Grouped Variable Selection with Discrete Optimization: Computational and Statistical Perspectives

Hussein Hazimeh* Rahul Mazumder† Peter Radchenko‡

April 16, 2021

Abstract

We present a new algorithmic framework for grouped variable selection that is based on discrete mathematical optimization. While there exist several appealing approaches based on convex relaxations and nonconvex heuristics, we focus on optimal solutions for the ℓ_0-regularized formulation, a problem that is relatively unexplored due to computational challenges. Our methodology covers both high-dimensional linear regression and nonparametric sparse additive modeling with smooth components. Our algorithmic framework consists of approximate and exact algorithms. The approximate algorithms are based on coordinate descent and local search, with runtimes comparable to popular sparse learning algorithms. Our exact algorithm is based on a standalone branch-and-bound (BnB) framework, which can solve the associated mixed integer programming (MIP) problem to certified optimality. By exploiting the problem structure, our custom BnB algorithm can solve to optimality problem instances with 5×10^6 features in minutes to hours – over 1000 times larger than what is currently possible using state-of-the-art commercial MIP solvers. We also explore statistical properties of the ℓ_0-based estimators. We demonstrate, theoretically and empirically, that our proposed estimators have an edge over popular group-sparse estimators in terms of statistical performance in various regimes.

1 Introduction

Sparsity plays a ubiquitous role in modern statistical regression, especially when the number of predictors is large relative to the number of observations. In this paper, we focus on the case where predictors have a natural group structure. Typical examples where such a structure appears are models with multilevel categorical predictors and models that represent nonlinear effects of continuous variables using basis functions [15, 59, 23]. Grouping may also arise from scientifically meaningful prior knowledge about the collection of the predictor variables. More specifically, we consider the usual linear regression framework with response $\mathbf{y}_{n\times 1}$ and model matrix $\mathbf{X}_{n\times p} = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$. We suppose that the p predictors are divided into q pre-specified, non-overlapping groups. For a given $\boldsymbol{\beta} \in \mathbb{R}^p$ and each $g \in \{1, ..., q\}$, we denote by $\boldsymbol{\beta}_g$ the sub-vector of $\boldsymbol{\beta}$ whose coefficients correspond to the predictors in group g. Following the traditional approach in high-dimensional regression, we assume that few of the regression coefficients are

*Massachusetts Institute of Technology, hazimeh@mit.edu
†Massachusetts Institute of Technology, rahulmaz@mit.edu
‡University of Sydney, peter.radchenko@sydney.edu.au

nonzero, i.e., the model is sparse. This leads to a natural generalization of the classical best subset selection problem in linear regression [40, 8] to the group setting:

$$\min_{\boldsymbol{\beta}} \quad \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_0 \sum_{g=1}^{q} \mathbf{1}(\boldsymbol{\beta}_g \neq \mathbf{0}), \tag{1}$$

where $\mathbf{1}(\cdot)$ is the indicator function, and λ_0 is a non-negative regularization parameter that controls the number of nonzero groups selected. We will refer to Problem (1) as the Group ℓ_0 problem.

Problem (1) is NP-Hard [43] and poses computational challenges. A rich body of prior work explores sparsity-inducing methods to obtain approximate solutions to (1). Popular methods include: convex optimization based procedures, such as Group Lasso [59], which is a generalization of the Lasso approach [54] to the grouped setting, and local solutions to nonconvex optimization problems arising from group-nonconvex regularizers, such as SCAD, MCP and others [61, 30]. Despite the appeal of these approaches, the statistical and computational aspects of *optimal* solutions to (1) remain to be understood at a deeper level. To this end, we aim to advance the computational frontiers of Problem (1) using novel tools from discrete optimization. Our proposed combinatorial optimization-based algorithms are scalable. In particular, they can deliver optimal solutions to (1) for instances that are much larger than state-of-the-art approaches. We also develop a better understanding of the statistical properties of Problem (1) both theoretically and empirically.

Computation. We propose new algorithms based on combinatorial optimization for solving Problem (1) and its variants. First we present approximate algorithms: they deliver high-quality solutions using a combination of cyclic coordinate descent and local combinatorial optimization [24]. These algorithms have runtimes comparable to popular approaches for grouped variable selection (for example, Group Lasso or MCP), but deliver solutions with considerably improved statistical performance (for example, in terms of prediction and variable selection), as we demonstrate in our experiments. Our approximate algorithms deliver good-quality feasible solutions to (1) but are unable to certify (global) optimality of solutions via matching lower bounds on the optimal objective value of (1). Certifying optimality is not only important from a methodological perspective but can also be beneficial in practice for mission-critical applications. For example, having certifiably optimal solutions can engender trust and provide transparency in consequential applications such as healthcare. Thus, we propose a new tailored branch-and-bound based optimization framework for solving (1) to certifiable optimality.

In our exact (global optimization) framework, we formulate the Group ℓ_0 problem as a Mixed Integer Program (MIP). However, in a departure from earlier work [8, 7], we propose a custom branch-and-bound (BnB) algorithm to solve the MIP. Indeed, MIP-based techniques have gained considerable traction recently to solve to (near) optimality the best subset selection problem, where all groups are of size one [8, 7, 36, 38, 24, 58, 26]. All these works, with the exception of [26], leverage capabilities of powerful commercial MIP solvers such as Gurobi and CPLEX. These solvers have gained wide adoption in the past two decades due to major advances in algorithms and software development [10, 31]. However, these general-purpose solvers may take several hours to certify optimality on small instances (for example, with $p = 1000$). In contrast, our custom BnB algorithm exploits problem-specific structure to scale to much larger instances. For example, it can solve to optimality instances with $p = 5 \times 10^6$ – this is 1000 times larger than

what can be handled using Gurobi's MIP-solver. Our BnB algorithm generalizes to the grouped setting the approach of [26] developed for the best subset selection problem.

