D-optimal partitioning:
design of experiments under heterogeneous treatment effects

Jiaqi Wang Weijun Xie Ilya O. Ryzhov

June 15, 2026

Abstract

Modern experimentation in business and public policy often studies targeted interventions whose effects depend on the heterogeneous attributes of individuals. We examine heterogeneous treatment effects through the lens of optimal design of experiments, which allocates treatment decisions to maximize the precision of estimated treatment-covariate interactions. We introduce the D-optimal partitioning problem for balancing the information matrices of the treatment and control groups. This problem generalizes the classical concept of D-optimal design, but is much more difficult: we prove that no polynomial-time additive approximation algorithm for D-optimal partitioning can exist unless $P = NP$. We derive a novel semidefinite programming formulation of this problem and propose a column generation algorithm with promising practical performance on both synthetic and real data.

1 Introduction

Experimentation has become a core technology of modern decision-making, allowing firms and policymakers to evaluate prices (Li et al., 2015), product features and new product ideas (Kohavi et al., 2020), digital services (Retana et al., 2016; Somanchi et al., 2023), educational programs (Konstantopoulos, 2008), and public interventions (Imai & Strauss, 2011; Vivalt, 2015) before deploying them at scale. In these and other contexts, the central question of the experiment is no longer simply, "Does the treatment work?" Increasingly, it is, "For whom does it work?" A product feature may improve engagement for some users while discouraging others; a customer-outreach intervention may help inexperienced users adopt a service while doing little for experienced ones; an instructional program may be most valuable for students whose prior achievement makes standard classroom instruction poorly matched to their needs. In these settings, treatment heterogeneity is the information needed to decide how an intervention should be targeted.

Randomized controlled trials remain the gold standard for experimentation because random assignment provides transparent causal identification (Athey & Imbens, 2017). However, a distinction should be made between identification and efficiency: in finite sample, some assignments provide more precise information than others about the covariates that moderate treatment response. It has been observed (Bertsimas et al., 2015) that random assignment can produce dissimilar treatment and control groups in limited-data regimes, and that more powerful inference can be obtained by taking the covariates into account. Indeed, this principle has long been the foundation of the literature on optimal design of experiments (Pukelsheim, 2006), in which a decision-maker observes the covariates of a pool of individuals and selects a subset of these to optimize a criterion that quantifies statistical precision. In much of this literature, the goal was simply to learn the parameter vector of a regression model, without the presence of a treatment decision. More recently, Bhat et al. (2020) used the D-optimality criterion in an A/B testing problem without heterogeneity; in such a setting, the treatment effect is a scalar and the problem reduces to minimizing the variance of its estimator.

This paper studies optimal design in the presence of heterogeneous treatment effects. We build on D-optimality, a criterion that has long been canonical in the literature because it provides an interpretable measure of joint precision that can be directly connected to statistical inference: in linear models, D-optimality minimizes the volume of a joint confidence region for the parameters of interest (Gilmour & Trinca, 2012). This feature is particularly well-matched to heterogeneous treatment effects, where the experimenter learns a vector of treatment-covariate interactions whose components jointly determine how an intervention should be targeted. Thus, our work is related to Bhat et al. (2020) and other papers using D-optimality for A/B testing, such as Pokhilko et al. (2019), with the key difference that the object of interest in our paper is a vector.

This difference turns out to change the fundamental mathematical structure of the problem. In classical D-optimal design, one maximizes the determinant of the information matrix constructed from the selected data vectors. However, in the context of learning heterogeneous treatment effects, we prove that D-optimality has a representation as a balancing problem between *two* information matrices: one constructed from the treated individuals, and the other from the untreated ones. We refer to this problem as *D-optimal partitioning*, to distinguish it from D-optimal design, because the central issue of the problem is now the precise split between the two groups, not the information value of one group in isolation.

In D-optimal design, a data point that accounts for a larger share of the information content of the dataset is always better. Data points are the easiest to select when they offer the most improvement in the objective value. In D-optimal partitioning, a more informative vector makes the problem harder, not easier, because no matter whether that individual is treated or untreated, one of the two information matrices will always improve at the expense of the other, and the imbalance is difficult to mitigate. This structure invalidates many well-established algorithms for D-optimal design. A fairly mature literature, exemplified by Singh & Xie (2020), has focused on suboptimal algorithms with provable approximation guarantees. Not only do these ideas not work for D-optimal partitioning, but we formally prove the non-existence of *any* polynomial time approximation algorithm for this problem unless $P = NP$. This result alone sets our problem apart from existing work. Structurally, D-optimal partitioning is closer to matrix discrepancy problems, a difficult theoretical subfield of algebra and functional analysis (Marcus et al., 2015).

For these reasons, we focus on exact solution techniques based on column generation (Desrosiers & Lübbecke, 2005). We first derive a novel reformulation of D-optimal partitioning as a discrete semidefinite program, whose continuous relaxations are tighter than those of the more traditional vector-based formulation. We then tackle the tightest possible relaxation (the convex hull of the constraints) using a Frank-Wolfe scheme. The approach alternates between three stages: the first is a gradient step where we optimize a linear approximation of the objective function over a set of possible directions; the second checks whether the best of these directions offers any improvement; and, if so, the third incorporates that direction into a list of vertices of the convex hull. As with all column generation methods for integer programming, the first stage is the most expensive, involving a discrete nonconvex pricing problem. However, as is also common with such procedures (see, e.g., Desaulniers et al., 2019 for examples in transportation, where column generation methods are widely used), improvement can often be obtained by using cheap pricing heuristics, and only the final iteration requires the pricing problem to be solved to optimality.

Our approach can potentially be integrated into exact solution frameworks such as branch-and-price or branch-price-and-cut (Costa et al., 2019), but one appealing aspect of it is that the pricing step provides us with feasible discrete solutions (lower bounds), even if heuristics were used to compute it. The least computationally expensive approach is simply to report the best pricing solution obtained across all iterations of the procedure. We implemented this approach on both synthetic and real data and found that it works exceptionally well in many situations. For example, in one of our instances (which uses real 401(k) pension data), column generation finds a near-optimal solution in 5.54 seconds. A more principled exact row-generation approach that uses column generation internally was unable to improve on that solution in an hour.

This paper makes the following contributions: 1) We are the first to pose the D-optimal partitioning problem and to show that the D-optimality criterion takes this form in the presence of heterogeneous treatment effects. 2) We prove that, unless $P = NP$, D-optimal partitioning does not admit any polynomial-time algorithm with an additive performance guarantee, which strongly sets this problem apart from classical optimal design. 3) We derive a novel semidefinite reformulation of D-optimal partitioning and design a column-generation algorithm that provides both lower and upper bounds on solution quality and can tractably solve moderately-sized problems. 4) We demonstrate the strong empirical performance of this approach, relative to several nontrivial benchmarks, on real datasets that have been used in the past to study heterogeneous treatment effects.

2 Literature review

Heterogeneous treatment effects arise in a wide variety of applications in information systems (Leng & Dimmery, 2024), revenue management (Pauphilet, 2024), labor economics (Smith, 2022), medicine (Rekkas et al., 2020), and education (Bonesrønning et al., 2022). These and many other studies focus mainly on post-experimental estimation and analysis under a given randomized design.

We focus on the design stage, and thus our paper is closer to the literature on A/B testing. Bhat et al. (2020) stands out as a recent example centered around design. In fact, this work also uses the D-optimality criterion, though the problem has a simpler form because only homogeneous (scalar) treatment effects are considered. On the other hand, Bhat et al. (2020) considers online decision-making, where the result of a single treatment decision can be observed before assigning the next individual. Online experimentation is related to contextual bandits (Bastani et al., 2021), which can be practical in some fields, such as e-commerce, where responses can be observed quickly and the cost of a single experiment is relatively low. However, there are many other situations, e.g., rollouts of new app features or public health interventions (Xiong et al., 2024), where a candidate set of individuals is observed before an intervention is deployed. Furthermore, many business outcomes such as bookings, retention, conversion, or lifetime value are observed with delay, making real-time adaptation less useful because informative feedback is unavailable until after many assignments have been made (Deng et al., 2023). For these reasons, we focus on the nonadaptive setting in this paper, treating design as distinct from implementation. There has also been extensive recent research on A/B testing in this setting (Pokhilko et al., 2019; Zhang & Kang, 2022), mostly also with homogeneous treatment effects. A recent paper by Zhang et al. (2022) considered optimal design for heterogeneous effects with a focus on scalar treatment-effect contrasts over a prescribed covariate set, whereas our criterion pertains more to joint inference over the treatment-covariate interaction vector.

The design-focused literature has found that precise estimation of treatment effects requires the covariates to be distributed similarly within the treatment and control groups. The methods proposed to handle this issue include propensity score matching (Li et al., 2018), rerandomization (Morgan & Rubin, 2012), and integer programming (Bertsimas et al., 2015). Because these papers generally work with homogeneous treatment effects, they find it sufficient to balance the marginal distributions of covariates across the two groups, which does not address the multicollinearity between main effects and treatment-effect interactions. On the other hand, the D-optimality criterion is explicitly known to target multicollinearity (O'Brien & Funk, 2003), and the geometry of the covariates both within and between the two groups is central to our formulation.

D-optimal design has been extensively studied in statistics; see the survey by Huan et al. (2024). There is a substantial literature on using mathematical programming models and methods to solve this problem (Vandenberghe et al., 1998; Lu et al., 2018). Work by Sagnol & Harman (2015) and Coey et al. (2023) investigated conic optimization (used as a benchmark in our paper) as a solution framework. There is also a stream of research on suboptimal algorithms with approximation guarantees for classical D-optimal design (Singh & Xie, 2020; Allen-Zhu et al., 2021), as well as extensions with prior information (Li et al., 2024) or knapsack constraints (Wang et al., 2026). None of this work considers heterogeneous treatment effects; indeed, a central result of our paper is that the D-optimal partitioning problem cannot be solved using such techniques.

As mentioned in Section 1, our problem has some structural similarity to matrix-balancing problems such as those studied in Marcus et al. (2015), Dadush et al. (2022), and Bansal et al. (2023). For the most part, this literature does not make an explicit connection to statistics, and the objective in these problems usually depends only on the largest and smallest eigenvalues of the information matrix, whereas our D-optimality criterion balances all eigen-directions. Harshaw et al. (2024) used some ideas from discrepancy theory for the purpose of experimental design, but that work focused on scalar treatment effects only.

3 Model and first formulation

This section derives a formulation of D-optimal partitioning as a discrete concave program. Our solution technique will be based on a different formulation based on semidefinite programming, to be introduced in Section 4. We begin with the formulation presented in this section because it is more intuitive, makes a clean and interpretable contrast with the D-optimal design problem, and is more tractable for complexity analysis.

Section 3.1 begins with a linear regression model with heterogeneous treatment effects and derives a representation for the D-optimality criterion as a partitioning problem. Section 3.2 proves that, unlike classical D-optimal design, this problem does not admit a polynomial-time additive approximation algorithm unless $P = NP$. Section 3.3 discusses the connection between D-optimal partitioning and the matrix discrepancy literature.

3.1 Derivation and formulation

Suppose that there are n individuals. For $i = 1, ..., n$, let $\boldsymbol{x}_i \in \mathbb{R}^d$ represent attributes (covariates) of the ith individual. Let $Z_i \in \{-1, 1\}$ denote the individual's treatment status ($Z_i = 1$ if treated, $Z_i = -1$ otherwise). We formulate the linear regression model

$$y_i = Z_i \cdot \boldsymbol{x}_i^\top \boldsymbol{\theta} + \boldsymbol{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, ..., n, \tag{1}$$

where $\boldsymbol{\beta} \in \mathbb{R}^d$ represents a baseline effect for the covariates and $\boldsymbol{\theta}$ is a modification due to treatment status. The residual errors ε_i are assumed to be independent with zero mean and common variance σ^2. Following Bhat et al. (2020), we do not consider issues of endogeneity (unobserved covariates) in this paper.

The effect of the treatment is heterogeneous, i.e., dependent on the covariates: for the ith individual, the expected response is $\boldsymbol{x}_i^\top (\boldsymbol{\beta} + \boldsymbol{\theta})$ if the individual is treated and $\boldsymbol{x}_i^\top (\boldsymbol{\beta} - \boldsymbol{\theta})$ if not. The difference $2\boldsymbol{x}_i^\top \boldsymbol{\theta}$ is precisely the additional, individual-specific effect of the treatment on the response, also known as the conditional average treatment effect (Athey & Imbens, 2016).

Let $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n]^\top$ denote the design matrix, assumed to have full column rank. We also denote $\boldsymbol{Y} = [y_1, ..., y_n]^\top$, $\boldsymbol{Z} = [Z_1, ..., Z_n]^\top$ and $\boldsymbol{Z}^* = [\text{diag}(\boldsymbol{Z}) \boldsymbol{X}, \boldsymbol{X}]$. The ordinary least squares estimator of $(\boldsymbol{\theta}, \boldsymbol{\beta})$ is given by

$$\left[\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}\right]^\top = \left((\boldsymbol{Z}^*)^\top \boldsymbol{Z}^*\right)^{-1} \left((\boldsymbol{Z}^*)^\top \boldsymbol{Y}\right)$$

At a high level, the optimal design problem treats the covariates $\boldsymbol{x}_i$ as given, and aims to choose Z_i for each individual in a way that optimizes some measure of uncertainty in the estimated treatment effect. When the treatment effect is homogeneous (e.g., in Pokhilko et al., 2019 or Bhat et al., 2020), $\boldsymbol{\theta}$ reduces to a scalar and it is natural to simply minimize the variance of its estimator. In (1), however, $\boldsymbol{\theta}$ is a vector, and the standard approach in optimal design is to optimize some statistical criterion derived from the covariance matrix (or the information matrix) of the estimated regression coefficients. In this paper, we are concerned only with the accuracy of $\hat{\boldsymbol{\theta}}$, and we focus on D-optimal designs, which choose $\boldsymbol{Z}$ to minimize $\det\left(Cov\left(\hat{\boldsymbol{\theta}}\right)\right)$ or some tractable reformulation thereof. Such designs are known to reduce collinearity between the dimensions of the covariate vector and lower worst-case prediction uncertainty (O'Brien & Funk, 2003).

The first step in formulating the D-optimal design criterion for heterogeneous treatment effects is to derive the relevant covariance matrix. By the properties of ordinary least squares regression, we know that

$$Cov\left(\left[\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\beta}}\right]\right) = \sigma^2\left(\left(\boldsymbol{Z}^*\right)^\top \boldsymbol{Z}^*\right)^{-1} = \sigma^2 \begin{pmatrix} \boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})^2 \boldsymbol{X} & \boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\boldsymbol{X} \\ \boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\boldsymbol{X} & \boldsymbol{X}^\top\boldsymbol{X} \end{pmatrix}^{-1}. \tag{2}$$

The covariance matrix of $\boldsymbol{\theta}$ is the submatrix corresponding to the first d indices of (2). Using Schur complements, as well as the fact that $\operatorname{diag}(\boldsymbol{Z})^2 = \boldsymbol{I}$ because $Z_i \in \{-1,1\}$, we may derive

$$\begin{aligned} Cov\left(\hat{\boldsymbol{\theta}}\right) &= \left(\boldsymbol{X}^\top\boldsymbol{X} - \boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\boldsymbol{X}\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\boldsymbol{X}\right)^{-1} \\ &= \left(\boldsymbol{I} - \left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\boldsymbol{X}\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1} \\ &= \left(\boldsymbol{I} - \left(\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\boldsymbol{X}\right)^2\right)^{-1}\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1} \\ &= \left(\boldsymbol{I} - \left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\boldsymbol{X}\right)^{-1}\left(\boldsymbol{I} + \left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1} \end{aligned} \tag{3}$$

Because $\boldsymbol{X}^\top\boldsymbol{X}$ is a constant matrix (we only control the treatment variables Z_i), it can be seen that the covariance depends on $\boldsymbol{Z}$ only through the matrix $\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\boldsymbol{X}$.

In the optimization literature (Lu et al., 2018), one often maximizes the log-determinant of the information matrix rather than minimizing the determinant of the covariance matrix. This equivalent objective is more amenable to mathematical programming techniques. We also follow this approach, and therefore focus on

$$\det\left(\boldsymbol{X}^\top\boldsymbol{X} - \boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\boldsymbol{X}\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\boldsymbol{X}\right)$$

$$= \det\left(\boldsymbol{I} - \left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\,\boldsymbol{X}\right)\det\left(\boldsymbol{I} + \left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\,\boldsymbol{X}\right)\det\left(\boldsymbol{X}^\top\boldsymbol{X}\right)$$
$$= \det\left(\boldsymbol{X}^\top\boldsymbol{X} - \boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\,\boldsymbol{X}\right)\det\left(\boldsymbol{X}^\top\boldsymbol{X} + \boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\,\boldsymbol{X}\right)\det^{-1}\left(\boldsymbol{X}^\top\boldsymbol{X}\right).$$

Taking logarithms and ignoring the constant term, we arrive at

$$\begin{aligned} & \log\det\left(\boldsymbol{X}^\top\boldsymbol{X} - \sum_{i=1}^{n} Z_i \boldsymbol{x}_i\boldsymbol{x}_i^\top\right) + \log\det\left(\boldsymbol{X}^\top\boldsymbol{X} + \sum_{i=1}^{n} Z_i \boldsymbol{x}_i\boldsymbol{x}_i^\top\right) \\ = \; & \log\det\left(\sum_{i=1}^{n}(1 - Z_i)\,\boldsymbol{x}_i\boldsymbol{x}_i^\top\right) + \log\det\left(\sum_{i=1}^{n}(1 + Z_i)\,\boldsymbol{x}_i\boldsymbol{x}_i^\top\right). \end{aligned} \tag{4}$$

To obtain a more standard formulation as a combinatorial optimization problem, we can make a change of variables $z_i = \frac{1+Z_i}{2}$. Then, rewriting (4) and dropping another constant term, we arrive at the *D-optimal partitioning* objective

$$\max_{z_i \in \{0,1\}} \log\det\left(\sum_{i=1}^{n} z_i x_i x_i^\top\right) + \log\det\left(\sum_{i=1}^{n}(1 - z_i)\, x_i x_i^\top\right). \tag{5}$$

In many situations, treatment decisions are subject to an additional budget constraint

$$\sum_{i=1}^{n} z_i \le k, \tag{6}$$

for some pre-specified $k < \lceil \frac{n}{2} \rceil$; when k is above this threshold, the constraint is redundant due to the symmetry of the objective. When (6) is not present (i.e., we solve (5) only), we refer to the problem as "unconstrained." Otherwise, we refer to (5)-(6) together as the D-optimal partitioning problem.

At first glance, Problem (5)-(6) appears to be closely related to the classical D-optimal design problem studied by Singh & Xie (2020) and others. Indeed, the objective of D-optimal design is simply the first term in (5), whose value is affected only by those i with $z_i = 1$. D-optimal partitioning is distinguished from D-optimal design by the presence of the second term, whose value is affected by those i with $z_i = 0$. Because *every* individual impacts exactly one of the two terms, it is more accurate to say that we are optimally partitioning the set of individuals into treated and untreated, rather than optimally selecting the treatment group.

Despite the apparent similarity, D-optimal partitioning turns out to be very different structurally from D-optimal design, and also much more difficult. We will prove a formal notion of difficulty shortly, but the high-level intuition is as follows. The D-optimal objective $\log\det\sum_i z_i \boldsymbol{x}_i\boldsymbol{x}_i^\top$ is monotonic, in the sense that, all else being equal, it is always better to treat more individuals. Because the log-determinant is submodular, myopic methods such as greedy algorithms or local search can work exceptionally well in practice. In D-optimal partitioning, however, there is always a tradeoff between the two log-determinants in (5). We discuss this issue further in Section 3.3, but in brief, an $\boldsymbol{x}_i$ vector that is unambiguously beneficial in classical D-optimal design (because it improves a single log-determinant term by a large amount) can cause problems in D-optimal partitioning (because the improvement in one log-determinant always comes at the expense of the other).

