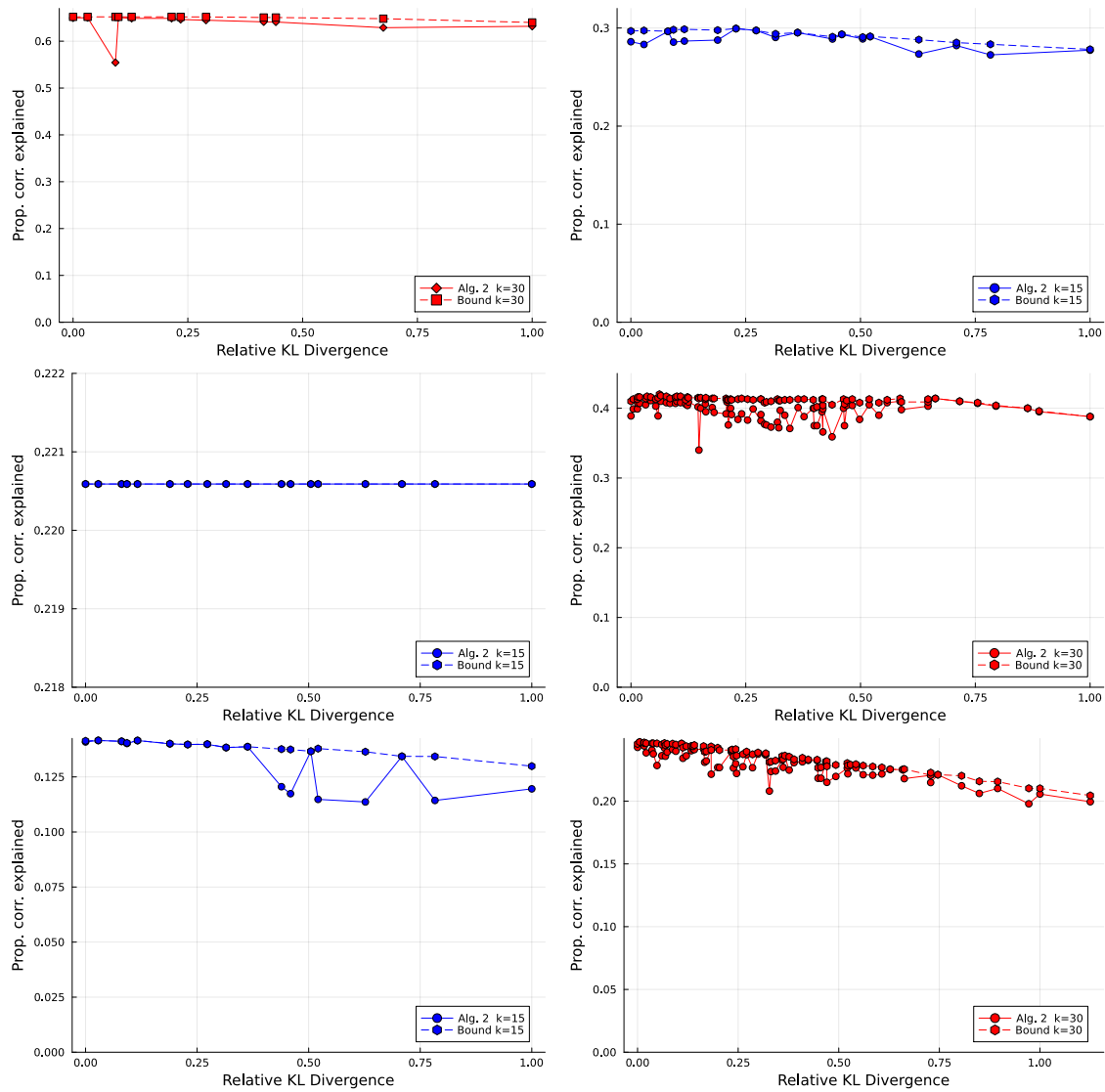## EC.9.5. Instance-Wise Plots of Symmetry vs. Proportion of Correlation Explained

Figure EC.1   Symmetry of sparsity budget allocation vs. proportion of correlation in the dataset explained for pitprops $k = 30$ (top left), ionosphere $k = 15$ (top right), geographical $k = 15$ (middle left), geographical $k = 30$ (middle right), communities $k = 15$ (bottom left), and communities $k = 30$ (bottom right). Note that we normalize the KL divergence for $k = 15$ and $k = 30$ separately.

EC.10. Non-Redundancy of Rank Constraints in Problem (7)

We claimed in Remark 2 that the rank-one constraints in Problem (7) are not redundant. We now demonstrate this by example, by providing an example where, after constraining the support pattern, Problem (7) attains a different optimal value than the following optimization problem:

$$\max_{\substack{\boldsymbol{Z} \in \{0,1\}^{p \times r} : \boldsymbol{Y} \in \mathcal{S}^p, \boldsymbol{Y}^t \in \mathcal{S}^p_+ \\ \langle \boldsymbol{E}, \boldsymbol{Z} \rangle \leq k}} \langle \boldsymbol{Y}, \boldsymbol{\Sigma} \rangle \text{ s.t. } \boldsymbol{Y} \preceq \mathrm{Diag}\left(\min\left(\boldsymbol{e}, \sum_t \boldsymbol{Z}_t\right)\right), \boldsymbol{Y} = \sum_{t=1}^r \boldsymbol{Y}^t, \qquad \text{(EC.14)}$$

$$\mathrm{tr}(\boldsymbol{Y}^t) = 1, \ \forall t \in [r], \ Y^t_{i,j} = 0 \text{ if } Z_{i,t} = 0, \ \forall t \in [r], i, j \in [p],$$

where we take $r = 3, p = 4$, and fix the support in both problems by setting

$$\boldsymbol{Z} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \ \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 8 & 2 & 1 \\ 0 & 2 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{pmatrix}.$$

Solving Problem (7) by letting $\boldsymbol{Y}_t = \boldsymbol{u}_t \boldsymbol{u}_t^\top$ via Gurobi with `NonConvex` $= 2$ gives an optimal objective value of 12.14006. On the other hand, solving Problem (EC.14) via Mosek gives an optimal objective value of 12.25765. Thus, Problems (7)–(EC.14) cannot be equivalent, as they give a different optimal objective value for a fixed binary support.