Statistical properties. Statistical properties of Group Lasso have been extensively studied, and it has been shown, both empirically and theoretically, that it performs well in sparse high-dimensional settings [16, 2, 42, 28, 57, 34, 44], under certain assumptions on the data. However, Group Lasso also has its shortcomings, similar to those of Lasso in high dimensional linear regression [8, 15, 24]. More specifically, depending on the penalty weight, the resulting model may either be very dense or, alternatively, comes with overly shrunk nonzero coefficients. This problem is aggravated when the groups are correlated with each other, as Group Lasso tends to bring in all of the correlated groups in lieu of searching for a more parsimonious model. For further discussions of these issues in the special case of Lasso see, for example, [62, 37, 15, 8], and the references therein. In this paper, we demonstrate, both empirically and theoretically, that the Group ℓ_0 methodology has advantages over its Group Lasso counterpart in a variety of regimes. In particular, as a consequence of directly controlling the sparsity level in the optimization problem, our framework leads to substantially sparser models under similar data fidelity. Moreover, in many scenarios where the predictors are highly correlated, our approach performs better in terms of both estimation and prediction.

Additive models with ℓ_0-sparsity. In addition to linear models, we also study an important example of regression with group structure that arises in high-dimensional sparse additive modeling [23, 22]. Here, we estimate a nonparametric multivariate regression function in q covariates, $(x_1, \ldots, x_q)$, which we model as a sparse additive sum of the form $\sum_{j \in S} f_j(x_j)$, where $S \subset \{1, \ldots, q\}$. In this setting, each group generally corresponds to the basis representation of a given additive component, one for each of the q predictors. Because the groups are allowed to be large, additional regularization needs to be imposed, typically in the form of a roughness type penalty on the regression functions. A number of successful Group Lasso-based approaches have been proposed and analyzed in this setting – see, for example, [39, 51, 29, 32, 49, 60] and the references therein. To our knowledge, this is the first paper to explore statistical and computational aspects of Group ℓ_0-based formulations in the context of sparse additive modeling. We show theoretically and empirically that Group ℓ_0 based methods enjoy certain statistical advantages when compared to the Group Lasso-based counterparts.

Contributions. The focus of this paper is on Problem (1) and the sparse additive modeling problem (which can be formulated as a variant of Problem (1), as we discuss in Section 2). Our main contributions for these two problems can be summarized as follows:

- We develop fast approximate algorithms, based on first-order and local combinatorial optimization. We establish convergence guarantees for these algorithms and provide useful characterizations of the corresponding local minima. Our experiments indicate that these algorithms can have an edge in terms of statistical performance over popular alternatives for grouped variable selection.

- We present mixed integer second order cone program (MISOCP) formulations for the Group ℓ_0-based estimators; and design a novel specialized, nonlinear branch-and-bound (BnB) framework for solving the MISOCP to global optimality. Our custom BnB solver can handle instances with 5×10^6 variables – more than a 1000 times larger than what can be handled by state-of-the-art commercial MISOCP solvers.

- We establish non-asymptotic prediction and estimation error bounds for our proposed estimators, for both the high-dimensional linear regression and sparse additive modeling problems. We show that under the assumption of sparsity, these error bounds compare favorably with the ones for Group Lasso.
- We demonstrate empirically that our approach appears to outperform the state of the art (for example, Group Lasso and available algorithms for nonconvex penalized estimators) in a variety of high-dimensional regimes and under different statistical metrics (for example, prediction, estimation, and variable selection).

Organization. In Section 2, we present formulations for the Group ℓ_0 and sparse additive modeling problems. Section 3 presents approximate algorithms based on first-order and local combinatorial optimization algorithms. Then, in Section 4, we present our exact MIP algorithm. Statistical properties of our approach are investigated in Section 5. Section 6 presents computational experiments. Technical proofs and additional computational details are provided in the supplement.

Notation. For any non-negative integer k, we denote the set $\{1, ..., k\}$ by $[k]$. The complement of a set A is denoted by A^c. We denote the index sets corresponding to the q groups of predictors by $\mathcal{G}_g$, for $g \in [q]$ so that $\cup_{g=1}^{q} \mathcal{G}_g = [p]$ and $\mathcal{G}_g \cap \mathcal{G}_\ell = \emptyset$ for all $g \neq \ell$. For a vector $\boldsymbol{\theta}$, we use the notation $\text{Supp}(\boldsymbol{\theta})$ to denote the group support, i.e., $\text{Supp}(\boldsymbol{\theta}) = \{g \mid \boldsymbol{\theta}_g \neq 0, g \in [q]\}$. We also define a measure of ℓ_0-group sparsity (i.e., number of nonzero groups): $G(\boldsymbol{\theta}) := \sum_{g=1}^{q} \mathbf{1}(\boldsymbol{\theta}_g \neq \mathbf{0})$. We denote the gradient of a scalar-valued function, say $J(\boldsymbol{\theta})$, by $\nabla J(\boldsymbol{\theta})$. Moreover, we use the notation $\nabla_{\boldsymbol{\theta}_g} J(\boldsymbol{\theta})$ to refer to the subvector of $\nabla J(\boldsymbol{\theta})$ corresponding to the variables in $\boldsymbol{\theta}_g$. Vectors and matrices are denoted in boldface.

2 Optimization problems considered

In this section, we present optimization formulations for the Group ℓ_0 approach (and its variants), as well as the ℓ_0-sparse additive function estimation approach.

2.1 Group ℓ_0 with ridge regularization

The algorithms discussed in this paper apply to the Group ℓ_0 estimator (1) with an optional ridge regularization term:

$$\min_{\boldsymbol{\beta}} \quad \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_0 \sum_{g=1}^{q} \mathbf{1}(\boldsymbol{\beta}_g \neq \mathbf{0}) + \lambda_2 \|\boldsymbol{\beta}\|_2^2, \tag{2}$$

where $\lambda_0 > 0$ controls the number of selected groups, and $\lambda_2 \geq 0$ controls the strength of the ridge regularization. Our proposed algorithms apply to *both* settings: $\lambda_2 = 0$ and $\lambda_2 > 0$ in Problem (2). The choice of the ridge term in (2) is motivated by earlier work in the context of best-subset selection [38, 24], which suggest that when the signal-to-noise ratio (SNR) is low, additional ridge regularization can improve the prediction performance of best-subset selection (both theoretically and empirically). Additionally, as discussed in Section 4.2, the choice $\lambda_2 > 0$, allows for deriving stronger MIP formulations by appealing to perspective formulations [19, 21].