It is worth mentioning the continuous relaxation of (5), given by

$$\max_{z_i \in [0,1]} \log\det\left(\sum_{i=1}^{n} z_i x_i x_i^\top\right) + \log\det\left(\sum_{i=1}^{n} (1-z_i)\, x_i x_i^\top\right). \tag{7}$$

It is straightforward to show that, if (6) is not included, (7) is optimally solved by letting $z_i \equiv \frac{1}{2}$ for all i. This provides some intuition for the appeal of randomized control trials: if it were possible, we would prefer to divide each data vector into two equal parts. However, the fact that the optimal solution to the continuous relaxation has no dependence on the data vectors $\boldsymbol{x}_i$ themselves suggests that the relaxation is likely to be weak.

Indeed, this can be seen with a simple example. Let $n = 2$, $d = 1$, $x_1 = M$ for some $M > 0$, and $x_2 = 1$. Then, there exists only one feasible solution up to symmetry, which is $S = \{1\}$. The optimal value of (5) is easily found to be $2\log M$. However, the optimal value of (7) is $2\log\left(\frac{1+M^2}{2}\right)$. The gap $2\log\left(\frac{1+M^2}{2}\right) - 2\log M \to \infty$ as $M \to \infty$, so the relaxation can be arbitrarily weak. This is another hint that D-optimal partitioning is fundamentally different from D-optimal design, as the continuous relaxation of that problem was used by Singh & Xie (2020) and others to develop efficient algorithms. Here, the same approach does not work, which helps to motivate a different SDP-based formulation, to be introduced in Section 4, whose continuous relaxation is stronger than that of (5).

3.2 Impossibility of polynomial-time approximation scheme

In the context of D-optimal partitioning, an additive approximation is defined as follows. Let f be a set-valued function returning

$$f(S) = \log\det\left(\sum_{i\in S} \boldsymbol{x}_i \boldsymbol{x}_i^\top\right) + \log\det\left(\sum_{i\notin S} \boldsymbol{x}_i \boldsymbol{x}_i^\top\right)$$

for $S \subseteq \{1, ..., n\}$ and possibly $|S| \leq k$ if we wish to include the budget constraint. Then, for $\alpha > 0$, a feasible S_0 provides an additive α-approximation to the optimal solution if

$$f(S_0) \geq \max_{S \in \mathcal{S}} f(S) - \alpha, \tag{8}$$

with $\mathcal{S}$ being the desired feasible solution space.

Previous work on D-optimal design and its variants, including Singh & Xie (2020), Madan et al. (2020), Li et al. (2024) and other papers, focused on the development of polynomial-time approximation schemes, or algorithms that provably yield α-optimal solutions. Our main result in this section is that, unless $P = NP$, such algorithms *cannot* exist even for the unconstrained version of D-optimal partitioning. This is a striking contrast with D-optimal design, and establishes D-optimal partitioning as an entirely distinct problem. The result also helps motivate the need for a different approach, to be developed in Section 4.

Our proof is based on a connection to the MAX-CUT problem, formally defined as follows.

Definition 3.1. *Let $G = (V, E)$ be an undirected graph of degree at most 3. The MAX-CUT problem finds a partition of V into two disjoint sets P_R and P_G such that the cardinality of edges that are cut (that is, edges with one endpoint in P_R and the other in P_G) is maximized. We denote by $c(G)$ the cardinality of the optimal cut.*

Alimonti & Kann (2000) showed that MAX-CUT belongs to a complexity class known as APX-complete (Crescenzi et al., 1999). This means that there exists some $\varepsilon_0 > 0$ such that, unless $P = NP$, there is no polynomial-time algorithm that can be guaranteed to return a cut value of at least $(1 - \varepsilon_0)\, c(G)$.

We first state a lemma connecting this notion of hardness of MAX-CUT to an intermediate problem. Then, we state the main hardness theorem. The proofs of these and subsequent results are deferred to the Appendix due to space considerations.

Lemma 3.1. *There exist constants $0 < \gamma_1 < \gamma_2 < 1$ such that, among all graphs $G = (V, E)$ of degree at most 3 with MAX-CUT value $c(G)$, it is NP-hard to distinguish a graph satisfying $c(G) \geq \gamma_2 |E|$ from one satisfying $c(G) \leq \gamma_1 |E|$.*

Theorem 3.1. *Consider the unconstrained version of the D-optimal partitioning problem. For any $\alpha > 0$, unless $P = NP$, there is no polynomial-time approximation algorithm that can be guaranteed to output S_0 satisfying (8).*

Note that the problem defined in Lemma 3.1 is not the MAX-CUT problem itself, but rather, a different "gap-distinguishing" problem whose formulation is related to the optimal MAX-CUT value of the graphs. The proof of Theorem 3.1 proceeds by contradiction: we construct a special instance of D-optimal partitioning and show that, if there existed a polynomial-time algorithm yielding an α-optimal solution to it, the objective value of that solution could be used to efficiently solve the problem of Lemma 3.1. This, of course, is impossible unless $P = NP$. Consequently, D-optimal partitioning is not only more difficult than D-optimal design, but also more difficult than the entire APX-complete problem class, because the latter still allows constant additive approximations.

3.3 Connection to the matrix discrepancy problem

The difficulty of D-optimal partitioning is closely related to the concept of the *leverage score* of a data vector. For $i = 1, ..., n$, let

$$\boldsymbol{u}_i = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-\frac{1}{2}} \boldsymbol{x}_i. \tag{9}$$

Note that $\sum_{i=1}^n \boldsymbol{u}_i \boldsymbol{u}_i^\top = \boldsymbol{I}$. The ith leverage score is the quantity $0 < \boldsymbol{u}_i^\top \boldsymbol{u}_i < 1$. Roughly speaking, it measures how much of the total information $\boldsymbol{X}^\top \boldsymbol{X}$ is contained in $\boldsymbol{x}_i$. For example, if the ith vector is orthogonal or nearly orthogonal to all others, its leverage score will be close to 1. On the other hand, if the dataset contains many vectors similar to $\boldsymbol{x}_i$, the ith leverage score will be low.

The D-optimal partitioning problem can be reformulated in terms of the transformed data vectors $\boldsymbol{u}_i$. We may rewrite (4) as

$$\begin{aligned} & \log\det\left(\boldsymbol{X}^\top \boldsymbol{X} - \operatorname{diag}(\boldsymbol{Z})\right) + \log\det\left(\boldsymbol{X}^\top \boldsymbol{X} + \operatorname{diag}(\boldsymbol{Z})\right) \\ = & \log\det\left(\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{\frac{1}{2}}\left(\boldsymbol{I} - \sum_{i=1}^n Z_i \boldsymbol{u}_i \boldsymbol{u}_i^\top\right)\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{\frac{1}{2}}\right) \\ + & \quad \log\det\left(\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{\frac{1}{2}}\left(\boldsymbol{I} + \sum_{i=1}^n Z_i \boldsymbol{u}_i \boldsymbol{u}_i^\top\right)\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{\frac{1}{2}}\right) \\ = & 2\log\det\left(\boldsymbol{X}^\top \boldsymbol{X}\right) + \log\det\left(\boldsymbol{I} - \left(\sum_{i=1}^n Z_i \boldsymbol{u}_i \boldsymbol{u}_i^\top\right)^2\right). \end{aligned}$$

The constant term can be dropped, leaving the objective

$$\max_{Z_i \in \{-1,1\}} \log\det\left(\boldsymbol{I} - \left(\sum_{i=1}^n Z_i \boldsymbol{u}_i \boldsymbol{u}_i^\top\right)^2\right). \tag{10}$$

For simplicity during this discussion, suppose that we are not budget-constrained. Problem (10) is an example of a "matrix discrepancy" problem, a class where sign assignments $\boldsymbol{Z} \in \{\pm 1\}^n$ are chosen to balance some function of a sum of rank-1 matrices. By maximizing the log-determinant, we are essentially balancing the sum of the log-eigenvalues of the matrix. A closely related objective in the literature minimizes the difference between the largest and smallest eigenvalues. This problem is notoriously intractable for computation; the state of the art (e.g., the seminal paper by Marcus et al., 2015) studies the *existence* of solutions whose optimality gap can be provably bounded, rather than computing them. Even these existence results require bounds of the form $\boldsymbol{u}_i^\top \boldsymbol{u}_i \leq L < 1$ on the leverage scores, with the bound being better for smaller L.

This last point is the most relevant for the present paper. If there exists one $\boldsymbol{x}_i$ whose leverage score is close to 1, such a vector would be unambiguously good in the classical D-optimal design problem. However, in D-optimal partitioning, such a vector causes pathological issues for analysis. As we develop other formulations and relaxations in Section 4, we will make use of the leverage score concept to describe when the relaxations are tighter.

4 Semidefinite programming formulation

We now present a different formulation of D-optimal partitioning, based on semidefinite programming. This problem, derived in Section 4.1, is equivalent to (5), but admits a tighter relaxation and is more amenable to mathematical programming techniques. Section 4.2 introduces the convex-hull relaxation of this problem, which will be the main focus of our algorithmic approach.

4.1 Derivation of semidefinite formulation

We revisit (2) and this time do not apply the identity $\operatorname{diag}(\boldsymbol{Z})^2 = \boldsymbol{I}$ immediately. Then,

$$\begin{aligned}
Cov\left(\hat{\boldsymbol{\theta}}\right) &= \left(\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})^2 \boldsymbol{X} - \boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z}) \boldsymbol{X} \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z}) \boldsymbol{X}\right)^{-1} \\
&= \left(\boldsymbol{X}^\top \left(\operatorname{diag}(\boldsymbol{Z})^2 - \operatorname{diag}(\boldsymbol{Z}) \boldsymbol{X} \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\right) \boldsymbol{X}\right)^{-1} \\
&= \left(\boldsymbol{X}^\top \left(\operatorname{diag}(\boldsymbol{Z}) \boldsymbol{I} \operatorname{diag}(\boldsymbol{Z}) - \operatorname{diag}(\boldsymbol{Z}) \boldsymbol{X} \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z})\right) \boldsymbol{X}\right)^{-1} \\
&= \left(\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{Z}) \left(\boldsymbol{I} - \boldsymbol{X} \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top\right) \operatorname{diag}(\boldsymbol{Z}) \boldsymbol{X}\right)^{-1}
\end{aligned}$$

$$= \left(\boldsymbol{X}^\top \left(\boldsymbol{P} \circ \left(\boldsymbol{Z}\boldsymbol{Z}^\top\right)\right) \boldsymbol{X}\right)^{-1},$$

where $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top$ is the projection matrix onto the orthogonal complement of the column space of $\boldsymbol{X}$, and $\circ$ denotes the Hadamard (elementwise) product. This representation of $Cov\left(\hat{\boldsymbol{\theta}}\right)$ depends on $\boldsymbol{Z}$ only through the rank-1 matrix $\boldsymbol{Z}\boldsymbol{Z}^\top$, giving rise to the formulation

$$\max_{\boldsymbol{Z} \in \{\pm 1\}^n, \boldsymbol{Y}} \log\det\left(\boldsymbol{X}^\top \left(\boldsymbol{P} \circ \boldsymbol{Y}\right) \boldsymbol{X}\right)$$

subject to $\boldsymbol{Y} = \boldsymbol{Z}\boldsymbol{Z}^\top$ and, if we are budget-constrained,

$$\sum_{i=1}^{n} Z_i \leq 2k - n, \tag{11}$$

again needed only for the case $k < \lceil \frac{n}{2} \rceil$. We may rewrite the problem in the following equivalent form, which has a more obvious convex relaxation:

$$\max_{\boldsymbol{W}, \boldsymbol{Y} \succeq 0, \boldsymbol{Z} \in \{\pm 1\}^n} \log\det\left(\boldsymbol{X}^\top \left(\boldsymbol{P} \circ \boldsymbol{Y}\right) \boldsymbol{X}\right) \tag{12}$$

subject to

$$\boldsymbol{W} = \begin{pmatrix} 1 & \boldsymbol{Z}^\top \\ \boldsymbol{Z} & \boldsymbol{Y} \end{pmatrix}, \tag{13}$$

$$Y_{ii} = 1, \qquad i = 1, ..., n, \tag{14}$$

$$\operatorname{rank}(\boldsymbol{W}) = 1, \tag{15}$$

and also (11) in the budget-constrained case. Problem (12)-(15) is equivalent to (5), but since various constants were dropped from the derivations, the optimal values of the two problems differ by $\log\det\left(\boldsymbol{X}^\top \boldsymbol{X}\right) - 2d \log 2$. The budget constraints (6) and (11) are also equivalent.

Just as (5) has a natural relaxation (7), Problem (12)-(15) can be relaxed by simply removing (15) and dropping the requirement $\boldsymbol{Z} \in \{-1, 1\}^n$. Note that (13)-(14) together with $\boldsymbol{W} \succeq 0$ imply that $\boldsymbol{Y} \succeq \boldsymbol{Z}\boldsymbol{Z}^\top$, whence $Z_i^2 \leq Y_{ii} = 1$ holds without any other explicit constraints.

The continuous relaxation of Problem (12)-(15) is provably stronger than the relaxation (7) of the original formulation. The following result formalizes this notion.

Proposition 4.1. *Let V^* be the optimal value of the problem*

$$\max_{\boldsymbol{W}, \boldsymbol{Y} \succeq 0, \boldsymbol{Z}} \log\det\left(\boldsymbol{X}^\top \left(\boldsymbol{P} \circ \boldsymbol{Y}\right) \boldsymbol{X}\right) \tag{16}$$

subject to (13)-(14) and (11). Let V' be the optimal value of Problem (7) subject to (6). Then,

$$V^* \leq V' - \log\det\left(\boldsymbol{X}^\top \boldsymbol{X}\right) + 2d\log 2,$$

where the constant on the RHS is purely from the transformation rather than the optimality gap of either relaxation.

Lastly, both (12)-(15) and its relaxation can be simplified further, without the need to explicitly model $\boldsymbol{Z}$ at all, as shown in the following results. The simplified versions of the discrete and continuous problems look very similar, but the proofs require different arguments, so the two results are stated separately.

Proposition 4.2. *Problem (12)-(15) is equivalent to*

$$\max_{Y \succeq 0} \log\det\left(\boldsymbol{X}^\top \left(\boldsymbol{P} \circ \boldsymbol{Y}\right) \boldsymbol{X}\right) \tag{17}$$

subject to (14) and $\operatorname{rank}(\boldsymbol{Y}) = 1$*, as well as*

$$\mathbf{1}^\top \boldsymbol{Y} \mathbf{1} \geq (2k - n)^2. \tag{18}$$

if we are budget-constrained.

Proposition 4.3. *The continuous relaxation of Problem (12)-(15) is equivalent to (17) subject to (14), as well as (18) if we are budget-constrained. (In other words, the rank-1 constraint from Proposition 4.2 is dropped.)*

4.2 Convex-hull relaxation

The formulation given in Proposition 4.2 suggests that a tighter relaxation is possible. It is known (Boyd & Vandenberghe, 2004) that the tightest convexification of a region is its convex hull. Because our objective is a concave function, the convex-hull relaxation is not exact, but is tighter than the problem in Proposition 4.3, which itself was already tighter than the problem in (7). To describe the convex hull, recall the fact, used in the proof of Proposition 4.2, that any $\boldsymbol{Y} \succeq 0$ satisfying $\operatorname{rank}(\boldsymbol{Y}) = 1$ and $Y_{ii} = 1$ for all i can be written as $\boldsymbol{Y} = \boldsymbol{Z}\boldsymbol{Z}^\top$ for some $\boldsymbol{Z} \in \{-1, 1\}^n$. Therefore, the convex hull can be described as

$$\mathcal{C} = \operatorname{conv}\left\{\boldsymbol{Z}\boldsymbol{Z}^\top \mid \boldsymbol{Z} \in \{-1,1\}^n, \sum_{i=1}^n Z_i \leq 2k - n\right\}, \tag{19}$$

and for any $\boldsymbol{Y} \in \mathcal{C}$, we have $\boldsymbol{Y} = \sum_{r \in \mathcal{A}} w_r \boldsymbol{Z}_r \boldsymbol{Z}_r^\top$, where $\mathcal{A}$ is some finite index set, each $\boldsymbol{Z}_r$ satisfies the conditions of (19), and $w_r \geq 0$ with $\sum_{r \in \mathcal{A}} w_r = 1$. The condition $\sum_{i=1}^n Z_i \leq 2k - n$ is present only if we are budget-constrained. We thus have the semidefinite program

$$\max_{\boldsymbol{Y} \in \mathcal{C}} \log\det\left(\boldsymbol{X}^\top \left(\boldsymbol{P} \circ \boldsymbol{Y}\right) \boldsymbol{X}\right). \tag{20}$$

The tightness of (20) is related to the change of basis introduced in Section 3.3. The following result uses (9) to characterize one situation where the convex-hull relaxation is exact. A practical example of this situation is when the data consist of one-hot covariates.

Proposition 4.4. *Suppose that we are not budget-constrained. Let* $\boldsymbol{U} = \boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-\frac{1}{2}}$ *with* $\boldsymbol{u}_i^\top$ *being the* i*th row of* $\boldsymbol{U}$*. Assume that, for all* i, j*, we have* $\boldsymbol{u}_i \perp \boldsymbol{u}_j$ *or* $\boldsymbol{u}_i \parallel \boldsymbol{u}_j$*. Then, (20) is exact: that is, there exists* $\boldsymbol{Z}^* \in \{-1, 1\}^n$ *such that* $\boldsymbol{Y}^* = \boldsymbol{Z}^* \left(\boldsymbol{Z}^*\right)^\top$ *maximizes (20).*

When the data are "close" to the conditions of Proposition 4.4, the convex-hull relaxation will be "almost exact" in the sense of the following result. For example, if the rows of $\boldsymbol{U}$ can be clustered into d groups, and each group is close to one of d mutually orthogonal lines, then this result will apply.

Proposition 4.5. *Let* $\boldsymbol{U} = \boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-\frac{1}{2}}$*, and define*

$$\Phi_{\boldsymbol{X}} = \max_{\boldsymbol{Z} \in \{\pm 1\}^n} \log\det\left(\boldsymbol{X}^\top \left(\boldsymbol{P} \circ \boldsymbol{Z}\boldsymbol{Z}^\top\right) \boldsymbol{X}\right), \qquad \Phi_{\boldsymbol{X}}^{rel} = \max_{\boldsymbol{Y} \in \mathcal{C}} \log\det\left(\boldsymbol{X}^\top \left(\boldsymbol{P} \circ \boldsymbol{Y}\right) \boldsymbol{X}\right). \tag{21}$$

Let $\bar{\boldsymbol{X}} \in \mathbb{R}^{n \times d}$ *be a different design matrix, also with full column rank, and define* $\bar{\boldsymbol{U}}$*,* $\Phi_{\bar{\boldsymbol{X}}}$ *and* $\Phi_{\bar{\boldsymbol{X}}}^{rel}$ *analogously, using the projection matrix* $\bar{\boldsymbol{P}} = \boldsymbol{I} - \bar{\boldsymbol{U}}\bar{\boldsymbol{U}}^\top$*. Suppose that* $\bar{\boldsymbol{U}}$ *satisfies the conditions of Proposition 4.4, and define*

$$\delta\left(\boldsymbol{X}, \bar{\boldsymbol{X}}\right) = \sum_{i=1}^n \|\boldsymbol{u}_i \boldsymbol{u}_i^\top - \bar{\boldsymbol{u}}_i \bar{\boldsymbol{u}}_i^\top\|_2.$$

Then,

$$\left|\exp\left(\Phi_{\boldsymbol{X}} - \log\det\left(\boldsymbol{X}^\top \boldsymbol{X}\right)\right) - \exp\left(\Phi_{\bar{\boldsymbol{X}}} - \log\det\left(\bar{\boldsymbol{X}}^\top \bar{\boldsymbol{X}}\right)\right)\right| \leq 2d \cdot \delta\left(\boldsymbol{X}, \bar{\boldsymbol{X}}\right),$$
$$\left|\exp\left(\Phi_{\boldsymbol{X}}^{rel} - \log\det\left(\boldsymbol{X}^\top \boldsymbol{X}\right)\right) - \exp\left(\Phi_{\bar{\boldsymbol{X}}}^{rel} - \log\det\left(\bar{\boldsymbol{X}}^\top \bar{\boldsymbol{X}}\right)\right)\right| \leq 2d \cdot \delta\left(\boldsymbol{X}, \bar{\boldsymbol{X}}\right).$$

To summarize the results of this section, we now have a reformulation of D-optimal partitioning as a semidefinite integer program that is equivalent to (5), and two relaxations of this problem (natural and convex-hull) that improve on (7). Our algorithmic developments will primarily use the convex-hull relaxation, but the natural relaxation can be useful numerically to assess how much is gained by using the stronger upper bounds.