2.2 Nonparametric additive models with ℓ_0-sparsity

In the multivariate setting, estimating the conditional mean function $\mathbb{E}(y|\mathbf{x}) = f(x_1, \ldots, x_q)$ becomes notoriously difficult, due to curse of dimensionality. To overcome this problem, additive approximation schemes [22] are commonly used as an effective methodology: $f(\mathbf{x}) = \sum_{j=1}^{q} f_j(x_j)$. A popular approach [see, for example, 56] is to choose f_j from some smooth functional class $\mathcal{C}_j$, such as the class of twice continuously differentiable functions. Given the observations $(y_i, \mathbf{x}_i)$, $i \in [n]$, the additive model $f(\mathbf{x})$ can be estimated by solving the following optimization problem:

$$\min_{f} \quad \sum_{i=1}^{n} (y_i - \sum_{j=1}^{q} f_j(x_{ij}))^2 + \lambda \sum_{j=1}^{q} \mathrm{Pen}(f_j), \tag{3}$$

where $\mathrm{Pen}(f_j)$ is a roughness penalty that controls the amount of smoothness in function f_j.

A key ingredient in the additive function fitting framework is the estimation of a univariate smooth regression function based on observations $(y_i, u_i), i \in [n]$. Suppose, for simplicity, that the u_is are distinct and $u_i \in [0, 1]$ for all i. For illustration, let us take $\mathrm{Pen}(g) = \int_0^1 (g''(u))^2 du$. Then, the solution to the corresponding (infinite dimensional) univariate problem is of the form: $g(u) = \alpha_0 + \alpha_1 u + \sum_{j=1}^{n} \gamma_j N_j(u)$, where $N_j(u)$ are some cubic spline basis functions, such as truncated power series functions, natural cubic splines or the B-spline basis functions, with knots chosen at the distinct data points $u_i, i \in [n]$. Note that $\int_0^1 (g''(u))^2 du = \boldsymbol{\gamma}' \boldsymbol{\Omega} \boldsymbol{\gamma}$, where $\boldsymbol{\Omega}$ is an $n \times n$ positive definite matrix with the elements $\omega_{ij} = \int_0^1 N_i''(u) N_j''(u) du$. If we refer to the corresponding functional class as $\mathcal{C}$, define the elements of $\mathbf{g}$ as $g_i := g(u_i)$, for $i = 1, \ldots, n$, and let $\|\mathbf{g}\|_{\mathcal{C}}^2 := \boldsymbol{\gamma}' \boldsymbol{\Omega} \boldsymbol{\gamma}$, then the univariate optimization problem is equivalent to

$$(\hat{g}_1, \ldots, \hat{g}_m) = \hat{\mathbf{g}} \in \arg\min \ \|\mathbf{y} - \mathbf{g}\|_2^2 + \lambda \|\mathbf{g}\|_{\mathcal{C}}^2. \tag{4}$$

Problem (4) is a generalized least squares problem in $(\alpha_0, \alpha_1, \boldsymbol{\gamma})$. A direct extension to the additive model setting is given by the following formulation:

$$\min \quad \|\mathbf{y} - \sum_{j=1}^{q} \mathbf{f}_j\|_2^2 + \lambda \sum_{j=1}^{q} \|\mathbf{f}_j\|_{\mathcal{C}_j}^2, \tag{5}$$

where $f_j \in \mathcal{C}_j$ and $\mathbf{f}_j = (f_j(x_{ij}), \ldots, f_j(x_{nj}))$.

We wish to impose sparsity on the additive components f_j, $j \in [q]$, which naturally leads to the following optimization problem:

$$\min \quad \|\mathbf{y} - \sum_{j=1}^{q} \mathbf{f}_j\|_2^2 + \lambda_0 \sum_{j=1}^{q} \mathbf{1}(\mathbf{f}_j \neq 0) + \lambda \sum_{j=1}^{q} \|\mathbf{f}_j\|_{\mathcal{C}_j}^2. \tag{6}$$

We note that the choice $\mathrm{Pen}(f_j) = \sqrt{\int (f_j''(u))^2 du}$ leads to the optimization problem

$$\min \quad \|\mathbf{y} - \sum_{j=1}^{q} \mathbf{f}_j\|_2^2 + \lambda_0 \sum_{j=1}^{q} \mathbf{1}(\mathbf{f}_j \neq 0) + \lambda \sum_{j=1}^{q} \|\mathbf{f}_j\|_{\mathcal{C}_j}. \tag{7}$$

Problems (6) and (7) are close cousins and result in similar estimators. The terms $\sum_j \|\mathbf{f}_j\|_{\mathcal{C}_j}$ and $\sum_j \|\mathbf{f}_j\|_{\mathcal{C}_j}^2$ encourage smoothness in each of the additive components, while the sum of indicators

directly controls the number of included predictors. In Section 5, we establish theoretical error bounds for the estimator that corresponds to Problem (7).

Connections with Group Lasso-type penalization schemes. For Grouped Lasso-type penalization schemes, the choice of the penalty becomes rather subtle. Problem (3) with $\text{Pen}(f_j) = \|\mathbf{f}_j\|^2_{\mathcal{C}_j}$ does *not* induce sparsity in $\|\mathbf{f}_j\|_{\mathcal{C}_j}$'s for finite λ. Alternatively, the choice $\text{Pen}(f_j) = \|\mathbf{f}_j\|_{\mathcal{C}_j}$ does result in several components $\|\mathbf{f}_j\|_{\mathcal{C}_j}$ being set to zero when λ is large. Note, however, that $\|\mathbf{f}_j\|_{\mathcal{C}_j} = 0$ does not imply $f_j = 0$. This is because $\|\mathbf{f}_j\|_{\mathcal{C}_j}$ is a seminorm that is not affected by the linear components of f_j. To set $f_j = 0$ one needs to include the linear components into the penalty. To overcome these limitations, alternatives have been proposed – here we mention some penalization schemes that are used to encourage selection and smoothness. One possible choice [39] is $\text{Pen}(f_j) = \sqrt{\|\mathbf{f}_j\|_2^2 + \lambda' \|\mathbf{f}_j\|^2_{\mathcal{C}_j}}$, where $\|\mathbf{f}_j\|_2$ denotes the usual ℓ_2 norm of the vector $\mathbf{f}_j$. The corresponding penalization term is $\lambda \sum_j \text{Pen}(f_j)$, and, hence, the parameters λ and λ' jointly control smoothness and sparsity. The sum of $\|\mathbf{f}_j\|_2^2$ and $\lambda' \|\mathbf{f}_j\|^2_{\mathcal{C}_j}$ leads to double penalization, thereby potentially resulting in unwanted shrinkage that may interfere with variable selection. Similar issues arise with the choices $\text{Pen}(f_j) = \|\mathbf{f}_j\|_2 + \lambda' \|\mathbf{f}_j\|_{\mathcal{C}_j}$, considered in [15], and $\text{Pen}(f_j) = \sqrt{\|\mathbf{f}_j\|_2^2 + \lambda' \|\mathbf{f}_j\|^2_{\mathcal{C}_j}} + \tilde{\lambda} \|\mathbf{f}_j\|^2_{\mathcal{C}_j}$, which appears in [39].

Thus, the choice of $\text{Pen}(f_j)$ plays an important role in obtaining sparsity for Lasso-type regularization methods. In contrast, the levels of smoothness and sparsity are controlled separately in the ℓ_0-formulations: Problems (6) and (7). Group Lasso-type penalization schemes may be interpreted as convex relaxations of the ℓ_0-penalty appearing in Problem (7), as discussed in the Supplement A.

Other choices of smooth function classes. We note that the above framework, where each additive component is taken to be a cubic spline, can be generalized to more flexible smooth nonparametric models, depending upon the choice of $\text{Pen}(\cdot)$ and the functional classes $\mathcal{C}_j$s. For example, one may consider the class of functions that are τ times continuously differentiable, together with the choice $\text{Pen}(f_j) = \int f_j^{(\tau)}(u) du$, where $f^{(\tau)}$ denotes the τth derivative of f_j – solutions to these problems are given by splines of order τ [56].

Another popular paradigm pursued in several works [32, 33, 50] is the Reproducing Kernel Hilbert Space (RKHS) framework, wherein every $\mathcal{C}_j$ is taken to be a Hilbert space encouraging some form of smoothness on f_j. Here, $\text{Pen}(f_j) = \|\mathbf{f}_j\|_{K_j}$ is an appropriate Hilbert space norm.

2.3 General problem formulation considered in this paper

Our focus in this paper is on Problem (2) and the sparse additive modeling problems defined in (6) and (7). These three problems can all be formulated as follows:

$$\min_{\boldsymbol{\beta}} \quad \boldsymbol{\beta}' \mathbf{P} \boldsymbol{\beta} + \langle \mathbf{a}, \boldsymbol{\beta} \rangle + \lambda_0 G(\boldsymbol{\beta}) + \lambda_1 \sum_{g=1}^{q} \|\mathbf{P}_g \boldsymbol{\beta}_g\|_2, \tag{8}$$

for suitable choices of $\boldsymbol{a}$, $\mathbf{P} \succeq \mathbf{0}$, $\mathbf{P}_g \succ \mathbf{0}$, $g \in [q]$, where we recall that $G(\boldsymbol{\beta}) := \sum_{g=1}^{q} \mathbf{1}(\boldsymbol{\beta}_g \neq \mathbf{0})$. The term $\sum_{g=1}^{q} \|\mathbf{P}_g \boldsymbol{\beta}_g\|_2$ is only used for the sparse additive modeling problem in (7). Problems (1) and (6) can be obtained by setting $\lambda_1 = 0$ and choosing $\mathbf{P}$ and $\boldsymbol{a}$ appropriately.

To simplify the presentation, we apply a change of variable in Problem (8): $\boldsymbol{\theta}_g = \mathbf{P}_g^{\frac{1}{2}} \boldsymbol{\beta}_g$ for

$g \in [q]$. This leads to the following equivalent problem:

$$\min_{\boldsymbol{\theta}} \quad h(\boldsymbol{\theta}) := \underbrace{\boldsymbol{\theta}'\mathbf{W}\boldsymbol{\theta} + \langle \mathbf{b}, \boldsymbol{\theta} \rangle}_{:=\ell(\boldsymbol{\theta})} \quad + \quad \underbrace{\lambda_0 G(\boldsymbol{\theta}) + \lambda_1 \sum_{g=1}^{q} \|\boldsymbol{\theta}_g\|_2}_{:=\Omega(\boldsymbol{\theta})}, \tag{9}$$

for appropriately defined[1] $\mathbf{W}$ and $\mathbf{b}$. Our algorithmic development will focus on (9).

Overview of our algorithms: Problem (9) is nonconvex due to the discontinuity in $G(\boldsymbol{\theta})$. In Section 3, we design fast algorithms that can obtain high-quality approximate solutions for this problem. In Section 4, we develop an exact algorithmic framework, based on a custom MIP solver, which obtains certifiably optimal solutions to (9). Our algorithm constructs: (i) a sequence of feasible solutions, whose objective values are valid upper bounds, and (ii) a sequence of lower bounds (a.k.a. dual bounds). As our BnB algorithm progresses, these upper and lower bounds converge towards the optimal objective of Problem (9). The solver terminates and certifies optimality when the upper and lower bounds match[2]. Our experiments indicate that high-quality initial solutions, as available from the algorithms presented in Section 3, can significantly speed up convergence and reduce memory requirements in our BnB algorithm.

3 Approximate Algorithms

In this section, we develop fast approximate algorithms to obtain high quality local minimizers for Problem (9). While these algorithms do not deliver certificates of optimality (via dual bounds), they attain nearly-optimal (and at times optimal) solutions to many statistically challenging instances, in running times comparable to group Lasso-based algorithms.