5 Column-generation algorithm

We propose a column-generation algorithm that can be used to solve (20) exactly. In the process, it generates feasible discrete solutions to (17)-(18) as well as upper bounds on the optimality gap of those solutions. The other relaxations, namely (7) subject to (6), and (16) subject to (13)-(14) and (11), can also be combined with exact methods and used as benchmarks, but the convex-hull relaxation is the tightest and yields the best performance. All three relaxations can also potentially be used to generate cutting planes for branch-and-cut methods.

Algorithm 1 belongs to the Frank-Wolfe class of methods (Hendrych et al., 2025). At a high level, we approximate (19) using a finite collection of binary $\boldsymbol{Z}$ vectors (atoms) that is iteratively updated. In each iteration, we solve the dual pricing problem (23) to obtain a new atom that improves the objective value, and terminate when no further improvement is possible. In the following, we discuss and justify each step.

For notational simplicity, let $f(\boldsymbol{Y}) = \log\det\left(\boldsymbol{X}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{X}\right)$ denote the objective function of (20). In Step 1, (22) is simply the gradient $\nabla f(\boldsymbol{Y}_\ell)$ of f with respect to $\boldsymbol{Y}$, evaluated at the candidate solution $\boldsymbol{Y}_\ell$. In Step 2, we consider the linear approximation

$$f(\boldsymbol{Y}) \approx f(\boldsymbol{Y}_\ell) + \langle \boldsymbol{G}_\ell, \boldsymbol{Y} - \boldsymbol{Y}_\ell\rangle \tag{25}$$

and optimize (25) over $\boldsymbol{Y} \in \mathcal{C}$. However, (25) is linear in $\boldsymbol{Y}$, and a linear function over a convex hull is maximized at an extreme point. Because all extreme points of $\mathcal{C}$ are of the form $\boldsymbol{Z}\boldsymbol{Z}^\top$ with $\boldsymbol{Z} \in \{\pm 1\}^n$, it is sufficient to maximize $\langle \boldsymbol{G}_\ell, \boldsymbol{Z}\boldsymbol{Z}^\top\rangle = \boldsymbol{Z}^\top\boldsymbol{G}_\ell\boldsymbol{Z}$ over $\boldsymbol{Z} \in \{\pm 1\}^n$, as in (23).

In Step 3, $g_{\ell+1}$ is the directional derivative if we move from the candidate $\boldsymbol{Y}_\ell$ toward the new atom found in Step 2, that is,

$$\begin{aligned}
\frac{d}{d\alpha} f\left((1-\alpha)\boldsymbol{Y}_\ell + \alpha\boldsymbol{Z}_{\ell+1}\boldsymbol{Z}_{\ell+1}^\top\right)\Big|_{\alpha=0} &= \langle \boldsymbol{G}_\ell, \boldsymbol{Z}_{\ell+1}\boldsymbol{Z}_{\ell+1}^\top - \boldsymbol{Y}_\ell\rangle & (26)\\
&= \langle \boldsymbol{G}_\ell, \boldsymbol{Z}_{\ell+1}\boldsymbol{Z}_{\ell+1}^\top\rangle - d & (27)\\
&= g_{\ell+1},
\end{aligned}$$

where (27) is obtained from

$$\langle \boldsymbol{Y}_\ell, \boldsymbol{G}_\ell\rangle = \langle \boldsymbol{Y}_\ell, \boldsymbol{P}\circ\left(\boldsymbol{X}\left(\boldsymbol{X}^\top(\boldsymbol{P}\circ\boldsymbol{Y}_\ell)\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top\right)\rangle$$

Algorithm 1 Column-generation algorithm for solving (20).

Step 0: Initialize $\ell = 0$, $\boldsymbol{Z}_0 \in \{\pm 1\}^n$ and set $\boldsymbol{Y}_0 = \boldsymbol{Z}_0\boldsymbol{Z}_0^\top$, and $\mathcal{A} = \{\boldsymbol{Z}_0\}$. Fix a small tolerance $\varepsilon > 0$.

Step 1: Calculate the gradient

$$\boldsymbol{G}_\ell = \boldsymbol{P} \circ \left(\boldsymbol{X}\left(\boldsymbol{X}^\top\left(\boldsymbol{P}\circ\boldsymbol{Y}_\ell\right)\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top\right). \tag{22}$$

Step 2: Solve the pricing problem

$$\max_{\boldsymbol{Z}\in\{\pm 1\}^n} \boldsymbol{Z}^\top\boldsymbol{G}_\ell\boldsymbol{Z} \tag{23}$$

subject to (11) if we are budget-constrained. Let $\boldsymbol{Z}_{\ell+1}$ be the optimal solution.

Step 3: Compute $g_{\ell+1} = \langle \boldsymbol{G}_\ell, \boldsymbol{Z}_{\ell+1}\boldsymbol{Z}_{\ell+1}^\top\rangle - d$. If $g_{\ell+1} < \varepsilon$, terminate the procedure; otherwise, continue.

Step 4: Update $\mathcal{A} \leftarrow \mathcal{A} \cup \{\boldsymbol{Z}_{\ell+1}\}$ and solve

$$\max_{w\geq 0:\sum_j w_j = 1} \log\det\left(\boldsymbol{X}^\top\left(\boldsymbol{P}\circ\sum_{j=0}^{|\mathcal{A}|} w_j\boldsymbol{Z}_j\boldsymbol{Z}_j^\top\right)\boldsymbol{X}\right). \tag{24}$$

Let w' be the optimal solution.

Step 5: Update $\boldsymbol{Y}_{\ell+1} = \sum_{j=0}^{|\mathcal{A}|} w'_j\boldsymbol{Z}_j\boldsymbol{Z}_j^\top$. Increment $\ell \leftarrow \ell + 1$ and return to Step 1.

$$\begin{aligned}
&= \langle \boldsymbol{X}^\top\left(\boldsymbol{P}\circ\boldsymbol{Y}_\ell\right)\boldsymbol{X}, \left(\boldsymbol{X}^\top\left(\boldsymbol{P}\circ\boldsymbol{Y}_\ell\right)\boldsymbol{X}\right)^{-1}\rangle \\
&= \operatorname{tr}\left(\boldsymbol{I}_d\right) \\
&= d.
\end{aligned}$$

The following result shows that this quantity is always nonnegative; therefore, if $g_{\ell+1} = 0$, no improvement is possible and $\boldsymbol{Y}_\ell$ is optimal. Furthermore, $g_{\ell+1}$ can be used to compute an upper bound on the optimal value of the D-optimal partitioning problem.

Proposition 5.1. *For any iteration $\ell \geq 0$ of Algorithm 1, we have $g_{\ell+1} \geq 0$ and*

$$\max_{\boldsymbol{Z}\in\{\pm 1\}^n} \log\det\left(\boldsymbol{X}^\top\left(\boldsymbol{P}\circ\boldsymbol{Z}\boldsymbol{Z}^\top\right)\boldsymbol{X}\right) \leq \log\det\left(\boldsymbol{X}^\top\left(\boldsymbol{P}\circ\boldsymbol{Y}_\ell\right)\boldsymbol{X}\right) + g_{\ell+1}, \tag{28}$$

with the LHS additionally subject to (11) if we are budget-constrained.

Finally, Steps 4-5 represent a fully corrective version of Frank-Wolfe. Instead of performing a line search between $\boldsymbol{Y}_\ell$ and the new atom, we instead construct $\boldsymbol{Y}_{\ell+1}$ from an optimal mixture of all currently available atoms, including the new one. Potentially Step 4 can also drop old atoms by assigning zero weight to them.

The most computationally intensive part of the procedure is Step 2. The pricing problem (23) is an instance of MAX-CUT, and therefore hard to solve. This difficulty is very typical of column generation methods (Savelsbergh, 2024). It should be noted, however, that it is not necessary to solve (23) to optimality in every iteration. In order to proceed to the next iteration, it is sufficient to obtain any $\boldsymbol{Z}_{\ell+1}$ that violates the termination criterion of Step 3. Thus, strictly speaking, one only requires an optimal solution of the pricing problem in the very last iteration, to certify that no further improvement is possible. In practice, the MAX-CUT problem can be handled directly by a commercial solver such as Gurobi, and one can program the solver to stop as soon as an improvement is found. One can also use heuristics to find an improving solution.

A widely used heuristic, appearing in Li et al. (2024) and many other papers, is local search. Given $\boldsymbol{Z} \in \{-1, 1\}$, we flip the sign of a single element Z_i if doing so improves the value of $\boldsymbol{Z}^\top \boldsymbol{G}_\ell \boldsymbol{Z}$, and repeat until no further improvement is possible from a single sign change, in which case $\boldsymbol{Z}$ is said to be "locally optimal." Local search can be performed on any solution obtained through any other means (e.g., from Gurobi or other heuristics), so it is always possible to ensure local optimality.

For the reasons described in Section 3.2, it is not possible to prove a general performance guarantee for local search. However, some performance bounds are possible if we impose further restrictions on the leverage scores introduced in Section 3.3. This case is explored in the next result.

Proposition 5.2. *Suppose that we are not budget-constrained. Define $\boldsymbol{u}_i$ as in (9) and suppose that $\boldsymbol{u}_i^\top \boldsymbol{u}_i \leq L < 1$ for all i. Let $\tilde{\boldsymbol{Z}} \in \{\pm 1\}$ be a locally optimal solution for the ℓth iteration of (23). Then,*

$$\max_{\boldsymbol{Z} \in \{\pm 1\}^n} \boldsymbol{Z}^\top \boldsymbol{G}_\ell \boldsymbol{Z} \leq \frac{1}{1-L} \tilde{\boldsymbol{Z}}^\top \boldsymbol{G}_\ell \tilde{\boldsymbol{Z}}. \tag{29}$$

From (28), it follows that $\frac{1}{1-L} \tilde{\boldsymbol{Z}}^\top \boldsymbol{G}_\ell \tilde{\boldsymbol{Z}}$ can serve as a worst-case upper bound on the optimal value of the original discrete D-optimal partitioning problem. At the same time, $\tilde{\boldsymbol{Z}}$ itself (and really any $\boldsymbol{Z}_\ell$ obtained through Algorithm 1) is feasible and therefore a valid lower bound. Thus, while column generation can be integrated with branch-and-price methodology to solve D-optimal partitioning exactly, it is also possible to simply keep track of the best among the discrete solutions obtained in Step 2 of Algorithm 1. In our experience, these solutions are extremely competitive with exact discrete solution techniques and can be obtained much more quickly. In this way, if it is too computationally expensive to run Algorithm 1 to completion, it is still possible, and very practical, to use the best lower bound obtained within some acceptable length of time.

6 Experiments and case studies

We compare our proposed methodology against several benchmarks on both synthetic and real data. Our experiments pursue several goals. First, we demonstrate the value of optimization-based treatment assignments relative to standard randomization-based assignments, using numerical performance comparisons and visualizations to obtain insight. Second, we evaluate the tightness of the various relaxations that have been discussed. Third, we demonstrate the practical potential of our approach on moderately sized, realistic applications using datasets from the literature. The problem size is limited in part by the need to compare against benchmark algorithms, some of which use computationally expensive exact solution techniques.

Section 6.1 describes the benchmark methods implemented in our study. Section 6.2 presents numerical results for synthetic data, and Sections 6.3-6.4 consider real data related to applications from finance and education policy. Due to space considerations, Sections 6.3-6.4 focus more on qualitative insights obtainable from visualizations, but the specific numerical results can be found in the Appendix.

6.1 Description of benchmark methods

Each method is given a design matrix $\boldsymbol{X}$ and a budget k if we wish to impose one, and returns a vector $\boldsymbol{Z} \in \{\pm 1\}^n$. The specifications of the data will be discussed later. First, we explain how each method computes the treatment decision.

Randomized control trials. We consider two versions of randomization: simple coin-flipping, where each Z_i is chosen independently of the others and the probabilities are chosen to make the expected number of treated individuals equal to the desired value (k in the budget-constrained case, and $\frac{n}{2}$ in the unconstrained case); and pivotal sampling (Deville & Tille, 1998) where the number of treated individuals is fixed to always equal the desired value. In each case study, we run each version 10 times and report the best-performing assignment.

Conic optimization. Sagnol & Harman (2015) proved that the function $\det^{\frac{1}{d}}\left(\sum_i z_i \boldsymbol{x}_i \boldsymbol{x}_i^\top\right)$ has a mixed-integer second-order cone representation, for which one can use specialized functionality in standard commercial software packages. It is relatively straightforward to extend this logic to the two-log-determinant formulation (5) of D-optimal partitioning. The software will return a feasible solution (lower bound) as well as an upper bound on the optimal value.

Row generation (RG). We can approximate the objective in (5) with cuts similar to those used in Li et al. (2024) and Wang et al. (2025). These cuts are easy to compute and can be handled using lazy constraint callback in Gurobi. They will solve the problem exactly given enough time, but improvement can be slow.

We also consider an "improved" variant of RG that additionally solves the convex-hull relaxation (20). Let $\boldsymbol{Y}^*$ be the optimal solution to the relaxation, and let $\boldsymbol{G}^*$ be the corresponding gradient. Then,

$$w \leq \log\det\left(\boldsymbol{X}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{X}\right) + \boldsymbol{Z}^\top \boldsymbol{G}^* \boldsymbol{Z} - d,$$

where w is an artificial variable standing in for the optimal objective value, is a valid gradient cut. To convert it to the two-log-determinant problem, we use $\boldsymbol{Z} = 2\boldsymbol{z} - 1$ and linearize using a McCormick envelope (McCormick, 1976). In our experiments, we used the convex-hull relaxation to generate $\boldsymbol{Y}^*$. Consequently, improved RG essentially adds another layer on top of CG and always has a higher computational cost.

Column generation (CG). We run Algorithm 1 and terminate as soon as one iteration of the pricing problem has running time greater than 5 seconds. The best-performing $\boldsymbol{Z}_\ell$ among all the iterations that we ran is returned. The 5-second rule may cause this method to stop running before the convex-hull relaxation has been solved to optimality, but we enforce it in order to show that CG can often obtain very good solutions in a very short time.

Each method was forced to terminate after 1 hour if it had not done so already; in this case, the best-found feasible solution was returned. In addition to the above-described methods, we also solved three continuous relaxations: Problem (7), Problem (17) subject to (14), and Problem (20), all with or without budget constraints as needed.

		LB	UB	Time (s)
Relaxations	Two-log-determinant	N/A	39.545	<0.01
	SDP (Prop. 4.3)	N/A	37.662	7.33
	Convex-hull	N/A	37.513	1183
Assignments	Coin-flipping	35.157	N/A	<0.01
	Pivotal sampling	35.504	N/A	<0.01
	Conic optimization	37.360	37.712	3600
	Row generation	36.843	38.517	3600
	Improved RG	37.406	37.513	3600
	Best CG solution	37.406	N/A	14.14

Table 1: Performance comparison for synthetic data with $n = 101$ and $d = 15$, no budget constraint.

6.2 Experiments on synthetic data

We generated datasets in which $n-1$ rows are generated uniformly on the d-dimensional unit sphere, and 1 row is an outlier generated on the sphere with norm 10. We considered both constrained ($k = 20$) and unconstrained versions of the problem. This construction is used as a stress test for the methods under an extreme case of covariate heterogeneity, with one high-norm outlier that creates an inevitable imbalance in the two terms of (5). This setting is challenging for randomized methods because placing the outlier without proper consideration for the other vectors may substantially exacerbate the imbalance.

Tables 1-2 report numerical results for $n = 101$ and $n = 1001$, respectively. These two instances have no budget constraint. We see that the relaxation (7) of the two-log-determinant objective is extremely fast but can be loose, whereas the SDP relaxation is tighter but becomes memory-intensive as n increases. The convex-hull relaxation produces the tightest bounds and remains usable even for the largest instance considered.

Among the assignment rules, the best performance is obtained by simply running CG and reporting the best $\boldsymbol{Z}_\ell$ obtained in the process of solving the convex-hull relaxation. The improved RG method, which also solves the convex-hull relaxation to create an initial cut, is essentially unable to improve on what is offered by CG alone even if allowed to run until the end of the hour. The other methods do not perform as well; while the conic method is the best among them for $n = 101$, it is not able to find a feasible solution for the large instance within one hour. We note that, in Table 2, the convex-hull relaxation is solved exactly with every iteration of the pricing problem taking less than 5 seconds, so the upper bounds reported for the relaxation, for CG, and

		LB	UB	Time (s)
Relaxations	Two-log-determinant	N/A	106.747	<0.01
	SDP (Prop. 4.3)	N/A	OOM	N/A
	Convex-hull	N/A	106.689	387.02
Assignments	Coin-flipping	106.119	N/A	<0.01
	Pivotal sampling	106.189	N/A	<0.01
	Conic optimization	N/A	106.747	3600
	Row generation	106.620	106.729	3600
	Improved RG	106.681	106.689	3600
	Best CG solution	106.681	106.689	387.02

Table 2: Performance comparison for synthetic data with $n = 1001$ and $d = 15$, no budget constraint. "OOM" stands for "out of memory."

for improved RG all match. In Table 1, the five-second rule causes our CG assignment rule to stop early (after only 14 seconds) without obtaining a good upper bound. However, the actual solution obtained in this short time is as good as the best solution obtained from improved RG. If one were to spend the full 1183 seconds to solve the convex-hull relaxation to optimality, one would see that this solution is near-optimal.

Tables 3-4 present analogous comparisons for the constrained case with $k = 20$. Overall the results are similar, except objective values are lower across the board (due to the additional constraint) while computation times are higher. Once more, running RG after solving the convex-hull relaxation essentially offers no improvement over simply reporting the best discrete solution obtained from pricing.

		LB	UB	Time (s)
Relaxations	Two-log-determinant	N/A	33.755	0.32
	SDP (Prop. 4.3)	N/A	31.688	8.36
	Convex-hull	N/A	31.270	971.42
Assignments	Coin-flipping	25.156	N/A	<0.01
	Pivotal sampling	24.643	N/A	<0.01
	Conic optimization	30.549	33.490	3600
	Row generation	28.176	36.525	3600
	Improved RG	30.682	31.270	3600
	Best CG solution	30.682	31.270	971.42

Table 3: Performance comparison for synthetic data with $n = 101$, $d = 15$, and $k = 20$.