A main workhorse of our approximate algorithms is a nonstandard application of cyclic block coordinate descent (BCD) to the discontinuous objective function (9). We draw inspiration from the appealing scalability properties of coordinate descent in sparse learning problems [see, for example, 20, 3, 24]. Our second algorithm is based on local combinatorial search and is used to improve the quality of solutions obtained by BCD. We establish convergence guarantees for these two algorithms.

Our algorithms arise from studying necessary optimality conditions for Problem (9). To this end, we show that the quality of solutions obtained by BCD are of higher quality than local solutions corresponding to the popular proximal gradient descent (PGD) [47] algorithm[3]. The local minimizers corresponding to local combinatorial search form a smaller subset of those available from BCD. In this section, we establish the following hierarchy among the classes of local minima:

$$\text{Global Minima} \subseteq \text{Local Search Minima} \subseteq \text{BCD Minima} \subseteq \text{PGD Minima}. \tag{10}$$

[1] Let $\mathbf{D}_1 = \text{diag}(\mathbf{P}_1^{\frac{1}{2}}, \ldots, \mathbf{P}_q^{\frac{1}{2}})$ be a block diagonal matrix. Then $\mathbf{W} = \mathbf{D}_1^{-1}\mathbf{P}\mathbf{D}_1^{-1}$, and $\mathbf{b} = \mathbf{D}_1^{-1}\mathbf{a}$.

[2] In practice, MIP solvers terminate when the difference between the upper and lower bounds are below a small, user-defined threshold.

[3] Though PGD is popularly used in the context of convex optimization problems, it also leads to useful algorithms for nonconvex sparse learning problems. In particular, PGD for our problem can be viewed as a generalization of the iterative hard thresholding (IHT) algorithm [12] to the group setting.

Above, PGD minima correspond to the fixed points of the PGD algorithm; they include all the fixed points of our proposed BCD algorithm. As we move from right to left in the above hierarchy, the classes become smaller, i.e., they impose stricter necessary optimality conditions. At the top of the hierarchy we have the global minimizers of the problem, which can be obtained using our exact MIP-based framework (we discuss this in Section 4). Our approximate algorithms are inspired by recent work [24] on the sparse regression problem, but the approach presented here has notable differences. In particular, the coordinate descent algorithm in [24] performs exact minimization per coordinate, which can be computationally expensive when extended to the group setting. Thus, our proposed BCD algorithm performs inexact minimization per group. In addition, the presence of ℓ_2 norms in our objective function makes the analysis for the rate of convergence for our algorithm different.

3.1 Block Coordinate Descent

We present a cyclic BCD algorithm to obtain good feasible solutions to Problem (9) and establish convergence guarantees. We first introduce a useful upper bound for $\ell(\boldsymbol{\theta})$. For every $g \in [q]$, we define $S_g = \{(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \mid \boldsymbol{\theta}_i = \tilde{\boldsymbol{\theta}}_i,\ \forall i \in [q] \text{ s.t. } i \neq g\}$. By the Block Descent Lemma [6], the following upper bound holds for every $g \in [q]$:

$$\ell(\boldsymbol{\theta}) \leq \ell(\tilde{\boldsymbol{\theta}}) + \langle \nabla_{\boldsymbol{\theta}_g} \ell(\tilde{\boldsymbol{\theta}}), \boldsymbol{\theta}_g - \tilde{\boldsymbol{\theta}}_g \rangle + \frac{L_g}{2} \|\boldsymbol{\theta}_g - \tilde{\boldsymbol{\theta}}_g\|_2^2, \qquad \forall (\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \in S_g, \tag{11}$$

where L_g is the "group-wise" Lipschitz constant of $\nabla \ell(\boldsymbol{\theta})$, i.e., L_g is a constant which satisfies: $\|\nabla_{\boldsymbol{\theta}_g} \ell(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}_g} \ell(\tilde{\boldsymbol{\theta}})\|_2 \leq L_g \|\boldsymbol{\theta}_g - \tilde{\boldsymbol{\theta}}_g\|_2$, for all $(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \in S_g$. Since $\ell(\boldsymbol{\theta})$ is a quadratic function, $L_g = 2\sigma_{\max}(\mathbf{W}_g)$, where $\mathbf{W}_g$ is the submatrix of $\mathbf{W}$ with columns and rows restricted to group g, and $\sigma_{\max}(\cdot)$ denotes the largest eigenvalue.

Cyclic BCD sequentially minimizes the objective of (9) with respect to one group of variables while the other groups are held fixed. Let $\boldsymbol{\theta}^l$ be the iterate obtained by the algorithm after the l-th iteration. Then, in iteration $l+1$, the variables in a group g (say), are updated while the other groups are held fixed. Specifically, we have $(\boldsymbol{\theta}^l, \boldsymbol{\theta}^{l+1}) \in S_g$. Using (11) with $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^l$, $\hat{L}_g > L_g$ and adding $\Omega(\boldsymbol{\theta})$ to both sides we get:

$$h(\boldsymbol{\theta}) \leq \tilde{g}(\boldsymbol{\theta}; \boldsymbol{\theta}^l) := \ell(\boldsymbol{\theta}^l) + \langle \nabla_{\boldsymbol{\theta}_g} \ell(\boldsymbol{\theta}^l), \boldsymbol{\theta}_g - \boldsymbol{\theta}^l_g \rangle + \frac{\hat{L}_g}{2} \|\boldsymbol{\theta}_g - \boldsymbol{\theta}^l_g\|_2^2 + \Omega(\boldsymbol{\theta}). \tag{12}$$

Note that the left hand side of (12) is the objective function of Problem (9). We obtain $\boldsymbol{\theta}^{l+1}_g$ by minimizing the upper bound on our objective, $\tilde{g}(\boldsymbol{\theta}; \boldsymbol{\theta}^l)$, with respect to $\boldsymbol{\theta}_g$:

$$\boldsymbol{\theta}^{l+1}_g \in \arg\min_{\boldsymbol{\theta}_g} \tilde{g}(\boldsymbol{\theta}; \boldsymbol{\theta}^l) = \arg\min_{\boldsymbol{\theta}_g} \frac{\hat{L}_g}{2} \left\| \boldsymbol{\theta}_g - \left(\boldsymbol{\theta}^l_g - \frac{1}{\hat{L}_g} \nabla_{\boldsymbol{\theta}_g} \ell(\boldsymbol{\theta}^l) \right) \right\|_2^2 + \Omega(\boldsymbol{\theta}_g). \tag{13}$$

Although nonconvex, the minimization problem in (13) admits a closed-form solution, which can be obtained via the operator $H : \mathbb{R}^u \to \mathbb{R}^u$ defined as follows:

$$H(\mathbf{z}; \boldsymbol{\lambda}; \hat{L}_g) = \begin{cases} \frac{\mathbf{z}}{\|\mathbf{z}\|_2} \left[\|\mathbf{z}\|_2 - \frac{\lambda_1}{\hat{L}_g} \right] & \text{if } \|\mathbf{z}\|_2 > \sqrt{\frac{2\lambda_0}{\hat{L}_g}} + \frac{\lambda_1}{\hat{L}_g} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where $\boldsymbol{\lambda} = (\lambda_0, \lambda_1)$. It can be readily seen that an optimal solution of (13) is given by $H(\mathbf{z}; \boldsymbol{\lambda}; \hat{L}_g)$, where $\mathbf{z} = \boldsymbol{\theta}^l_g - \frac{1}{\hat{L}_g} \nabla_{\boldsymbol{\theta}_g} \ell(\boldsymbol{\theta}^l)$. Below we summarize our proposed cyclic BCD algorithm.

Algorithm 1: Cyclic Block Coordinate Descent (BCD)

- **Input:** Initialization $\boldsymbol{\theta}^0$ and $\hat{L}_g$ for every $g \in [q]$.
- **Repeat** Steps 1, 2 for $l = 0, 1, 2, \ldots$ until convergence:
  1. $g \leftarrow 1 + (l \mod q)$ and $\boldsymbol{\theta}_j^{l+1} \leftarrow \boldsymbol{\theta}_j^l$ for all $j \neq g$
  2. $\boldsymbol{\theta}_g^{l+1} \leftarrow H(\mathbf{z}; \boldsymbol{\lambda}; \hat{L}_g)$, where $\mathbf{z} = \boldsymbol{\theta}_g^l - (1/\hat{L}_g)\nabla_{\boldsymbol{\theta}_g} \ell(\boldsymbol{\theta}^l)$.

Convergence Analysis. To establish convergence of the sequence $\boldsymbol{\theta}^l$ in Algorithm 1, we make use of the following assumption.

Assumption 1. *At least one of the following conditions holds:*

(a) Strong Convexity: $\mathbf{W} \succ 0$.

(b) Restricted Strong Convexity: Let $\hat{\boldsymbol{\theta}}$ *be a (Group Lasso) solution defined as* $\hat{\boldsymbol{\theta}} \in \arg\min_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) + \lambda_1 \sum_{g=1}^q \|\boldsymbol{\theta}_g\|_2$. *Let* $k = \max_{\boldsymbol{\theta}}\{\|\boldsymbol{\theta}\|_0 \mid G(\boldsymbol{\theta}) \leq G(\hat{\boldsymbol{\theta}})\}$. *Every collection of* k *columns in* $\mathbf{W}$ *are linearly independent, and the initial solution* $\boldsymbol{\theta}^0$ *(in Algorithm 1) satisfies* $h(\boldsymbol{\theta}^0) \leq h(\hat{\boldsymbol{\theta}})$.

Assumption 1(a) holds if a ridge regularization term is used, i.e., it holds for Problem (2) with $\lambda_2 > 0$. Assumption 1(b) is less restrictive because we can have $\mathbf{W} \succeq \mathbf{0}$. Suppose that for some non-negative integer u, every set of u columns in $\mathbf{W}$ are linearly independent. Then, in the Group Lasso problem (defined in Assumption 1(b)), λ_1 can be chosen sufficiently large so that some Group Lasso solution $\hat{\boldsymbol{\theta}}$ satisfies $k \leq u$. If $\hat{\boldsymbol{\theta}}$ is used to initialize Algorithm 1, then Assumption 1(b) is satisfied.

The following theorem establishes a linear convergence guarantee for the sequence generated by Algorithm 1.

Theorem 1. *Let* $\{\boldsymbol{\theta}^l\}$ *be the sequence generated by Algorithm 1 and suppose that Assumption 1 holds. Then,*

1. *The group support stabilizes after a finite number of iterations, i.e., there exists an integer* K *and a support* $S \subseteq [q]$ *such that* $Supp(\boldsymbol{\theta}^l) = S$ *for all* $l \geq K$.
2. *The sequence* $\{\boldsymbol{\theta}^l\}$ *converges to a solution* $\boldsymbol{\theta}^*$, *with* $Supp(\boldsymbol{\theta}^*) = S$, *satisfying:*

$$\boldsymbol{\theta}_S^* \in \arg\min_{\boldsymbol{\theta}_S} \quad \ell(\boldsymbol{\theta}_S) + \lambda_1 \sum_{g \in S} \|\boldsymbol{\theta}_g\|_2 \tag{15}$$

$$\|\boldsymbol{\theta}_g^*\|_2 \geq \sqrt{\frac{2\lambda_0}{\hat{L}_g}}, \quad \forall g \in S \tag{16}$$

$$\|\nabla_{\boldsymbol{\theta}_g} \ell(\boldsymbol{\theta}^*)\|_2 \leq \sqrt{2\lambda_0 \hat{L}_g} + \lambda_1, \quad \forall g \in S^c. \tag{17}$$

3. *The function* $\boldsymbol{\theta}_S \mapsto \ell(\boldsymbol{\theta}_S)$ *is strongly convex with a strong convexity parameter* $\sigma_S > 0$. *Let* L_S *be the Lipschitz constant of* $\nabla_{\boldsymbol{\theta}_S} \ell(\boldsymbol{\theta}_S)$. *Define* $\hat{L}_{max} = \max_{g \in S} \hat{L}_g + 2\lambda_1$ *and* $\hat{L}_{min} = \min_{g \in S} \hat{L}_g + 2\lambda_1$. *Then, for* $l \geq K$, *the following holds:*

$$h(\boldsymbol{\theta}^{(l+1)q}) - h(\boldsymbol{\theta}^*) \leq \left(1 - \frac{\sigma_S}{\eta}\right)\Big(h(\boldsymbol{\theta}^{lq}) - h(\boldsymbol{\theta}^*)\Big), \tag{18}$$

where $\eta = 2\hat{L}_{max}(1 + |S|(L_S + 2\lambda_1|S|)^2\hat{L}_{min}^{-2})$.