		LB	UB	Time (s)
Relaxations	Two-log-determinant	N/A	71.677	3.32
	SDP (Prop. 4.3)	N/A	OOM	N/A
	Convex-hull	N/A	71.367	733.30
Assignments	Coin-flipping	61.378	N/A	<0.01
	Pivotal sampling	63.577	N/A	<0.01
	Conic optimization	N/A	71.671	3600
	Row generation	63.692	81.353	3600
	Improved RG	70.291	71.367	3600
	Best CG solution	70.291	71.367	733.30

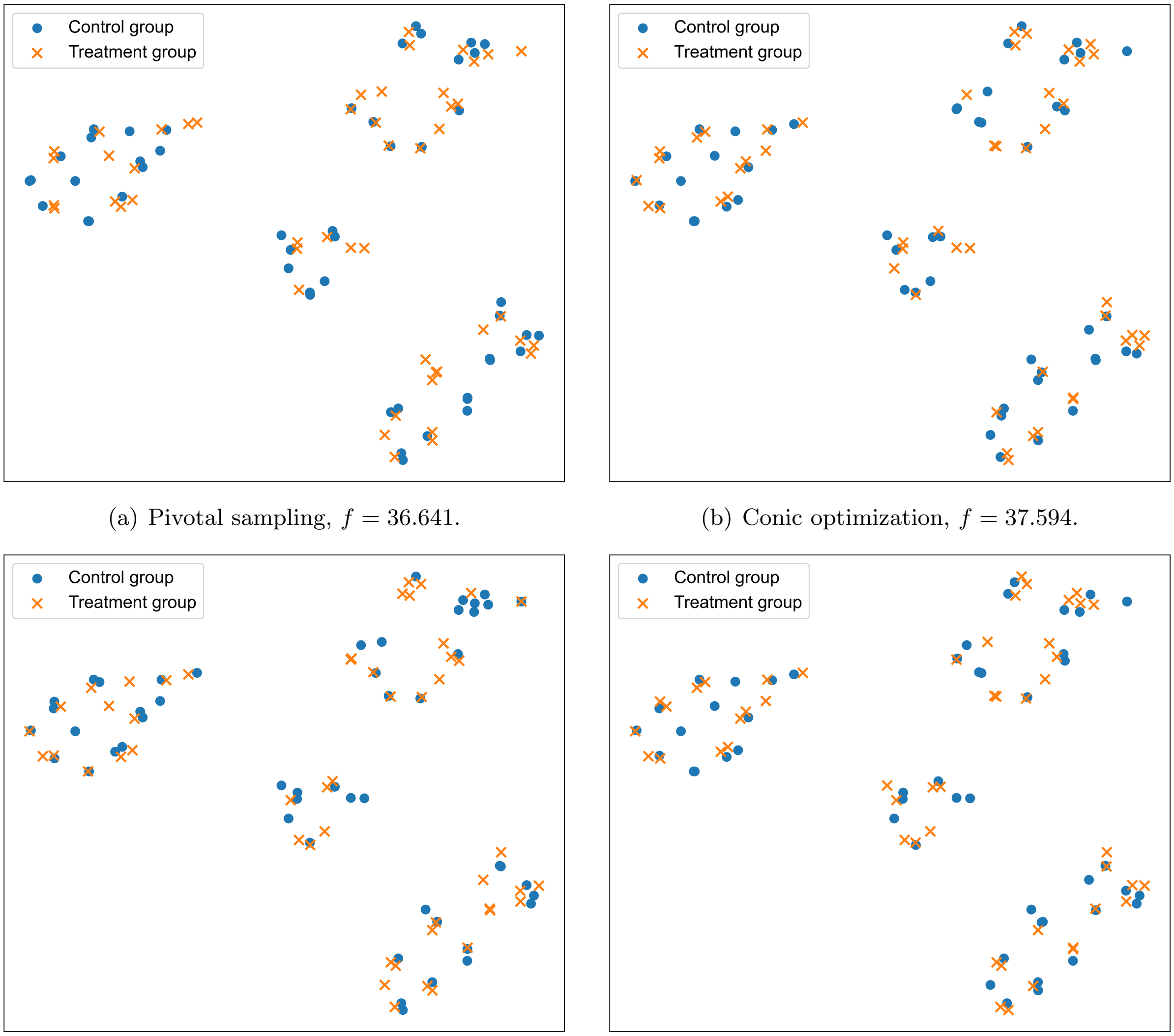
Table 4: Performance comparison for synthetic data with $n = 1001$, $d = 15$, and $k = 20$. "OOM" stands for "out of memory."

6.3 Experiments on 401(k) pension data

We ran experiments using a real-data benchmark originally studied by Chernozhukov & Hansen (2004). This dataset describes 401(k) retirement accounts for individuals with heterogeneous attributes such as age, income, family size, education, marital status, two-earner status, defined benefit (DB) pension status, IRA participation status, and home ownership. Education is a categorical variable with four groups. Thus, we have $d = 12$ features including both continuous and discrete variables. From an experimental design perspective, an employer might consider a pilot program for a particular type of retirement account. The treatment decision is whether to offer access to the plan to an individual, and the goal is to estimate the effect of the opportunity on wealth. The effect varies across different observed attributes of individuals.

We created a design matrix $\boldsymbol{X}$ by sampling $n = 100$ individuals without replacement from the dataset. As before, we consider both unconstrained and constrained (with $k = 15$) versions of the problem. Due to space considerations, the relevant tables of numbers are moved to the Appendix. In brief, however, the situation is quite similar to what we saw in Section 6.2, with CG producing the best-performing solutions. Perhaps the only difference is that now the improved RG method is able to marginally improve the upper bound (but not the lower bound) over the convex-hull relaxation.

Rather than repeat similar-looking tables, we aim instead to obtain visual insight into the treatment assignments made by various methods. Figure 1 uses the t-SNE method of van der Maaten & Hinton (2008) to project the 12-dimensional data vectors $\boldsymbol{x}_i$ onto a two-dimensional region. Similar vectors appear closer together in the figure, providing a visual representation of the



(a) Pivotal sampling, $f = 36.641$.

(b) Conic optimization, $f = 37.594$.

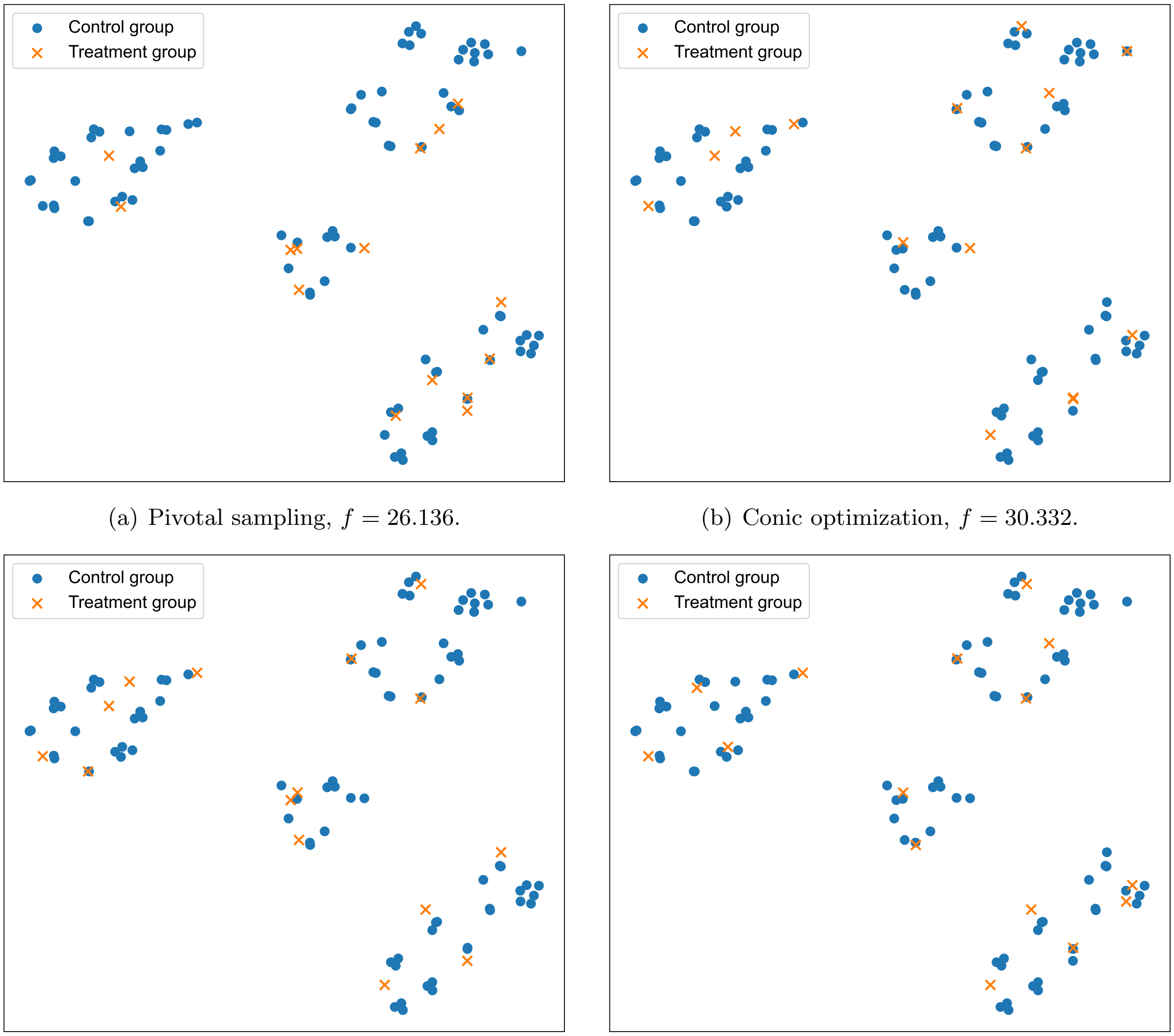
(c) Row generation, $f = 37.306$.

(d) Best CG solution, $f = 37.666$.

Figure 1: Two-dimensional visualizations of treatment assignments (unconstrained) for 401(k) pension data. The objective value f achieved by each method is reported.

empirical distribution of the data. For each method, we visually distinguish between treatment and control assignments made in the unconstrained setting; due to space considerations, we omit the coin-flipping method (which has the worst performance among all) as well as improved RG (which makes the same assignment decisions as CG) from the figure.

We see that the vectors can be grouped into four distinct clusters. Because we are considering the unconstrained setting, the precision of the estimated treatment effect will improve when we treat approximately half of the individuals in each cluster, though the precise number of treated individuals is also affected by the shape of the cluster (i.e., the data distribution within it). The main point is that it is not sufficient to simply treat half of all individuals without considering the



(a) Pivotal sampling, $f = 26.136$.

(b) Conic optimization, $f = 30.332$.

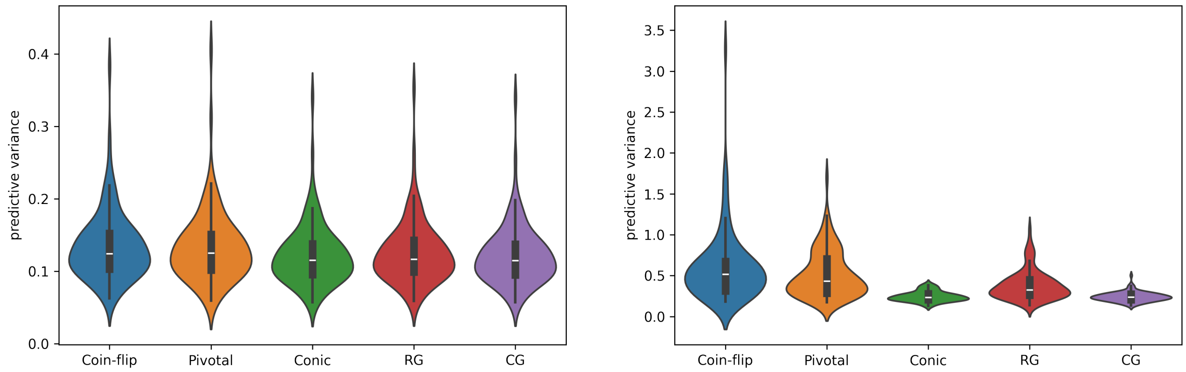
(c) Row generation, $f = 28.722$.

(d) Best CG solution, $f = 30.392$.

Figure 2: Two-dimensional visualizations of treatment assignments ($k = 15$) for 401(k) pension data. The objective value f achieved by each method is reported.

clusters: for example, in Figure 1(a), we see that pivotal sampling over-treats the cluster in the top right, though the total number of individuals treated by that method is exactly $\frac{n}{2}$. Similarly, RG over-treats the cluster in the bottom-right. Moreover, the distribution of the treated individuals within each cluster also makes a difference, as can be seen by comparing the left-most cluster under RG and CG.

Figure 2 presents an analogous visualization for the budget-constrained version of the same instance with $k = 15$. In this case, because the budget is so limited, it becomes even more important to divide the effort between clusters and to spread out the treated individuals within each cluster to maximize the information that is collected. Comparing the left-most cluster under



(a) Unconstrained.

(b) Constrained, $k = 15$.

Figure 3: Empirical distributions of V_i under each treatment assignment for 401(k) pension data.

RG and CG, we see that CG does a better job of spreading out the treatment assignments.

In the setting of A/B testing, one metric of interest is the predictive variance of the heterogeneous treatment effect. For the ith individual, this is the quantity $V_i = \boldsymbol{x}_i^\top Cov\left(\hat{\boldsymbol{\theta}}\right)\boldsymbol{x}_i$, where $Cov\left(\hat{\boldsymbol{\theta}}\right)$ is as derived in (3), with $\boldsymbol{Z}$ being the treatment assignment made by a particular method. In words, this quantity measures the accuracy of our estimate of the treatment effect for an individual, regardless of whether the individual is ever treated. In the context of linear regression, this metric is deterministic and may be computed without having to conduct any experiments. We compute V_i for each i under each treatment assignment and visualize its empirical distribution in Figure 3. Improved RG is omitted because it returns the same solution as CG.

The differences between methods are more pronounced in the constrained setting, where the number of treated individuals is smaller and it becomes critical to allocate treatments carefully. Consistently with Figures 1-2, where CG and conic optimization were the two top-performing methods, we also see that these same two assignments produce the smallest predictive variances, with the distribution for CG being more sharply peaked in Figure 3(b). It is clear that optimization-based approaches substantially reduce predictive uncertainty relative to randomized control trials.

6.4 Experiments on STAR data

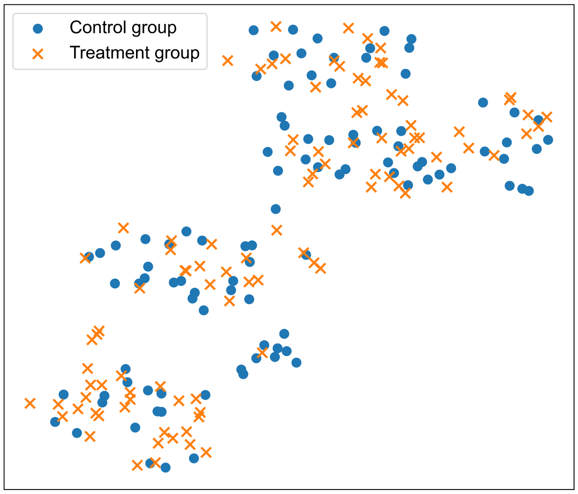
Project STAR was a four-year longitudinal class-size study funded by the Tennessee General Assembly and conducted in the late 1980s by the State Department of Education (Stock & Watson, 2015). The dataset contained information about student performance in reading and math from kindergarten through third grade, as well as attributes describing student gender and ethnicity, whether students qualified for a free lunch, school type (inner-city, suburban, etc.), and teacher education level and experience. The treatment is whether students are assigned to a regular class with 22-25 students or a smaller class with 13-17 students, and the goal is to assess the effect of class size on performance. By merging some categories of features and focusing on a portion of the multi-year time horizon, we obtain $d = 35$ dimensions of interest. Owing to the higher dimensionality, we now sample $n = 200$ individuals without replacement to construct $\boldsymbol{X}$.

As in Section 6.3, we defer the tables of numbers to the Appendix; in brief, however, the situation is similar to the previous examples, with CG yielding the best performance. We consider both unconstrained and constrained ($k = 40$) variants. For the constrained instance, conic optimization was not able to find a feasible solution (similar to the large instance in Section 6.2).

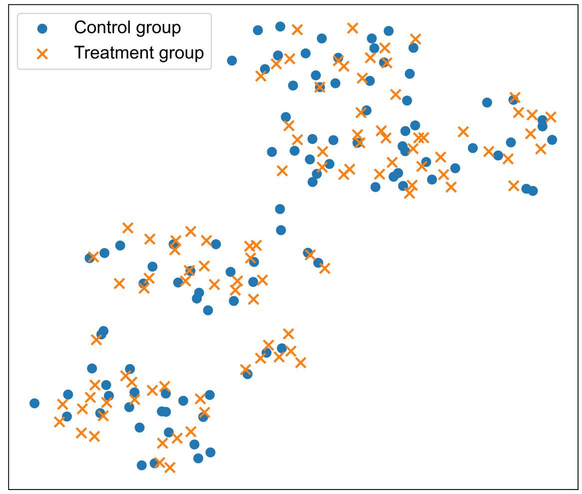
Figure 4 presents t-SNE visualizations for the unconstrained variant. This time, the data points group into three distinct clusters, with the top-right cluster being the largest and most distinct from the others. We can see that all of the methods are attempting to distribute the treated individuals evenly. Roughly speaking, the ideal assignment will not leave a group of similar individuals (close together in the visual) in which everyone is treated or everyone is untreated. Thus, we can see that pivotal sampling treats too many individuals in the lower-left cluster, while conic optimization treats too many in the center, while CG has a more even mixture of treated individuals interspersed among untreated ones.

Figure 5 presents analogous visuals for the constrained case. This time, we omit pivotal sampling because it significantly underperformed, as well as conic optimization because it never found a feasible solution. Neither RG nor CG is able to adequately cover the entire dataset with only 40 samples, so both methods have to make tradeoffs. The main difference is that RG prefers to cover more of the top-right cluster at the expense of the center cluster, whereas CG makes the opposite choice. It is worth noting that, although RG treats more individuals in the top right, the treated individuals are close to each other and a large part of the cluster remains uncovered. Thus, RG may simply not be obtaining enough value from the additional treatments.

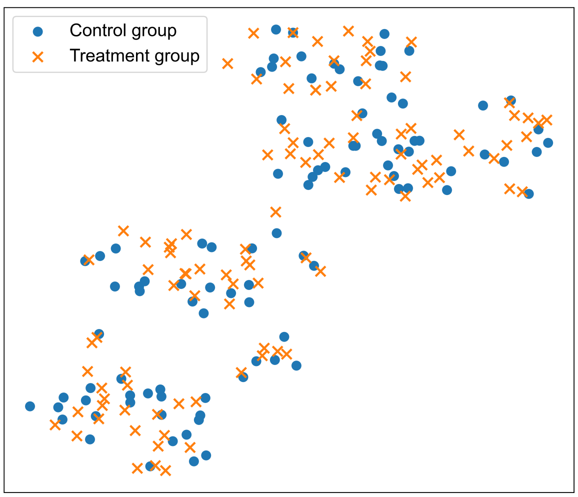
Finally, Figure 6 visualizes the empirical distributions of the predictive variances V_i under each method, for both unconstrained and constrained ($k = 40$) variants. For the unconstrained case, Figure 6(a) shows the same five methods as Figure 3 for the pension data. Here, CG has some benefit in reducing the upper tail of the distribution, but all five methods benefit from the larger



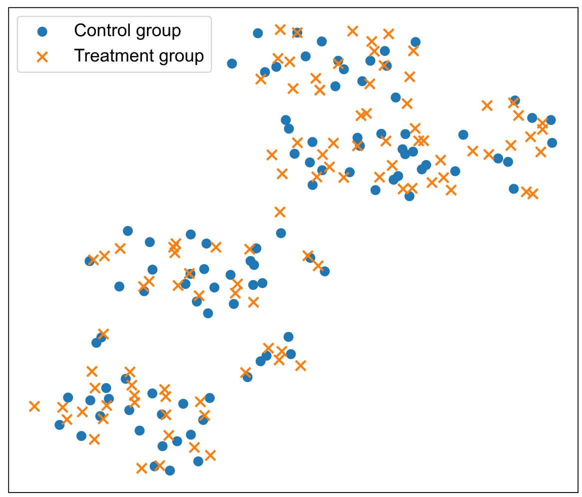
(a) Pivotal sampling, $f = 136.238$.