The proof of Theorem 1 is in the supplement. We present here a high-level sketch of the proof. We establish part 1 by proving a sufficient decrease condition. For part 2, we show that the objective function restricted to the group support S is strongly convex, and thus convergence follows from standard results on cyclic BCD, e.g., [6]. To establish the linear rate of convergence in part 3 of the theorem, we extend the result of [4] who show that cyclic BCD can achieve a linear rate of convergence on smooth and strongly convex functions: note that our objective function after support stabilization is *not* smooth due to the presence of the term $\sum_{g\in S}\|\boldsymbol{\theta}_g\|_2$.

Optimality conditions of BCD and PGD. The conditions in Theorem 1 (part 2) characterize a fixed point of Algorithm 1. These are necessary optimality conditions for Problem (9) since any global minimizer must be a fixed point for Algorithm 1. In what follows, we will show that the necessary optimality conditions imposed by PGD (which is a generalization of [12] to the group setting) are generally less restrictive compared to those imposed by Algorithm 1. Note that PGD is an iterative algorithm whose updates for Problem (9) are given by:

$$\boldsymbol{\theta}^{l+1} \in \arg\min_{\boldsymbol{\theta}} \left\{ \frac{1}{2\tau}\|\boldsymbol{\theta} - (\boldsymbol{\theta}^l - \tau\nabla\ell(\boldsymbol{\theta}^l))\|_2^2 + \Omega(\boldsymbol{\theta}) \right\}, \tag{19}$$

where $\tau > 0$ is a step size. Let L be the Lipschitz constant of $\nabla\ell(\boldsymbol{\theta})$. For a constant step size, the update in (19) converges if $\tau = 1/\hat{L}$ where $\hat{L}$ is a constant chosen such that $\hat{L} > L$ [see, for example, 24, 35]. For the choice $\tau = 1/\hat{L}$, it can be readily checked that any fixed point of PGD satisfies the three optimality conditions in Theorem 1 (part 2), but with $\hat{L}_g$ replaced by $\hat{L}$. The group-wise Lipschitz constant L_g satisfies $L_g \leq L$ (for any g). In many high-dimensional problems, we can have $L_g \ll L$ [see 3, 24]. Thus, Algorithm 1 generally imposes more restrictive necessary optimality conditions compared to PGD, which can lead to higher quality local minima in practice. This establishes a part of the hierarchy in (10).

3.2 Local Combinatorial Search

In this section, we introduce a local combinatorial search algorithm to improve the quality of solutions obtained by cyclic BCD (Algorithm 1). The algorithm performs the following two steps in the t-th iteration:

1. **Block Coordinate Descent:** We run Algorithm 1 initialized at the current solution $\boldsymbol{\theta}^{t-1}$ to obtain a solution $\boldsymbol{\theta}^t$. We denote the indices of the nonzero groups in $\boldsymbol{\theta}^t$ by $\text{Supp}(\boldsymbol{\theta}^t) = S$.
2. **Group Combinatorial Search:** We attempt to improve the solution $\boldsymbol{\theta}^t$ by *swapping* groups of variables from inside and outside the support S. In particular, we search for two subsets $S_1 \subseteq S$ and $S_2 \subseteq S^c$ such that removing S_1 from the support, adding S_2 to the support, and then optimizing over the groups in S_2, improves the current objective. To ensure that the local search problem is computationally feasible, we restrict our search to subsets satisfying $|S_1| \leq m$ and $|S_2| \leq m$, where m is a pre-specified integer that takes relatively small values (for example, in the range 1 to 10).

We present a formal description of the optimization problem in step 2 (above). We denote the standard basis of $\mathbb{R}^p$ by $\{\mathbf{e}_1, \ldots, \mathbf{e}_p\}$. Given a set $J \subseteq [q]$, we define the $p \times p$ matrix $\mathbf{U}^J$ as follows: the i-th column of $\mathbf{U}^J$ is $\mathbf{e}_i$ if $i \in \cup_{g\in J}\mathcal{G}_g$ and $\mathbf{0}$ otherwise. In other words, for any $\boldsymbol{\theta} \in \mathbb{R}^p$, we have $(\mathbf{U}^J\boldsymbol{\theta})_i = \theta_i$ if $i \in \cup_{g\in J}\mathcal{G}_g$ and 0 otherwise. The optimization problem in Step 2

is given by:

$$\min_{S_1,S_2,\boldsymbol{\theta}} \quad h(\boldsymbol{\theta}^t - \mathbf{U}^{S_1}\boldsymbol{\theta}^t + \mathbf{U}^{S_2}\boldsymbol{\theta}) \qquad \text{s.t.} \qquad S_1 \subseteq S, S_2 \subseteq S^c, |S_1| \leq m, |S_2| \leq m, \tag{20}$$

where we recall that $S = \text{Supp}(\boldsymbol{\theta}^t)$. If there is a feasible solution $\hat{\boldsymbol{\theta}}$ to (20) satisfying $h(\hat{\boldsymbol{\theta}}) < h(\boldsymbol{\theta}^t)$, then we move to the improved solution $\hat{\boldsymbol{\theta}}$; otherwise, we terminate the algorithm. We summarize the algorithm below:

Algorithm 2: Local Combinatorial Search

- **Input:** Initial solution $\boldsymbol{\theta}^0$ and swap subset size m.
- **Repeat** Steps 1–3 for $t = 1, 2, \ldots$ until convergence:
  1. Run Algorithm 1 initialized from $\boldsymbol{\theta}^{t-1}$ to obtain a solution $\boldsymbol{\theta}^t$.
  2. Search for a feasible solution $\hat{\boldsymbol{\theta}}$ to (20) satisfying $h(\hat{\boldsymbol{\theta}}) < h(\boldsymbol{\theta}^t)$.
  3. If step 2 succeeds, $\boldsymbol{\theta}^t \leftarrow \hat{\boldsymbol{\theta}}$. Otherwise, terminate.