(b) Conic optimization, $f = 127.368$.



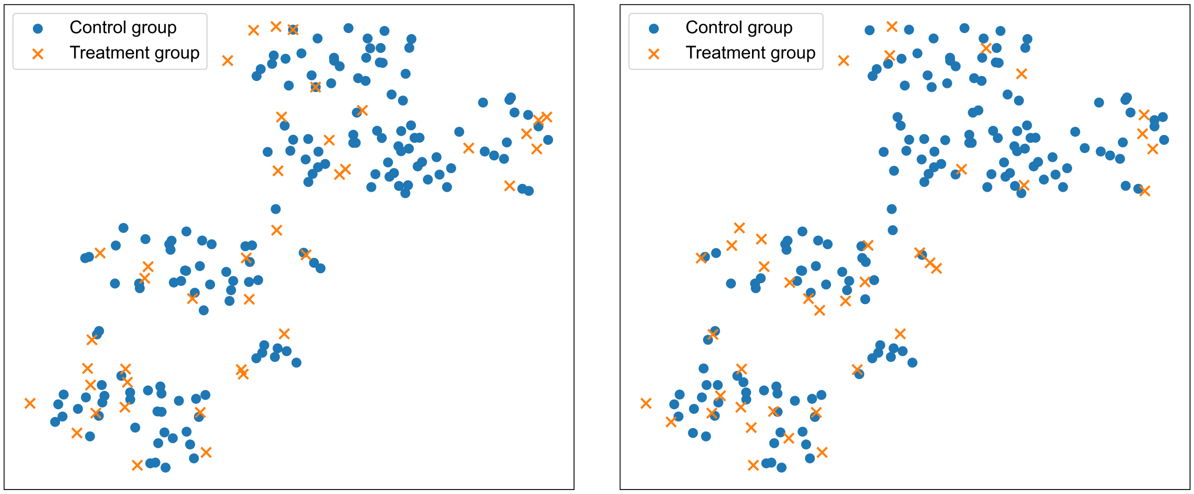
(c) Row generation, $f = 137.842$.



(d) Best CG solution, $f = 140.703$.

Figure 4: Two-dimensional visualizations of treatment assignments (unconstrained) for STAR data. The objective value f achieved by each method is reported.

sample size. For the constrained case, we omit the two randomized methods because they both performed very poorly and their predictive variances are vastly larger than for RG or CG. Conic optimization is also omitted because that method was not able to find a feasible solution. Thus, the remaining comparison is between RG and CG, and we see that CG is far more effective in reducing predictive variance.

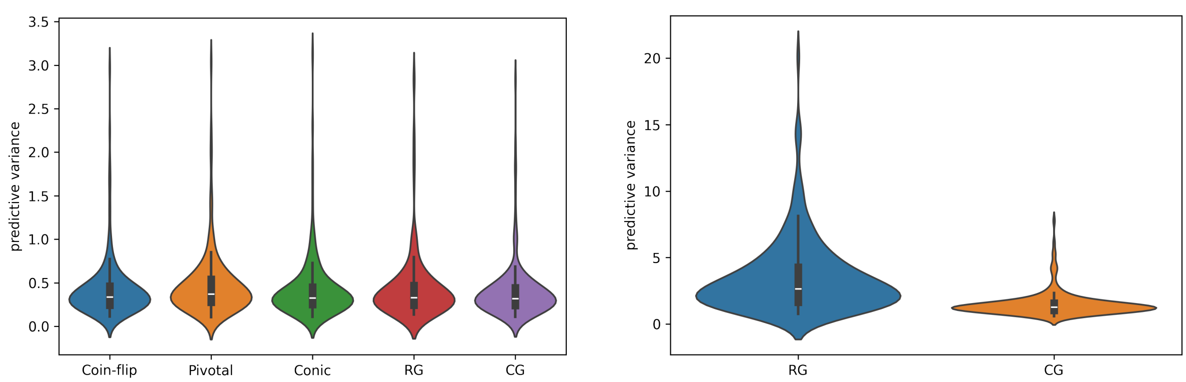


(a) Row generation, $f = 115.338$.
(b) Best CG solution, $f = 125.233$.

Figure 5: Two-dimensional visualizations of treatment assignments ($k = 40$) for STAR data. The objective value f achieved by each method is reported.

7 Conclusion

This paper has studied optimal experimental design in the presence of heterogeneous treatment effects. We have shown that, in this context, the classical D-optimality criterion from the statistical literature has a novel representation as the D-optimal partitioning problem, which aims to balance the information matrices for the treatment and control group, rather than simply maximizing the information value of one group or the other. This balancing structure makes the problem much more difficult: unlike classical D-optimal design, it does not admit a polynomial-time additive approximation algorithm. This difficulty motivated the development of a semidefinite programming



(a) Unconstrained.
(b) Constrained, $k = 40$.

Figure 6: Empirical distributions of V_i under each treatment assignment for STAR data.

reformulation that is amenable to a column generation approach. CG can be integrated into exact solution frameworks, but it also produces valid lower and upper bounds on its own, and we find that these are highly competitive with benchmark exact methods.

The limitations of our work are fairly typical for column generation algorithms. Like many such procedures in other contexts, our approach has to solve a difficult pricing problem. The development of effective heuristics for this problem could be one avenue for future work, though in light of our hardness result, it is unlikely that such techniques would admit provable guarantees without additional assumptions, e.g., on the leverage scores. A more general direction is the continued development of optimal design methods for more complex statistical and econometric models. Our work here highlights some of the challenges that may arise in such settings.

References

Alimonti, P. & Kann, V. (2000), 'Some APX-completeness results for cubic graphs', *Theoretical Computer Science* **237**(1-2), 123–134.

Allen-Zhu, Z., Li, Y., Singh, A. & Wang, Y. (2021), 'Near-optimal discrete optimization for experimental design: A regret minimization approach', *Mathematical Programming* **186**(1), 439–478.

Athey, S. & Imbens, G. (2016), 'Recursive partitioning for heterogeneous causal effects', *Proceedings of the National Academy of Sciences* **113**(27), 7353–7360.

Athey, S. & Imbens, G. W. (2017), The econometrics of randomized experiments, *in* A. V. Banerjee & E. Duflo, eds, 'Handbook of Economic Field Experiments', Vol. 1, Elsevier, pp. 73–140.

Bansal, N., Jiang, H. & Meka, R. (2023), Resolving matrix Spencer conjecture up to polylogarithmic rank, *in* B. Saha, ed., 'Proceedings of the 55th Annual ACM Symposium on Theory of Computing', pp. 1814–1819.

Bastani, H., Bayati, M. & Khosravi, K. (2021), 'Mostly exploration-free algorithms for contextual bandits', *Management Science* **67**(3), 1329–1349.

Bertsimas, D., Johnson, M. & Kallus, N. (2015), 'The power of optimization over randomization in designing experiments involving small samples', *Operations Research* **63**(4), 868–876.

Bhat, N., Farias, V. F., Moallemi, C. C. & Sinha, D. (2020), 'Near-optimal A-B testing', *Management Science* **66**(10), 4477–4495.

Bonesrønning, H., Finseraas, H., Hardoy, I., Iversen, J. M. V., Nyhus, O. H., Opheim, V., Salvanes, K. V., Sandsør, A. M. J. & Schøne, P. (2022), 'Small-group instruction to improve student performance in mathematics in early grades: Results from a randomized field experiment', *Journal of Public Economics* **216**, 104765.

Boyd, S. & Vandenberghe, L. (2004), *Convex Optimization*, Cambridge University Press.

Chernozhukov, V. & Hansen, C. (2004), 'The effects of 401(k) participation on the wealth distribution: an instrumental quantile regression analysis', *The Review of Economics and Statistics* **86**(3), 735–751.

Coey, C., Kapelevich, L. & Vielma, J. P. (2023), 'Conic optimization with spectral functions on Euclidean Jordan algebras', *Mathematics of Operations Research* **48**(4), 1906–1933.

Costa, L., Contardo, C. & Desaulniers, G. (2019), 'Exact branch-price-and-cut algorithms for vehicle routing', *Transportation Science* **53**(4), 946–985.

Crescenzi, P., Kann, V., Silvestri, R. & Trevisan, L. (1999), 'Structure in approximation classes', *SIAM Journal on Computing* **28**(5), 1759–1782.

Dadush, D., Jiang, H. & Reis, V. (2022), A new framework for matrix discrepancy: Partial coloring bounds via mirror descent, *in* S. Leonardi, ed., 'Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing', pp. 649–658.

Deng, A., Du, M., Matlin, A. & Zhang, Q. (2023), Variance reduction using in-experiment data: Efficient and targeted online measurement for sparse and delayed outcomes, *in* A. Singh & Y. Sun, eds, 'Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining', pp. 3937–3946.

Desaulniers, G., Pecin, D. & Contardo, C. (2019), 'Selective pricing in branch-price-and-cut algorithms for vehicle routing', *EURO Journal on Transportation and Logistics* **8**(2), 147–168.

Desrosiers, J. & Lübbecke, M. E. (2005), A primer in column generation, *in* G. Desaulniers, J. Desrosiers & M. M. Solomon, eds, 'Column generation', Springer, pp. 1–32.

Deville, J.-C. & Tille, Y. (1998), 'Unequal probability sampling without replacement through a splitting method', *Biometrika* **85**(1), 89–101.

Edwards, C. S. (1973), ‘Some extremal properties of bipartite subgraphs’, *Canadian Journal of Mathematics* **25**(3), 475–485.

Gilmour, S. G. & Trinca, L. A. (2012), ‘Optimum design of experiments for statistical inference’, *Journal of the Royal Statistical Society* **C61**(3), 345–401.

Harshaw, C., Sävje, F., Spielman, D. A. & Zhang, P. (2024), ‘Balancing covariates in randomized experiments with the Gram–Schmidt walk design’, *Journal of the American Statistical Association* **119**(548), 2934–2946.

Håstad, J. (2001), ‘Some optimal inapproximability results’, *Journal of the ACM* **48**(4), 798–859.

Hendrych, D., Troppens, H., Besançon, M. & Pokutta, S. (2025), ‘Convex mixed-integer optimization with Frank–Wolfe methods’, *Mathematical Programming Computation* **17**(4), 731–757.

Huan, X., Jagalur, J. & Marzouk, Y. (2024), ‘Optimal experimental design: Formulations and computations’, *Acta Numerica* **33**, 715–840.

Imai, K. & Strauss, A. (2011), ‘Estimation of heterogeneous treatment effects from randomized experiments, with application to the optimal planning of the get-out-the-vote campaign’, *Political Analysis* **19**(1), 1–19.

Kohavi, R., Tang, D. & Xu, Y. (2020), *Trustworthy online controlled experiments: A practical guide to A/B testing*, Cambridge University Press.

Konstantopoulos, S. (2008), ‘Do small classes reduce the achievement gap between low and high achievers? Evidence from Project STAR’, *The Elementary School Journal* **108**(4), 275–291.

Leng, Y. & Dimmery, D. (2024), ‘Calibration of heterogeneous treatment effects in randomized experiments’, *Information Systems Research* **35**(4), 1721–1742.

Li, F., Morgan, K. L. & Zaslavsky, A. M. (2018), ‘Balancing covariates via propensity score weighting’, *Journal of the American Statistical Association* **113**(521), 390–400.

Li, J. Q., Rusmevichientong, P., Simester, D., Tsitsiklis, J. N. & Zoumpoulis, S. I. (2015), ‘The value of field experiments’, *Management Science* **61**(7), 1722–1740.

Li, Y., Fampa, M., Lee, J., Qiu, F., Xie, W. & Yao, R. (2024), ‘D-optimal data fusion: Exact and approximation algorithms’, *INFORMS Journal on Computing* **36**(1), 97–120.

Lu, H., Freund, R. M. & Nesterov, Y. (2018), 'Relatively smooth convex optimization by first-order methods, and applications', *SIAM Journal on Optimization* **28**(1), 333–354.

Madan, V., Nikolov, A., Singh, M. & Tantipongpipat, U. (2020), Maximizing determinants under matroid constraints, *in* 'Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science', IEEE, pp. 565–576.

Marcus, A. W., Spielman, D. A. & Srivastava, N. (2015), 'Interlacing families II: Mixed characteristic polynomials and the Kadison–Singer problem', *Annals of Mathematics* **182**(1), 327–350.

McCormick, G. P. (1976), 'Computability of global solutions to factorable nonconvex programs: Part i – Convex underestimating problems', *Mathematical Programming* **10**(1), 147–175.

Morgan, K. L. & Rubin, D. B. (2012), 'Rerandomization to improve covariate balance in experiments', *The Annals of Statistics* **40**(2), 1263–1282.

O'Brien, T. E. & Funk, G. M. (2003), 'A gentle introduction to optimal design for regression models', *The American Statistician* **57**(4), 265–267.

Pauphilet, J. (2024), 'Robust and heterogenous odds ratio: Estimating price sensitivity for unbought items', *Manufacturing & Service Operations Management* **26**(1), 11–27.

Pokhilko, V., Zhang, Q., Kang, L. & Mays, D. P. (2019), 'D-optimal design for network A/B testing', *Journal of Statistical Theory and Practice* **13**(4), 61:1–61:23.

Pukelsheim, F. (2006), *Optimal design of experiments*, SIAM.

Rekkas, A., Paulus, J. K., Raman, G., Wong, J. B., Steyerberg, E. W., Rijnbeek, P. R., Kent, D. M. & van Klaveren, D. (2020), 'Predictive approaches to heterogeneous treatment effects: a scoping review', *BMC Medical Research Methodology* **20**(1), 264:1–264:12.

Retana, G. F., Forman, C. & Wu, D. J. (2016), 'Proactive customer education, customer retention, and demand for technology support: Evidence from a field experiment', *Manufacturing & Service Operations Management* **18**(1), 34–50.

Sagnol, G. & Harman, R. (2015), 'Computing exact D-optimal designs by mixed integer second-order cone programming', *The Annals of Statistics* **43**(5), 2198–2224.

Savelsbergh, M. W. P. (2024), Branch and price: Integer programming with column generation, *in* P. M. Pardalos & O. A. Prokopyev, eds, 'Encyclopedia of Optimization', Springer.

Singh, M. & Xie, W. (2020), 'Approximation algorithms for D-optimal design', *Mathematics of Operations Research* **45**(4), 1512–1534.

Smith, J. (2022), 'Treatment effect heterogeneity', *Evaluation Review* **46**(5), 652–677.

Somanchi, S., Abbasi, A., Kelley, K., Dobolyi, D. & Yuan, T. T. (2023), 'Examining user heterogeneity in digital experiments', *ACM Transactions on Information Systems* **41**(4), 1–34.

Stock, J. H. & Watson, M. W. (2015), *Introduction to econometrics (3rd ed.)*, Pearson London.

van der Maaten, L. & Hinton, G. (2008), 'Visualizing data using t-SNE', *Journal of Machine Learning Research* **9**(11), 2579–2605.

Vandenberghe, L., Boyd, S. & Wu, S.-P. (1998), 'Determinant maximization with linear matrix inequality constraints', *SIAM Journal on Matrix Analysis and Applications* **19**(2), 499–533.

Vivalt, E. (2015), 'Heterogeneous treatment effects in impact evaluation', *American Economic Review* **105**(5), 467–470.

Wang, J., Xie, W. & Ryzhov, I. O. (2026), 'Algorithms for budget-constrained D-optimal design', *Mathematics of Operations Research (to appear)*.

Wang, J., Xie, W., Ryzhov, I. O., Marković, N. & Ou, G. (2025), 'D-optimal orienteering for post-earthquake reconnaissance planning', *Operations Research* **73**(5), 2375–2395.

Xiong, R., Athey, S., Bayati, M. & Imbens, G. (2024), 'Optimal experimental design for staggered rollouts', *Management Science* **70**(8), 5317–5336.

Zhang, Q. & Kang, L. (2022), 'Locally optimal design for A/B tests in the presence of covariates and network dependence', *Technometrics* **64**(3), 358–369.

Zhang, Q., Khademi, A. & Song, Y. (2022), 'Min-max optimal design of two-armed trials with side information', *INFORMS Journal on Computing* **34**(1), 165–182.

8 Appendix: proofs

In the following, we give complete proofs for all results that were stated in the main text.

8.1 Proof of Lemma 3.1

Håstad (2001) used the APX-completeness of MAX-CUT to prove that, among all graphs $G = (V, E)$ of degree at most 3 with MAX-CUT value $c(G)$, it is NP-hard to distinguish a graph satisfying $c(G) \geq K$ from one satisfying $c(G) \leq (1-\varepsilon_0) K$, for any integer K.

By Thm. 11 in Edwards (1973), we have $\frac{|E|}{2} \leq c(G) \leq |E|$. Thus, it is sufficient to consider K in the range $\frac{|E|}{2} \leq K \leq |E|$. Pick a positive integer T and consider a finite multiplicative grid

$$\frac{1}{2} = t_0 < t_1 < ... < t_T = 1$$

such that $(1-\varepsilon_0) t_{j+1} < t_j$ for each $j = 0, ..., T-1$. We therefore have

$$t_j |E| \leq K \leq t_{j+1} |E|$$

for some $j \in \{0, ..., T-1\}$. If $c(G) \geq K$, then $c(G) \geq t_j |E|$. If $c(G) \leq (1-\varepsilon_0) K$, then $c(G) \leq (1-\varepsilon_0) t_{j+1} |E|$.

Let $\gamma_1 = (1-\varepsilon_0) t_{j+1}$ and $\gamma_2 = t_j$. If it were possible to distinguish $c(G) \geq \gamma_2 |E|$ from $c(G) \leq \gamma_1 |E|$ in polynomial time for all $\gamma_1 < \gamma_2$, then it would also be possible to distinguish $c(G) \geq K$ from $c(G) \leq (1-\varepsilon_0) K$ in polynomial time by checking the finite set of grid endpoints. This would contradict the APX-hardness of MAX-CUT.

8.2 Proof of Theorem 3.1

For a set $S \subseteq \{1, ..., n\}$, let $\boldsymbol{X}_S = \sum_{i \in S} \boldsymbol{x}_i \boldsymbol{x}_i^\top$. Define $\Psi(S) = \det(\boldsymbol{X}_S) \det(\boldsymbol{X}_{S^c})$. The unconstrained D-optimal partitioning problem may be equivalently written as $\max_S \Psi(S)$.

By Lemma 3.1, there exist $0 < \gamma_1 < \gamma_2 < 1$ such that it is NP-hard to distinguish whether a graph $G = (V, E)$ of degree at most 3 satisfies $c(G) \geq \gamma_2 |E|$ or $c(G) \leq \gamma_1 |E|$.

Let $G = (V, E)$ be a graph of degree at most 3 with $|V| = n$ and $|E| = m$. Let $\boldsymbol{e}_f \in \mathbb{R}^m$ denote the standard basis vector corresponding to edge $f \in E$. For each vertex $v \in V$, define its unsigned incidence vector $\boldsymbol{b}_v = \sum_{f \in \delta(v)} \boldsymbol{e}_f$, where $\delta(v)$ is the set of edges incident to v. Since G has degree at most 3, we have $\|\boldsymbol{b}_v\|_2^2 \leq 3$. We also have $\sum_v \|\boldsymbol{b}_v\|_2^2 = 2m$.

Suppose that $L \geq 1$ is a fixed integer whose value does not depend on m, and define

$$q = \frac{L}{\lceil m^{\frac{1}{4}} \rceil}, \qquad \varepsilon = q^2.$$

Construct the matrix $\boldsymbol{X} \in \mathbb{R}^{(2m+n)\times n}$ in the following manner: for each $f \in E$, add two copies of row $\boldsymbol{e}_f^\top$, and for each vertex $v \in V$, add one row $q\boldsymbol{b}_v^\top$. Let $S \subseteq \{1, ..., 2m+n\}$ be an arbitrary set of indices of rows of $\boldsymbol{X}$. Note that these indices can either come from the first $2m$ rows, which correspond to edges in the graph (we call these "edge rows"), or from the last n rows, which correspond to vertices ("vertex rows"). Let $U \subseteq V$ be the set of vertices whose row indices were included in S.

We now state the following three-part technical lemma, whose proof is deferred to a separate section of the Appendix for readability.

Lemma 8.1. *The following statements hold:*

i) Define $\boldsymbol{C}_U = \sum_{v \in U} \boldsymbol{b}_v \boldsymbol{b}_v^\top$. *Then,* $0 \preceq \boldsymbol{C}_U \preceq 6 \cdot \boldsymbol{I}_m$ *and* $0 \leq \preceq \boldsymbol{C}_{U^c} \preceq 6 \cdot \boldsymbol{I}_m$.

ii) There exists a graph $G' = (V', E')$ *with* $|E'| > m$ *such that solving MAX-CUT for* G' *is equivalent to solving MAX-CUT for* G.

iii) Suppose that $0 \preceq \boldsymbol{C} \preceq 6 \cdot \boldsymbol{I}_m$ *and* $0 < \varepsilon < \frac{1}{12}$. *Then,*

$$\varepsilon \operatorname{tr}(\boldsymbol{C}) - \frac{\varepsilon^2}{2} \operatorname{tr}\left(\boldsymbol{C}^2\right) \leq \log\det(\boldsymbol{I}_m + \varepsilon \boldsymbol{C}) \leq \varepsilon \operatorname{tr}(\boldsymbol{C}) - \frac{\varepsilon^2}{2} \operatorname{tr}\left(\boldsymbol{C}^2\right) + 72\varepsilon^3 m.$$

Now, for each $f \in E$, let $a_f \in \{0, 1, 2\}$ be the number of copies of $\boldsymbol{e}_f$ whose indices are elements of S (there are two copies of $\boldsymbol{e}_f$ among the first $2m$ rows of $\boldsymbol{X}$, and S can count neither of them, one of them, or both). We now derive bounds on $\Psi(S)$ based on the values of a_f.

Case 1: $a_f \neq 1$ for some $f \in E$. Then, either S or S^c contains no indices of rows of $\boldsymbol{X}$ corresponding to copies of $\boldsymbol{e}_f$. We may assume that this is true for S, because S^c is handled symmetrically. Then, in that case, the only remaining rows of $\boldsymbol{X}$ that have nonzero fth coordinate are the two vertex rows corresponding to the vertices connected by f. Therefore,

$$\boldsymbol{e}_f^\top \boldsymbol{X}_S \boldsymbol{e}_f \leq 2q^2 = 2\varepsilon,$$

whence $\lambda_{\min}(\boldsymbol{X}_S) \leq 2\varepsilon$. Moreover,

$$\operatorname{tr}(\boldsymbol{X}_S) + \operatorname{tr}(\boldsymbol{X}_{S^c}) = 2m + \sum_{v \in V} q^2 \cdot \|\boldsymbol{v}\|_2^2 = 2m(1+\varepsilon).$$

Applying the inequality of arithmetic and geometric means to the $2m-1$ eigenvalues of $\boldsymbol{X}_S$ and $\boldsymbol{X}_{S^c}$, leaving out only the smallest eigenvalue $\lambda_{\min}(\boldsymbol{X}_S)$ of $\boldsymbol{X}_S$, yields

$$\Psi(S) \leq 2\varepsilon \left(\frac{2m(1+\varepsilon)}{2m-1} \right)^{2m-1}.$$

Therefore,

$$\begin{aligned} \log(\Psi(S)) &\leq \log(2\varepsilon) + (2m-1)\log\left(\frac{1+\varepsilon}{1-\frac{1}{2m}} \right) \\ &= \log(2\varepsilon) + (2m-1)\log(1+\varepsilon) - (2m-1)\log\left(1 - \frac{1}{2m}\right). \end{aligned} \tag{30}$$

Using the inequalities $\log(1+\varepsilon) \leq \varepsilon$ and $-\log\left(1-\frac{1}{2m}\right) \leq \frac{1}{m}$, (30) continues as

$$\begin{aligned} \log(\Psi(S)) &\leq \log(2\varepsilon) + (2m-1)\varepsilon + \frac{2m-1}{m} \\ &\leq \log(2\varepsilon) + 2\varepsilon m + 2. \end{aligned} \tag{31}$$

Case 2: $a_f = 1$ for all $f \in E$. Then, letting $\boldsymbol{C}_U = \sum_{v \in U} \boldsymbol{b}_v \boldsymbol{b}_v^\top$, we have

$$\boldsymbol{X}_S = \boldsymbol{I}_m + \varepsilon \boldsymbol{C}_U, \qquad \boldsymbol{X}_{S^c} = \boldsymbol{I}_m + \varepsilon \boldsymbol{C}_{U^c}$$

whence

$$\Psi(S) = \det(\boldsymbol{I}_m + \varepsilon \boldsymbol{C}_U) \det(\boldsymbol{I}_m + \varepsilon \boldsymbol{C}_{U^c}).$$

Applying Lemma 8.1(i) and 8.1(iii), we have

$$\varepsilon \operatorname{tr}(\boldsymbol{C}) - \frac{\varepsilon^2}{2} \operatorname{tr}\left(\boldsymbol{C}^2\right) \leq \log\det(\boldsymbol{I}_m + \varepsilon \boldsymbol{C}) \leq \varepsilon \operatorname{tr}(\boldsymbol{C}) - \frac{\varepsilon^2}{2} \operatorname{tr}\left(\boldsymbol{C}^2\right) + 72\varepsilon^3 m \tag{32}$$

for $\boldsymbol{C} \in \{\boldsymbol{C}_U, \boldsymbol{C}_{U^c}\}$. Note that, by Lemma 8.1(ii), we may enlarge m as needed without changing the problem, so we may assume that ε meets the conditions of Lemma 8.1(iii).

We now compute the bounds in (32) more explicitly. First,

$$\operatorname{tr}(\boldsymbol{C}_U) + \operatorname{tr}(\boldsymbol{C}_{U^c}) = \sum_{v \in V} \|\boldsymbol{b}_v\|_2^2 = 2m. \tag{33}$$

Also,

$$\mathrm{tr}\left(\boldsymbol{C}_U^2\right) = \sum_{u,v\in U}\left(\boldsymbol{b}_u^\top \boldsymbol{b}_v^\top\right)^2 = \sum_{v\in U}\left(\|\boldsymbol{b}_v\|_2^2\right)^2 + 2\cdot|E_U|,$$

where $E_U \subseteq E$ is the set of edges that have both endpoints in U. Applying the same logic to U^c, we obtain

$$\begin{aligned}\mathrm{tr}\left(\boldsymbol{C}_U^2\right) + \mathrm{tr}\left(\boldsymbol{C}_{U^c}^2\right) &= \sum_{v\in V}\left(\|\boldsymbol{b}_v\|_2^2\right)^2 + 2\left(|E_U| + |E_{U^c}|\right)\\ &= \sum_{v\in V}\left(\|\boldsymbol{b}_v\|_2^2\right)^2 + 2\left(m - |\delta(U)|\right),\end{aligned} \tag{34}$$

where $\delta(U)$ is the set of edges with one endpoint in U and the other in U^c. Plugging (33) and (34) into (32), we obtain

$$2\varepsilon m - \frac{\varepsilon^2}{2}\left(\sum_{v\in V}\left(\|\boldsymbol{b}_v\|_2^2\right)^2 + 2m\right) + \varepsilon^2|\delta(U)| \le \log(\Psi(S)) \le 2\varepsilon m - \frac{\varepsilon^2}{2}\left(\sum_{v\in V}\left(\|\boldsymbol{b}_v\|_2^2\right)^2 + 2m\right) + \varepsilon^2|\delta(U)| + 72\varepsilon^3 m. \tag{35}$$

Thus, Cases 1 and 2 have yielded the respective bounds (31) and (35). We now seek to combine them. Suppose that G belongs to the first of the two possible situations in the gap-distinguishing problem of Lemma 3.1, that is, there exists $U \subseteq V$ such that $|\delta(U)| \ge \gamma_2 m$. We may assume $|U| \le \frac{n}{2}$ without loss of generality (if this does not hold, we consider U^c instead). We select S by taking all the vertex rows of $\boldsymbol{X}$ corresponding to U, and exactly one of the two copies of the edge rows $\boldsymbol{e}_f$ for every edge f. Then, by (35), we have

$$\log\left(\max_S \Psi(S)\right) \ge 2\varepsilon m - \frac{\varepsilon^2}{2}\left(\sum_{v\in V}\left(\|\boldsymbol{b}_v\|_2^2\right)^2 + 2m\right) + \gamma_2\varepsilon^2 m. \tag{36}$$

However, if G belongs to the second situation of Lemma 3.1, that is, every $U \subseteq V$ satisfies $|\delta(U)| \le \gamma_1 m$, then the same construction of S yields

$$\log(\Psi(S)) \le 2\varepsilon m - \frac{\varepsilon^2}{2}\left(\sum_{v\in V}\left(\|\boldsymbol{b}_v\|_2^2\right)^2 + 2m\right) + \gamma_1\varepsilon^2 m + 72\varepsilon^3 m. \tag{37}$$

On the other hand, any S that does not include exactly one copy of every edge row satisfies (31). By Lemma 8.1(ii), we can enlarge m as needed without changing the nature of the problem. Since $\varepsilon = \frac{L^2}{\lceil m^{\frac{1}{4}}\rceil^2}$ taking $m \to \infty$ yields $\varepsilon \to 0$, $\varepsilon^2 m \to L^4$, and $\varepsilon^3 m \to 0$. In other words, we can take m to be large enough to make

$$\log(2\varepsilon) + 2\varepsilon m + 2 \le 2\varepsilon m - \frac{\varepsilon^2}{2}\left(\sum_{v\in V}\left(\|\boldsymbol{b}_v\|_2^2\right)^2 + 2m\right) + \gamma_1\varepsilon^2 m + 72\varepsilon^3 m.$$

Therefore, when G belongs to the second situation of Lemma 3.1, we may combine the bounds (31) and (35) of Cases 1 and 2 and obtain

$$\log\left(\max_S \Psi(S)\right) \le 2\varepsilon m - \frac{\varepsilon^2}{2}\left(\sum_{v\in V}\left(\|\boldsymbol{b}_v\|_2^2\right)^2 + 2m\right) + \gamma_1\varepsilon^2 m + 72\varepsilon^3 m. \tag{38}$$

Thus, (36) gives a lower bound on $\log(\max_S \Psi(S))$ in the first situation of Lemma 3.1, and (38) gives an upper bound on $\log(\max_S \Psi(S))$ in the second situation of Lemma 3.1. Let $\bar{\gamma} = \gamma_2 - \gamma_1$. We exponentiate both sides of (36) and (38) and take the ratio between the lower bound and the upper bound. This number is at least

$$\exp\left(\bar{\gamma}\varepsilon^2 m + 72\varepsilon^3 m\right) = \exp\left(\bar{\gamma}\cdot L^4 + \bar{\gamma}\cdot L^4\left(\frac{m}{\lceil m^{\frac{1}{4}}\rceil^4} - 1\right) + 72L^6\frac{m}{\lceil m^{\frac{1}{4}}\rceil^6}\right). \tag{39}$$

If we take

$$m \ge \left\lceil\left(\frac{1}{2^{\frac{1}{4}} - 1}\right)^4\right\rceil = 781, \tag{40}$$

we derive

$$\begin{aligned} & \bar{\gamma}\cdot L^4 + \bar{\gamma}\cdot L^4\left(\frac{m}{\lceil m^{\frac{1}{4}}\rceil^4} - 1\right) + 72L^6\frac{m}{\lceil m^{\frac{1}{4}}\rceil^6} \\ \ge\ & \bar{\gamma}\cdot L^4\left(\frac{m}{\lceil m^{\frac{1}{4}}\rceil^4}\right) \\ \ge\ & \bar{\gamma}\cdot L^4\left(\frac{m}{\left(m^{\frac{1}{4}}+1\right)^4}\right) \\ =\ & \bar{\gamma}\cdot L^4\left(\frac{1}{m^{-\frac{1}{4}}+1}\right)^4 \\ \ge\ & \frac{\bar{\gamma}\cdot L^4}{2}, \end{aligned}$$

where the last line is obtained from (40). Consequently, (39) becomes

$$\exp\left(\bar{\gamma}\varepsilon^2 m + 72\varepsilon^3 m\right) \ge \exp\left(\frac{\bar{\gamma}\cdot L^4}{2}\right). \tag{41}$$

Pick any $\rho \in (0,1)$ and choose $L = \lceil\left(-\frac{2\log\rho}{\bar{\gamma}} + \frac{2}{\bar{\gamma}}\right)^{\frac{1}{4}}\rceil$. Then, plugging this value of L into (41) yields

$$\rho\exp\left(\frac{\bar{\gamma}\cdot L^4}{2}\right) \ge e > 1. \tag{42}$$

The final step of the proof is to show that, if there exists a polynomial time ρ-approximation algorithm for the D-optimal partitioning problem, then running it on the constructed instance would distinguish between the two situations of Lemma 3.1, contradicting the result of that lemma. Suppose that such an algorithm existed, i.e., for any instance of D-optimal partitioning, we could obtain S_0 satisfying $\Psi(S_0) \geq \rho \cdot \max_S \Psi(S)$. Then, in the first situation of Lemma 3.1, we have

$$
\begin{aligned}
\Psi(S_0) &\geq \rho \cdot \max_S \Psi(S) \\
&\geq \rho \cdot \exp\left(2\varepsilon m - \frac{\varepsilon^2}{2}\left(\sum_{v \in V}\left(\|\boldsymbol{b}_v\|_2^2\right)^2 + 2m\right) + \gamma_2 \varepsilon^2 m\right) && (43) \\
&> \exp\left(2\varepsilon m - \frac{\varepsilon^2}{2}\left(\sum_{v \in V}\left(\|\boldsymbol{b}_v\|_2^2\right)^2 + 2m\right) + \gamma_1 \varepsilon^2 m + 72\varepsilon^3 m\right), && (44)
\end{aligned}
$$

where (43) follows by (36) and (44) is due to (41)-(42). On the other hand, in the second situation of Lemma 3.1, we have

$$
\begin{aligned}
\Psi(S_0) &\leq \max_S \Psi(S) \\
&\leq \exp\left(2\varepsilon m - \frac{\varepsilon^2}{2}\left(\sum_{v \in V}\left(\|\boldsymbol{b}_v\|_2^2\right)^2 + 2m\right) + \gamma_1 \varepsilon^2 m + 72\varepsilon^3 m\right) && (45)
\end{aligned}
$$

where (45) follows by (38). This allows us to distinguish between the two situations: we evaluate $\Psi(S_0)$ and check whether (44) or (45) holds. Since this approach applies to any $\rho \in (0, 1)$, we conclude that there is no polynomial-time ρ-approximation algorithm for the D-optimal partitioning problem unless $P = NP$.

8.3 Proof of Proposition 4.1

Let $\boldsymbol{W}, \boldsymbol{Y} \succeq 0$ and $\boldsymbol{Z}$ satisfy (13)-(14) as well as (11). Define $z_i = \frac{1+Z_i}{2}$. Because $W \succeq 0$ satisfies (13) and $Y_{ii} = 1$ for all i, every 2×2 principal minor of W has positive determinant, meaning that

$$
1 - Z_i^2 = \det\begin{pmatrix} 1 & Z_i \\ Z_i & 1 \end{pmatrix} \geq 0.
$$

Therefore, $|Z_i| \leq 1$, so $0 \leq z_i \leq 1$. Furthermore,

$$
\sum_{i=1}^n z_i = \frac{1}{2}\left(n + \sum_{i=1}^n Z_i\right) \leq \frac{1}{2}(n + 2k - n) = k,
$$

so $\boldsymbol{z}$ is feasible for (7) subject to (6).

For notational convenience, let $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{Z})$ and $\boldsymbol{M} = \boldsymbol{X}^\top \boldsymbol{D} \boldsymbol{X}$. From (13), we have $\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{Z}^\top \succeq 0$. Because $\boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \succeq 0$ and the Hadamard product preserves positive semidefiniteness,

$$\boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \circ \boldsymbol{Y} \succeq \boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \circ \boldsymbol{Z}\boldsymbol{Z}^\top = \boldsymbol{D}\boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{D}.$$

Using $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top$ and $\boldsymbol{I} \circ \boldsymbol{Y} = \boldsymbol{I}$, we obtain

$$\boldsymbol{P} \circ \boldsymbol{Y} = \boldsymbol{I} - \boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \circ \boldsymbol{Y} \preceq \boldsymbol{I} - \boldsymbol{D}\boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{D},$$

whence

$$\begin{aligned} \boldsymbol{X}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{X} &\preceq \boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{X}^\top \boldsymbol{D}\boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{D}\boldsymbol{X} \\ &= \boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{M}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{M}. \end{aligned}$$

Therefore,

$$\log\det\left(\boldsymbol{X}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{X}\right) \leq \log\det\left(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{M}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{M}\right), \tag{46}$$

with the convention that the log-determinant is equal to $-\infty$ on singular matrices.

Next, since $\boldsymbol{z} = \frac{1+\boldsymbol{Z}}{2}$, we also have $\operatorname{diag}(\boldsymbol{z}) = \frac{\boldsymbol{I}+\boldsymbol{D}}{2}$, whence

$$\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{z}) \boldsymbol{X} = \frac{\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{M}}{2}, \qquad \boldsymbol{X}^\top (\boldsymbol{I} - \operatorname{diag}(\boldsymbol{z})) \boldsymbol{X} = \frac{\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{M}}{2}.$$

Therefore, the objective value of (7) evaluated at $\boldsymbol{z}$ is given by

$$\begin{aligned} & \log\det\left(\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{z}) \boldsymbol{X}\right) + \log\det\left(\boldsymbol{X}^\top (\boldsymbol{I} - \operatorname{diag}(\boldsymbol{z})) \boldsymbol{X}\right) \\ = \;& \log\det\left(\frac{\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{M}}{2}\right) + \log\det\left(\frac{\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{M}}{2}\right) \\ = \;& -2d\log 2 + \log\det\left(\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{M}\right) + \log\det\left(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{M}\right). \\ = \;& \log\det\left(\boldsymbol{X}^\top \boldsymbol{X}\right) - 2d\log 2 + \log\det\left(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{M}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{M}\right). \end{aligned} \tag{47}$$

Combining (46) with (47) yields

$$\begin{aligned} & \log\det\left(\boldsymbol{X}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{X}\right) \\ \leq \;& \log\det\left(\boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{z}) \boldsymbol{X}\right) + \log\det\left(\boldsymbol{X}^\top (\boldsymbol{I} - \operatorname{diag}(\boldsymbol{z})) \boldsymbol{X}\right) - \log\det\left(\boldsymbol{X}^\top \boldsymbol{X}\right) + 2d\log 2. \end{aligned}$$

Since $\boldsymbol{z}$ is feasible for (7), taking the maximum over all feasible $(\boldsymbol{Y}, \boldsymbol{Z})$ proves the desired result.

8.4 Proof of Proposition 4.2

First, let $\boldsymbol{Z} \in \{-1,1\}^n$ be part of a feasible solution to Problem (12)-(15). Let $\boldsymbol{Y} = \boldsymbol{Z}\boldsymbol{Z}^\top$. Then, $\boldsymbol{Y} \succeq 0$, $\text{rank}(\boldsymbol{Y}) = 1$, and $Y_{ii} = Z_i^2 = 1$ for all i. If we are not budget-constrained, we are done; otherwise, we have $2k - n < 0$ and $\mathbf{1}^\top \boldsymbol{Z} \leq 2k - n$ by (11). We then observe that $\mathbf{1}^\top \boldsymbol{Y} \mathbf{1} = \left(\mathbf{1}^\top \boldsymbol{Z}\right)^2$, whence $\mathbf{1}^\top \boldsymbol{Y} \mathbf{1} \geq (2k-n)^2$, so (18) is satisfied. Thus, $\boldsymbol{Y}$ is feasible for the new problem.

Conversely, let $\boldsymbol{Y} \succeq 0$ satisfy $\text{rank}(\boldsymbol{Y}) = 1$ and $Y_{ii} = 1$ for all i. Then, $\boldsymbol{Y} = \boldsymbol{u}\boldsymbol{u}^\top$ for some $\boldsymbol{u} \in \{-1, 1\}$. If we are not budget-constrained, $\boldsymbol{Y}$ and $\boldsymbol{Z} = \boldsymbol{u}$ are feasible for Problem (12)-(15). If we are budget-constrained, then we further suppose that $\mathbf{1}^\top \boldsymbol{Y} \mathbf{1} \geq (2k-n)^2$. Choose $\boldsymbol{Z} \in \{\boldsymbol{u}, -\boldsymbol{u}\}$ so that $\mathbf{1}^\top \boldsymbol{Z} = -\sqrt{1^\top \boldsymbol{Y} \mathbf{1}}$, whence (11) holds and $\boldsymbol{Y} = \boldsymbol{Z}\boldsymbol{Z}^\top$ is feasible for the original problem. Thus, the two formulations have equivalent feasible regions and therefore the same optimal values.

8.5 Proof of Proposition 4.3

Let $\boldsymbol{W}, \boldsymbol{Y} \succeq 0$ and $\boldsymbol{Z}$ be a feasible solution to (13)-(14). If we are not budget-constrained, we are done since $\boldsymbol{Y}$ is feasible for the new problem. If we are budget-constrained (so, $2k - n < 0$), we first observe that (13) is equivalent by the Schur complement to $\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{Z}^\top \succeq 0$. This implies that

$$\mathbf{1}^\top \boldsymbol{Y} \mathbf{1} \geq \left(\mathbf{1}^\top \boldsymbol{Z}\right)^2 \geq (2k-n)^2,$$

so (18) is satisfied.

Now, consider $\boldsymbol{Y} \succeq 0$ satisfying $Y_{ii} = 1$ for all i. If we are not budget-constrained, we simply let $\boldsymbol{Z} = 0$ and (13) holds. If we are budget-constrained, then we further suppose that $\mathbf{1}^\top \boldsymbol{Y} \mathbf{1} \geq (2k-n)^2$. Define $\boldsymbol{Z} = \frac{2k-n}{\mathbf{1}^\top \boldsymbol{Y} \mathbf{1}} \boldsymbol{Y} \mathbf{1}$. Then, $\mathbf{1}^\top \boldsymbol{Z} = 2k - n$, so (11) is satisfied. It remains only to show that $\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{Z}^\top \succeq 0$. Observe that

$$\boldsymbol{Z}\boldsymbol{Z}^\top = \frac{(2k-n)^2}{\left(\mathbf{1}^\top \boldsymbol{Y} \mathbf{1}\right)^2} \boldsymbol{Y} \mathbf{1} \mathbf{1}^\top \boldsymbol{Y}.$$

By (18), it is enough to show that

$$\frac{1}{\mathbf{1}^\top \boldsymbol{Y} \mathbf{1}} \boldsymbol{Y} \mathbf{1} \mathbf{1}^\top \boldsymbol{Y} \preceq \boldsymbol{Y}.$$

To see this, take any $\boldsymbol{u} \in \mathbb{R}^n$ and observe that

$$\boldsymbol{u}^\top \left(\boldsymbol{Y} - \frac{1}{\mathbf{1}^\top \boldsymbol{Y} \mathbf{1}} \boldsymbol{Y} \mathbf{1} \mathbf{1}^\top \boldsymbol{Y}\right) \boldsymbol{u} = \boldsymbol{u}^\top \boldsymbol{Y} \boldsymbol{u} - \frac{\left(\boldsymbol{u}^\top \boldsymbol{Y} \mathbf{1}\right)^2}{\mathbf{1}^\top \boldsymbol{Y} \mathbf{1}} \geq 0$$

applying the Cauchy-Schwarz inequality for the positive semidefinite matrix $\boldsymbol{Y}$. Hence, $\boldsymbol{Y}, \boldsymbol{Z}$ are feasible for the original (relaxed) problem. Thus, the two formulations have equivalent feasible regions and therefore the same optimal values.

8.6 Proof of Proposition 4.4

Because $\boldsymbol{X} = \boldsymbol{U}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{\frac{1}{2}}$ and $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\top$, we may write

$$\boldsymbol{X}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{X} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{\frac{1}{2}} \boldsymbol{U}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{U} \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{\frac{1}{2}},$$

whence

$$\log\det\left(\boldsymbol{X}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{X}\right) = \log\det\left(\boldsymbol{X}^\top \boldsymbol{X}\right) + \log\det\left(\boldsymbol{U}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{U}\right).$$

Therefore, we may drop the constant term and solve $\max_{\boldsymbol{Y}\in\mathcal{C}} \log\det\left(\boldsymbol{U}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{U}\right)$.

By assumption, the one-dimensional subspaces spanned by the rows of $\boldsymbol{U}$ are mutually orthogonal. Since $\boldsymbol{U}$ has rank d, there exists an orthonormal basis $\boldsymbol{q}_1, ..., \boldsymbol{q}_d$ of $\mathbb{R}^d$, a partition $G_1 \cup ... \cup G_d = \{1, ..., n\}$, and scalars $\alpha_i \in \mathbb{R}$ such that $\boldsymbol{u}_i = \alpha_i \boldsymbol{q}_\ell$ for every $i \in G_\ell$. We may then write

$$\boldsymbol{I}_d = \sum_{i=1}^n \boldsymbol{u}_i \boldsymbol{u}_i^\top = \sum_{\ell=1}^d \left(\sum_{i\in G_\ell} \alpha_i^2\right) \boldsymbol{q}_\ell \boldsymbol{q}_\ell^\top, \tag{48}$$

and therefore $\sum_{i\in G_\ell} \alpha_i^2 = 1$ for each $\ell = 1, ..., d$.

Now fix $\boldsymbol{Z} \in \{\pm 1\}^n$. Since $\boldsymbol{P} \circ \left(\boldsymbol{Z}\boldsymbol{Z}^\top\right) = \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{P} \operatorname{diag}(\boldsymbol{Z})$, we obtain

$$\boldsymbol{U}^\top \left(\boldsymbol{P} \circ \boldsymbol{Z}\boldsymbol{Z}^\top\right) \boldsymbol{U} = \boldsymbol{U}^\top \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{P} \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{U} = \boldsymbol{I}_d - \left(\boldsymbol{U}^\top \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{U}\right)^2.$$

Define $s_\ell(\boldsymbol{Z}) = \sum_{i\in G_\ell} \alpha_i^2 Z_i$. Then, by similar arguments as in (48), we have

$$\boldsymbol{U}^\top \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{U} = \sum_{i=1}^n Z_i \boldsymbol{u}_i \boldsymbol{u}_i^\top = \sum_{\ell=1}^d s_\ell(\boldsymbol{Z})\, \boldsymbol{q}_\ell \boldsymbol{q}_\ell^\top,$$

and therefore

$$\boldsymbol{U}^\top \left(\boldsymbol{P} \circ \boldsymbol{Z}\boldsymbol{Z}^\top\right) \boldsymbol{U} = \sum_{\ell=1}^d \left(1 - s_\ell(\boldsymbol{Z})^2\right) \boldsymbol{q}_\ell \boldsymbol{q}_\ell^\top. \tag{49}$$

Let $\boldsymbol{Y} \in \mathcal{C}$. Then, by the properties of the convex hull, we may write $\boldsymbol{Y} = \mathbb{E}\left(\tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^\top\right)$ for some discrete random vector $\tilde{\boldsymbol{Z}} \in \{\pm 1\}^n$. Using linearity of the expectation together with (49), we obtain

$$\boldsymbol{U}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{U} = \mathbb{E}\left(\boldsymbol{U}^\top \left(\boldsymbol{P} \circ \tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^\top\right) \boldsymbol{U}\right) = \sum_{\ell=1}^d \left(1 - \mathbb{E}\left(s_\ell\left(\tilde{\boldsymbol{Z}}\right)^2\right)\right) \boldsymbol{q}_\ell \boldsymbol{q}_\ell^\top.$$

Therefore,

$$\log\det\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)=\sum_{\ell=1}^{d}\log\left(1-\mathbb{E}\left(s_\ell\left(\tilde{\boldsymbol{Z}}\right)^2\right)\right). \tag{50}$$

For each ℓ, define

$$m_\ell=\min_{\varepsilon\in\{\pm1\}^{|G_\ell|}}\left(\sum_{i\in G_\ell}\alpha_\ell^2\varepsilon_\ell\right)^2.$$

Since $s_\ell\left(\tilde{\boldsymbol{Z}}\right)^2\geq m_\ell$, we have $\mathbb{E}\left(s_\ell\left(\tilde{\boldsymbol{Z}}\right)^2\right)\geq m_\ell$ for each ℓ. Consequently, (50) implies

$$\log\det\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)\leq\sum_{\ell=1}^{d}\log(1-m_\ell). \tag{51}$$

Because the sets $G_1,...,G_d$ are disjoint, we may choose $\boldsymbol{Z}^*\in\{\pm1\}^n$ so that, for each ℓ, the components of $\boldsymbol{Z}^*$ corresponding to elements of G_ℓ are set according to the minimum values given by m_ℓ. Then, $s_\ell(\boldsymbol{Z}^*)^2=m_\ell$, and by (49), we have

$$\boldsymbol{U}^\top\left(\boldsymbol{P}\circ\boldsymbol{Z}^*(\boldsymbol{Z}^*)^\top\right)\boldsymbol{U}=\sum_{\ell=1}^{d}(1-m_\ell)\,\boldsymbol{q}_\ell\boldsymbol{q}_\ell^\top.$$

Therefore, (51) holds with equality, and therefore

$$\max_{\boldsymbol{Y}\in\mathcal{C}}\log\det\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)=\boldsymbol{U}^\top\left(\boldsymbol{P}\circ\boldsymbol{Z}^*(\boldsymbol{Z}^*)^\top\right)\boldsymbol{U},$$

which completes the proof.

8.7 Proof of Proposition 4.5

First, we note that

$$\Delta^*=\exp\left(\Phi^{rel}_{\bar{\boldsymbol{X}}}-\log\det\left(\bar{\boldsymbol{X}}^\top\bar{\boldsymbol{X}}\right)\right)=\exp\left(\Phi_{\bar{\boldsymbol{X}}}-\log\det\left(\bar{\boldsymbol{X}}^\top\bar{\boldsymbol{X}}\right)\right)$$

due to Proposition 4.4.

By repeating arguments in the proof of 4.4, we may write

$$\begin{aligned}\Phi_{\boldsymbol{X}}-\log\det\left(\boldsymbol{X}^\top\boldsymbol{X}\right)&=\max_{\boldsymbol{Z}\in\{\pm1\}^n}\log\det\left(\boldsymbol{U}^\top\left(\boldsymbol{P}\circ\boldsymbol{Z}\boldsymbol{Z}^\top\right)\boldsymbol{U}\right),\\ \Phi^{rel}_{\boldsymbol{X}}-\log\det\left(\boldsymbol{X}^\top\boldsymbol{X}\right)&=\max_{\boldsymbol{Y}\in\mathcal{C}}\log\det\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right).\end{aligned}$$

Analogous identities hold for $\bar{\boldsymbol{X}}$.

Let $\boldsymbol{Z} \in \{\pm 1\}^n$. Since $\boldsymbol{P} \circ \boldsymbol{Z}\boldsymbol{Z}^\top = \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{P} \operatorname{diag}(\boldsymbol{Z})$, we obtain

$$\boldsymbol{U}^\top \left(\boldsymbol{P} \circ \boldsymbol{Z}\boldsymbol{Z}^\top\right) \boldsymbol{U} = \boldsymbol{U}^\top \operatorname{diag}(\boldsymbol{Z}) \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\top\right) \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{U} = \boldsymbol{I} - \left(\boldsymbol{U}^\top \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{U}\right)^2,$$

and analogously for $\bar{\boldsymbol{U}}, \bar{\boldsymbol{P}}$. Since $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}$ and $\|\operatorname{diag}(\boldsymbol{Z})\|_2 = 1$, we have

$$\|\boldsymbol{U}^\top \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{U}\|_2 \leq 1, \qquad \|\bar{\boldsymbol{U}} \operatorname{diag}(\boldsymbol{Z})\, \bar{\boldsymbol{U}}\|_2 \leq 1.$$

Moreover,

$$\begin{aligned}
\|\boldsymbol{U}^\top \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{U} - \bar{\boldsymbol{U}} \operatorname{diag}(\boldsymbol{Z})\, \bar{\boldsymbol{U}}\|_2 &= \|\sum_{i=1}^n Z_i \left(\boldsymbol{u}_i \boldsymbol{u}_i^\top - \bar{\boldsymbol{u}}_i \bar{\boldsymbol{u}}_i^\top\right)\|_2 \\
&\leq \sum_{i=1}^n \|\boldsymbol{u}_i \boldsymbol{u}_i^\top - \bar{\boldsymbol{u}}_i \bar{\boldsymbol{u}}_i^\top\|_2 \\
&= \delta\left(\boldsymbol{X}, \bar{\boldsymbol{X}}\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
& \|\boldsymbol{U}^\top \left(\boldsymbol{P} \circ \boldsymbol{Z}\boldsymbol{Z}^\top\right) \boldsymbol{U} - \bar{\boldsymbol{U}}^\top \left(\bar{\boldsymbol{P}} \circ \boldsymbol{Z}\boldsymbol{Z}^\top\right) \bar{\boldsymbol{U}}\|_2 \qquad (52) \\
= \; & \|\left(\boldsymbol{U}^\top \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{U}\right)^2 - \left(\bar{\boldsymbol{U}} \operatorname{diag}(\boldsymbol{Z})\, \bar{\boldsymbol{U}}\right)^2\|_2 \\
\leq \; & \left(\|\boldsymbol{U}^\top \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{U}\|_2 + \|\bar{\boldsymbol{U}} \operatorname{diag}(\boldsymbol{Z})\, \bar{\boldsymbol{U}}\|_2\right) \cdot \|\boldsymbol{U}^\top \operatorname{diag}(\boldsymbol{Z})\, \boldsymbol{U} - \bar{\boldsymbol{U}} \operatorname{diag}(\boldsymbol{Z})\, \bar{\boldsymbol{U}}\|_2 \\
\leq \; & 2\delta\left(\boldsymbol{X}, \bar{\boldsymbol{X}}\right). \qquad (53)
\end{aligned}$$

Now, let $\boldsymbol{Y} \in \mathcal{C}$. Then, we can write $\boldsymbol{Y} = \mathbb{E}\left(\tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^\top\right)$ for a random vector $\tilde{\boldsymbol{Z}}$ with finite support contained in $\{-1, 1\}^n$. By linearity,

$$\boldsymbol{U}^\top (\boldsymbol{P} \circ \boldsymbol{Y})\, \boldsymbol{U} = \mathbb{E}\left(\boldsymbol{U}^\top \left(\boldsymbol{P} \circ \tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^\top\right) \boldsymbol{U}\right),$$

and similarly for $\bar{\boldsymbol{U}}, \bar{\boldsymbol{P}}$. Applying Jensen's inequality with (53) yields the same bound

$$\|\boldsymbol{U}^\top (\boldsymbol{P} \circ \boldsymbol{Y})\, \boldsymbol{U} - \bar{\boldsymbol{U}}^\top \left(\bar{\boldsymbol{P}} \circ \boldsymbol{Y}\right) \bar{\boldsymbol{U}}\|_2 \leq 2\delta\left(\boldsymbol{X}, \bar{\boldsymbol{X}}\right)$$

for $\boldsymbol{Y}$.

Next, for every $\boldsymbol{Y} \in \mathcal{C}$, $0 \preceq \boldsymbol{U}^\top (\boldsymbol{P} \circ \boldsymbol{Y})\, \boldsymbol{U} \preceq \boldsymbol{I}_d$ because $\boldsymbol{U}^\top (\boldsymbol{P} \circ \boldsymbol{Y})\, \boldsymbol{U}$ is a convex combination of matrices of the form $0 \preceq \boldsymbol{U}^\top \left(\boldsymbol{P} \circ \boldsymbol{Z}\boldsymbol{Z}^\top\right) \boldsymbol{U} \preceq \boldsymbol{I}_d$. Therefore, if we can show that the determinant is Lipschitz on the set $\{\boldsymbol{A} \mid 0 \preceq \boldsymbol{A} \preceq \boldsymbol{I}_d\}$ with Lipschitz constant d, we will have

$$\left|\det\left(\boldsymbol{U}^\top (\boldsymbol{P} \circ \boldsymbol{Y})\, \boldsymbol{U}\right) - \det\left(\bar{\boldsymbol{U}}^\top \left(\bar{\boldsymbol{P}} \circ \boldsymbol{Y}\right) \bar{\boldsymbol{U}}\right)\right| \leq 2d \cdot \delta\left(\boldsymbol{X}, \bar{\boldsymbol{X}}\right)$$

for all $\boldsymbol{Y} \in \mathcal{C}$. Taking maxima over $\boldsymbol{Y} \in \mathcal{C}$ or over $Y = \boldsymbol{Z}\boldsymbol{Z}^\top$ for $\boldsymbol{Z} \in \{\pm 1\}^n$ then completes the proof. Thus, all that remains is the following technical lemma, whose proof is given in a separate section of the Appendix.

Lemma 8.2. *The determinant function is Lipschitz (with respect to $\|\cdot\|_2$) on $\left\{\boldsymbol{A} \in \mathbb{S}^d \mid 0 \preceq \boldsymbol{A} \preceq \boldsymbol{I}_d\right\}$ with Lipschitz constant d.*

8.8 Proof of Proposition 5.1

Recalling from (26) that $g_{\ell+1} = \langle \boldsymbol{G}_\ell, \boldsymbol{Z}_{\ell+1}\boldsymbol{Z}_{\ell+1}^\top - \boldsymbol{Y}_\ell \rangle$, it is sufficient to show that $\max_{\boldsymbol{Z}\in\{\pm1\}^n} \boldsymbol{Z}^\top \boldsymbol{G}_\ell \boldsymbol{Z} - \langle \boldsymbol{G}_\ell, \boldsymbol{Y}_\ell \rangle \geq 0$. The presence of the budget constraint (11) does not affect the arguments, so we drop it without loss of generality.

Since $\boldsymbol{Y}_\ell \in \mathcal{C}$, we may write $Y_\ell = \sum_{i \in \mathcal{A}} w_i \tilde{\boldsymbol{Z}}_i \tilde{\boldsymbol{Z}}_i^\top$ for some $\mathcal{A}$ and a finite set of positive weights w_i summing to 1. Therefore,

$$\langle \boldsymbol{G}_\ell, \boldsymbol{Y}_\ell \rangle = \sum_{i \in \mathcal{A}} w_i \tilde{\boldsymbol{Z}}_i^\top \boldsymbol{G}_\ell \tilde{\boldsymbol{Z}}_i \leq \max_{i \in \mathcal{A}} \tilde{\boldsymbol{Z}}_i^\top \boldsymbol{G}_\ell \tilde{\boldsymbol{Z}}_i,$$

and $g_{\ell+1} \geq 0$ follows. To prove (28), the concavity of f implies that

$$f\left(\boldsymbol{Z}\boldsymbol{Z}^\top\right) \leq f\left(\boldsymbol{Y}_\ell\right) + \langle \nabla f\left(\boldsymbol{Y}_\ell\right), \boldsymbol{Z}\boldsymbol{Z}^\top - \boldsymbol{Y}_\ell \rangle.$$

The desired result follows by maximizing over $\boldsymbol{Z} \in \{\pm 1\}^n$ on both sides.

8.9 Proof of Proposition 5.2

For simplicity, let us drop the iteration index ℓ since we will only consider a single iteration. Let $\boldsymbol{U}$ be as in Proposition 4.4. As in the proof of Proposition 4.4, the objective may be rewritten as

$$f\left(\boldsymbol{Y}\right) = \log\det\left(\boldsymbol{U}^\top\left(\boldsymbol{P} \circ \boldsymbol{Y}\right)\boldsymbol{U}\right) + \log\det\left(\boldsymbol{X}^\top \boldsymbol{X}\right),$$

where $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\top$. For notational convenience, denote

$$\boldsymbol{B} = \boldsymbol{U}\left(\boldsymbol{U}^\top\left(\boldsymbol{P} \circ \boldsymbol{Y}\right)\boldsymbol{U}\right)^{-1}\boldsymbol{U}^\top.$$

Then, we may write the gradient as

$$\boldsymbol{G} \quad = \quad \boldsymbol{P} \circ \boldsymbol{B}$$

$$
\begin{aligned}
&= \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\top\right) \circ B \\
&= (\boldsymbol{I} \circ \boldsymbol{B}) - \boldsymbol{U}\boldsymbol{U}^\top \circ \boldsymbol{B} \\
&= \operatorname{diag}(\boldsymbol{B}) - \boldsymbol{U}\boldsymbol{U}^\top \circ \boldsymbol{B}.
\end{aligned}
$$

Consequently,

$$
\begin{aligned}
\boldsymbol{Z}^\top \boldsymbol{G} \boldsymbol{Z} &= \operatorname{tr}(\operatorname{diag}(\boldsymbol{B})) - \boldsymbol{Z}^\top \left(\boldsymbol{U}\boldsymbol{U}^\top \circ \boldsymbol{B}\right) \boldsymbol{Z} \\
&= \operatorname{tr}\left(\left(\boldsymbol{U}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{U}\right)^{-1}\right) - \boldsymbol{Z}^\top \left(\boldsymbol{U}\boldsymbol{U}^\top \circ \boldsymbol{B}\right) \boldsymbol{Z} \qquad (54)
\end{aligned}
$$

Only the last term in (54) depends on $\boldsymbol{Z}$, so maximizing $\boldsymbol{Z}^\top \boldsymbol{G} \boldsymbol{Z}$ is equivalent to solving

$$
\min_{\boldsymbol{Z} \in \{\pm 1\}^n} \boldsymbol{Z}^\top \left(\boldsymbol{U}\boldsymbol{U}^\top \circ \boldsymbol{B}\right) \boldsymbol{Z}. \qquad (55)
$$

We claim that

$$
\boldsymbol{Z}^\top \left(\boldsymbol{U}\boldsymbol{U}^\top \circ \boldsymbol{B}\right) \boldsymbol{Z} = \| \sum_{i=1}^n Z_i \boldsymbol{w}_i \|^2,
$$

where $\boldsymbol{w}_i = \boldsymbol{u}_i \otimes \boldsymbol{v}_i$ (with $\otimes$ denoting the tensor product) and $\boldsymbol{v}_i^\top$ are the rows of the matrix $\boldsymbol{V} = \boldsymbol{U}\left(\boldsymbol{U}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{U}\right)^{-\frac{1}{2}}$. To see this, observe that

$$
B_{i,j} = \boldsymbol{u}_i^\top \left(\boldsymbol{U}^\top (\boldsymbol{P} \circ \boldsymbol{Y}) \boldsymbol{U}\right)^{-1} \boldsymbol{u}_j = \boldsymbol{v}_i^\top \boldsymbol{v}_j,
$$

whence

$$
\begin{aligned}
\left(\boldsymbol{U}\boldsymbol{U}^\top \circ \boldsymbol{B}\right)_{i,j} &= \left(\boldsymbol{U}\boldsymbol{U}^\top\right)_{i,j} B_{i,j} \\
&= \left(\boldsymbol{u}_i^\top \boldsymbol{u}_j\right) \left(\boldsymbol{v}_i^\top \boldsymbol{v}_j\right) \\
&= (\boldsymbol{u}_i \otimes \boldsymbol{v}_i)^\top (\boldsymbol{u}_i \otimes \boldsymbol{v}_i) \\
&= \boldsymbol{w}_i^\top \boldsymbol{w}_j.
\end{aligned}
$$

Letting $\boldsymbol{W}$ be the matrix whose ith row is $\boldsymbol{w}_i^\top$, we thus have

$$
\boldsymbol{Z}^\top \left(\boldsymbol{U}\boldsymbol{U}^\top \circ \boldsymbol{B}\right) \boldsymbol{Z} = \boldsymbol{Z}^\top \boldsymbol{W} \boldsymbol{W}^\top \boldsymbol{Z} = \|\boldsymbol{W}^\top \boldsymbol{Z}\|^2.
$$

Therefore, (55) may be replaced by

$$
\min_{\boldsymbol{Z} \in \{\pm 1\}^n} \| \sum_{i=1}^n Z_i \boldsymbol{w}_i \|^2.
$$

Now, consider the trace

$$\begin{aligned}
\operatorname{tr}(\boldsymbol{G}) &= \operatorname{tr}(\boldsymbol{P}\circ\boldsymbol{B}) \\
&= \sum_{i=1}^{n} P_{ii}B_{ii} \\
&= \sum_{i=1}^{n}\left(1-\boldsymbol{u}_i^\top\boldsymbol{u}_i\right)\boldsymbol{u}_i^\top\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)^{-1}\boldsymbol{u}_i \\
&= \sum_{i=1}^{n}\boldsymbol{u}_i^\top\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)^{-1}\boldsymbol{u}_i-\sum_{i}\|\boldsymbol{u}_i\|^2\boldsymbol{u}_i^\top\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)^{-1}\boldsymbol{u}_i.
\end{aligned} \tag{56}$$

The first sum in (56) can be worked out as

$$\begin{aligned}
\sum_{i=1}^{n}\boldsymbol{u}_i^\top\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)^{-1}\boldsymbol{u}_i &= \operatorname{tr}\left(\sum_{i=1}^{n}\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)^{-1}\boldsymbol{u}_i\boldsymbol{u}_i^\top\right) \\
&= \operatorname{tr}\left(\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)^{-1}\right).
\end{aligned} \tag{57}$$

Using the bound on the leverage scores, the second sum can be bounded as

$$\begin{aligned}
\sum_{i}\|\boldsymbol{u}_i\|^2\boldsymbol{u}_i^\top\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)^{-1}\boldsymbol{u}_i &= \operatorname{tr}\left(\left(U^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)^{-1}\sum_{i=1}^{n}\|\boldsymbol{u}_i\|^2\boldsymbol{u}_i\boldsymbol{u}_i^\top\right) \\
&\leq L\cdot\operatorname{tr}\left(\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)^{-1}\right).
\end{aligned} \tag{58}$$

Combining (57) and (58), (56) yields

$$\operatorname{tr}(\boldsymbol{G})\geq(1-L)\operatorname{tr}\left(\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)^{-1}\right).$$

Combining this bound with (54) yields

$$\boldsymbol{Z}^\top\boldsymbol{G}\boldsymbol{Z}\leq\operatorname{tr}\left(\left(\boldsymbol{U}^\top(\boldsymbol{P}\circ\boldsymbol{Y})\boldsymbol{U}\right)^{-1}\right)$$

for any $\boldsymbol{Z}$, whence

$$\operatorname{tr}(\boldsymbol{G})\geq(1-\delta)\max_{\boldsymbol{Z}\in\{\pm1\}^n}\boldsymbol{Z}^\top\boldsymbol{G}\boldsymbol{Z}. \tag{59}$$

By the local optimality condition on $\tilde{\boldsymbol{Z}}$, we have

$$\|\sum_{i=1}^{n}\tilde{Z}_i\boldsymbol{w}_i-2\tilde{Z}_j\boldsymbol{w}_j\|^2-\|\sum_{i=1}^{n}\tilde{Z}_i\boldsymbol{w}_i\|^2\leq 0,\qquad j=1,...,n,$$

or, equivalently,

$$\tilde{Z}_j \langle \sum_{i=1}^{n} \tilde{Z}_i \boldsymbol{w}_i, \boldsymbol{w}_j \rangle \leq \|\boldsymbol{w}_j\|^2. \tag{60}$$

Summing up (60) over j yields

$$\begin{aligned}
\sum_{j=1}^{n} \tilde{Z}_j \langle \sum_{i=1}^{n} \tilde{Z}_i \boldsymbol{w}_i, \boldsymbol{w}_j \rangle \leq \|\boldsymbol{w}_j\|^2 &= \langle \sum_{j=1}^{n} \tilde{Z}_j \boldsymbol{w}_j, \sum_{i=1}^{n} \tilde{Z}_i \boldsymbol{w}_i \rangle \\
&= \| \sum_{i=1}^{n} \tilde{Z}_i \boldsymbol{w}_i \|^2 \\
&\leq \sum_{i=1}^{n} \|\boldsymbol{w}_i\|^2 \\
&= \operatorname{tr}\left(\boldsymbol{W}\boldsymbol{W}^\top\right).
\end{aligned}$$

Thus, (54) yields

$$\begin{aligned}
\tilde{\boldsymbol{Z}}^\top \boldsymbol{G} \tilde{\boldsymbol{Z}} &\geq \operatorname{tr}\left(\left(\boldsymbol{U}^\top \left(\boldsymbol{P} \circ \boldsymbol{Y}\right) \boldsymbol{U}\right)^{-1}\right) - \operatorname{tr}\left(\boldsymbol{W}\boldsymbol{W}^\top\right) \\
&= \operatorname{tr}\left(\boldsymbol{G}\right),
\end{aligned} \tag{61}$$

as required. Combining (59) and (61) yields (29), as desired.

8.10 Proof of Lemma 8.1

To prove Lemma 8.1(i), let $\boldsymbol{y} \in \mathbb{R}^m$ and observe that

$$\boldsymbol{y}^\top \boldsymbol{C}_U \boldsymbol{y} = \sum_{v \in U} \left(\boldsymbol{b}_v^\top \boldsymbol{y}\right)^2 = \sum_{v \in U} \left(\sum_{f \in \delta(v)} y_f\right)^2.$$

By the Cauchy-Schwarz inequality,

$$\left(\sum_{f \in \delta(v)} y_f\right)^2 \leq |\delta(v)| \sum_{f \in \delta(v)} y_f^2.$$

Consequently,

$$\begin{aligned}
\boldsymbol{y}^\top \boldsymbol{C}_U \boldsymbol{y} &\leq \sum_{v \in U} |\delta(v)| \sum_{f \in \delta(v)} y_f^2 \\
&\leq \left(\max_{v \in V} |\delta(v)|\right) \cdot \sum_{v \in U} \sum_{f \in \delta(v)} y_f^2
\end{aligned}$$

$$\leq \left(\max_{v\in V} |\delta(v)|\right)\cdot 2\sum_{f\in E} y_f^2 \tag{62}$$

$$\leq 6\|\boldsymbol{y}\|_2^2, \tag{63}$$

where (62) uses the fact that each edge has two endpoints. Since $\boldsymbol{C}_U \succeq 0$, (63) implies $\boldsymbol{C}_U \preceq 6\cdot\boldsymbol{I}$. The proof for $\boldsymbol{C}_{U^c}$ is analogous.

Lemma 8.1(ii) can be shown by taking G' to be the disjoint union of multiple copies of G. Solving one instance will solve the other.

To show Lemma 8.1(iii), let $0 \leq \lambda_1, ..., \lambda_m \leq 6$ be the eigenvalues of $\boldsymbol{C}$. Then,

$$\log\det(\boldsymbol{I}_m + \varepsilon\boldsymbol{C}) = \sum_{i=1}^m \log(1+\varepsilon\lambda_i).$$

Consider the function $r(t) = \log(1+t) - t + \frac{t^2}{2}$. Then, $r(0) = 0$, and $r'(t) = \frac{t^2}{1+t} \geq 0$ for $t \geq 0$. Therefore,

$$0 \leq r(t) = \int_0^t \frac{s^2}{1+s}ds \leq \int_0^t s^2 ds = \frac{t^3}{3}. \tag{64}$$

On the other hand, for $t \in (-1, 0)$, we have Now, we write

$$\begin{aligned}\log\det(\boldsymbol{I}_m + \varepsilon\boldsymbol{C}) &= \sum_{i=1}^m \varepsilon\lambda_i - \sum_{i=1}^m \frac{(\varepsilon\lambda_i)^2}{2} + \sum_{i=1}^m r(\varepsilon\lambda_i)\\ &\leq \sum_{i=1}^m \varepsilon\lambda_i - \sum_{i=1}^m \frac{(\varepsilon\lambda_i)^2}{2} + \sum_{i=1}^m \frac{(\varepsilon\lambda_i)^3}{3} \qquad (65)\\ &\leq \varepsilon\,\mathrm{tr}(\boldsymbol{C}) - \frac{\varepsilon^2}{2}\mathrm{tr}\left(\boldsymbol{C}^2\right) + 72\varepsilon^3 m,\end{aligned}$$

where (65) applies (64), and the last line uses $\lambda_i \leq 6$. On the other hand,

$$\log\det(\boldsymbol{I}_m + \varepsilon\boldsymbol{C}) \geq \sum_{i=1}^m \varepsilon\lambda_i - \sum_{i=1}^m \frac{(\varepsilon\lambda_i)^2}{2} = \varepsilon\,\mathrm{tr}(\boldsymbol{C}) - \frac{\varepsilon^2}{2}\mathrm{tr}\left(\boldsymbol{C}^2\right),$$

completing the proof of the claim.

8.11 Proof of Lemma 8.2

Let $0 \preceq \boldsymbol{A}, \boldsymbol{B} \preceq \boldsymbol{I}_d$ be symmetric. Define $\boldsymbol{C}_t = \boldsymbol{A} + t(\boldsymbol{B} - \boldsymbol{A})$ with $t \in [0,1]$. By the fundamental theorem of calculus, we have

$$\det(\boldsymbol{B}) - \det(\boldsymbol{A}) = \int_0^1 \mathrm{tr}\left(\mathrm{adj}\left(\boldsymbol{C}_t(\boldsymbol{B} - \boldsymbol{A})\right)\right),$$

whence

$$|\det(\boldsymbol{B}) - \det(\boldsymbol{A})| \leq \sup_{\boldsymbol{C} \in \mathbb{S}^d : 0 \preceq \boldsymbol{C} \preceq \boldsymbol{I}_d} \| \operatorname{adj}(\boldsymbol{C}) \|_* \|\boldsymbol{B} - \boldsymbol{A}\|_2,$$

where $\| \cdot \|_*$ denotes the nuclear norm.

Let $\boldsymbol{C} = \boldsymbol{U} \operatorname{diag}(\lambda_1, ..., \lambda_d) \boldsymbol{U}^\top$ with $0 \leq \lambda_i \leq 1$ for all i. Then,

$$\operatorname{adj}(\boldsymbol{C}) = \boldsymbol{U} \operatorname{diag}\left(\prod_{i \neq 1} \lambda_i, ..., \prod_{i \neq d} \lambda_i \right) \boldsymbol{U}^\top$$

whence

$$\| \operatorname{adj}(\boldsymbol{C}) \|_* = \sum_{j=1}^{d} \prod_{i \neq j} \lambda_i \leq d.$$

Thus, d is a valid Lipschitz constant. To see that it is sharp, take $\boldsymbol{A} = \boldsymbol{I}_d$ and $\boldsymbol{B} = (1-t)\boldsymbol{I}_d$. Then, $\|\boldsymbol{A} - \boldsymbol{B}\|_2 = t$ and $|\det(A) - \det(B)| = 1 - (1-t)^d$. Taking $t \to 0$, we have $\frac{1-(1-t)^d}{t} \to d$, from which it follows that no smaller Lipschitz constant is possible.

9 Appendix: performance comparison tables

In the following, we present tables of numerical results for the datasets discussed in Sections 6.3-6.4. Tables 5-6 report numbers for the 401(k) pension data. Recall that this problem has $n = 100$ and $d = 12$. Table 5 covers the unconstrained version of the problem, while Table 6 covers the constrained version with $k = 15$. Overall, the results are similar to those seen in Section 6.2, with CG consistently producing the best-performing solutions. Improved RG matches these solutions, but is not able to improve on them within an hour, though it does marginally improve on the upper bound obtained from the convex-hull relaxation.

Tables 7-8 present analogous results for the STAR data of Section 6.4. The situation is overall similar, with the objective values being higher than the previous problems due to the higher dimensionality of the data. CG achieves near-optimal performance in both unconstrained and constrained variants. Conic optimization is not able to find a feasible solution for the constrained version.

		LB	UB	Time (s)
Relaxations	Two-log-determinant	N/A	38.810	<0.01
	SDP (Prop. 4.3)	N/A	37.772	8.94
	Convex-hull	N/A	37.761	1681.09
Assignments	Coin-flipping	36.532	N/A	<0.01
	Pivotal sampling	36.641	N/A	<0.01
	Conic optimization	37.594	37.803	3600
	Row generation	37.306	38.129	3600
	Improved RG	37.666	37.731	3600
	Best CG solution	37.666	N/A	5.54

Table 5: Performance comparison for 401(k) pension data with $n = 100$ and $d = 12$.

		LB	UB	Time (s)
Relaxations	Two-log-determinant	N/A	31.394	0.20
	SDP (Prop. 4.3)	N/A	31.176	13.30
	Convex-hull	N/A	30.867	1884.59
Assignments	Coin-flipping	25.297	N/A	<0.01
	Pivotal sampling	26.136	N/A	<0.01
	Conic optimization	30.332	31.037	3600
	Row generation	28.722	34.270	3600
	Improved RG	30.392	30.828	3600
	Best CG solution	30.392	30.867	1884.59

Table 6: Performance comparison for 401(k) pension data with $n = 100$, $d = 12$, and $k = 15$.

		LB	UB	Time (s)
Relaxations	Two-log-determinant	N/A	144.947	<0.01
	SDP (Prop. 4.3)	N/A	141.581	337.50
	Convex-hull	N/A	96.397	25.782
Assignments	Coin-flipping	105.186	N/A	<0.01
	Pivotal sampling	136.238	N/A	<0.01
	Conic optimization	127.368	144.708	3600
	Row generation	137.842	152.514	3600
	Improved RG	140.703	140.902	3600
	Best CG solution	140.703	N/A	16.55

Table 7: Performance comparison for STAR data with $n = 200$ and $d = 35$.

		LB	UB	Time (s)
Relaxations	Two-log-determinant	N/A	134.496	1.06
	SDP (Prop. 4.3)	N/A	130.380	504.38
	Convex-hull	N/A	129.124	1046.38
Assignments	Coin-flipping	74.644	N/A	<0.01
	Pivotal sampling	73.337	N/A	<0.01
	Conic optimization	N/A	134.251	3600
	Row generation	115.338	185.620	3600
	Improved RG	125.233	129.124	3600
	Best CG solution	125.233	129.124	1046.38

Table 8: Performance comparison for STAR data with $n = 200$, $d = 35$, and $k = 40$.