Theorem 2 establishes that Algorithm 2 converges in a finite number of iterations and characterizes the corresponding solution.

Theorem 2. *Let $\{\boldsymbol{\theta}^t\}$ be the sequence of iterates generated by Algorithm 2 and suppose Assumption 1 holds. Then, $\boldsymbol{\theta}^t$ converges in a finite number of iterations to a solution that we denote by $\boldsymbol{\theta}^\dagger$. Let $S = Supp(\boldsymbol{\theta}^\dagger)$. Then, $\boldsymbol{\theta}^\dagger$ satisfies the necessary optimality conditions in part 2 of Theorem 1. In addition, $\boldsymbol{\theta}^\dagger$ satisfies:*

$$h(\boldsymbol{\theta}^\dagger) \leq \min_{S_1,S_2,\boldsymbol{\theta}} \quad h(\boldsymbol{\theta}^\dagger - \mathbf{U}^{S_1}\boldsymbol{\theta}^\dagger + \mathbf{U}^{S_2}\boldsymbol{\theta}) \qquad s.t. \qquad S_1 \subseteq S, S_2 \subseteq S^c, |S_1| \leq m, |S_2| \leq m. \tag{21}$$

Theorem 2 shows that the solutions obtained by Algorithm 2 impose more restrictive necessary optimality conditions (in particular, condition (21)) compared to Algorithm 1, which justifies part of the hierarchy in (10). This is expected, as every iteration of Algorithm 2 improves over a solution obtained by Algorithm 1. The quality of solutions returned by Algorithm 2 depends on the swap subset size m. For a sufficiently large choice of m, the algorithm will return a global minimizer. Intuitively, the computational cost of the local search in step 2 of Algorithm 2 increases with m. In our experiments, we observe that small choices such as $m = 1$ can lead to significant improvements in solution quality compared to algorithms that do not incorporate combinatorial optimization. These improvements are most pronounced in settings where $n \ll p$ or the predictors across groups are highly correlated. In Section 4.1.2, we present a MIP formulation for the local search problem in Algorithm 2 for $m > 1$. For the special case of $m = 1$, we use our own custom implementation that is more efficient than using a MIP-based approach.

3.3 Algorithms for the cardinality constrained formulation

Algorithms 1 and 2 provide solutions for the (penalized) formulation in (9). While this leads to a family of high-quality estimators across a range of model sizes, it does not allow for explicit control

over the number of nonzero groups $G(\boldsymbol{\theta})$. To this end, we consider the cardinality constrained variant of problem (9):

$$\min_{\boldsymbol{\theta}} \quad E(\boldsymbol{\theta}) := \ell(\boldsymbol{\theta}) + \lambda_1 \sum_{g\in[q]} \|\boldsymbol{\theta}_g\|_2 \quad \text{s.t. } G(\boldsymbol{\theta}) \leq k. \tag{22}$$

In order to obtain a solution to (22) with a desired support size, we propose the following procedure. First, we run Algorithm 2 (say) over a grid of λ_0-values to obtain a sequence of solutions. Then, if a desired support size, say k, is missing, we obtain it by applying proximal gradient descent (PGD) to Problem (22):

$$\boldsymbol{\theta}^{l+1} \in \underset{\boldsymbol{\theta}:\ G(\boldsymbol{\theta})\leq k}{\arg\min} \left\{ \frac{1}{2\tau}\|\boldsymbol{\theta} - (\boldsymbol{\theta}^l - \tau\nabla\ell(\boldsymbol{\theta}^l))\|_2^2 + \lambda_1 \sum_{g\in[q]} \|\boldsymbol{\theta}_g\|_2 \right\}, \tag{23}$$

where $\tau > 0$ is a step size and the initial solution $\boldsymbol{\theta}^0$ can be obtained from Algorithm 2 (for example, we take a solution with group support size closest to k).

The next proposition establishes the convergence of update (23) and describes its fixed points.

Proposition 1. *Let $\{\boldsymbol{\theta}^l\}$ be the sequence of iterates generated the PGD updates* (23)*. Let L be the Lipschitz constant of $\nabla\ell(\boldsymbol{\theta})$ and a scalar $\hat{L}$ such that $\hat{L} > L$. Then, $\{\boldsymbol{\theta}^l\}$ converges for a step size $\tau = 1/\hat{L}$. Moreover, a solution $\boldsymbol{\theta}^*$ with group support S is a fixed point of* (23) *iff $G(\boldsymbol{\theta}^*) \leq k$, and*

$$\boldsymbol{\theta}^*_S \in \underset{\boldsymbol{\theta}_S}{\arg\min}\ E(\boldsymbol{\theta}_S) \quad \text{and} \quad \|\nabla_{\boldsymbol{\theta}_g}\ell(\boldsymbol{\theta}^*)\|_2 \leq \gamma_{(k)} \quad \text{for } g \in S^c,$$

where $\gamma_g = \|\hat{L}\boldsymbol{\theta}^*_g - \nabla_{\boldsymbol{\theta}_g}\ell(\boldsymbol{\theta}^*)\|_2$, and $\gamma_{(k)}$ denotes the kth largest value in the sequence $\{\gamma_g\}_{g=1}^q$.

We omit the proof of Proposition 1 as it can be established by a simple extension to the standard results on the convergence of IHT [for example, those in 12, 3].

4 Mixed Integer Programming

In this section, we propose MIP formulations and algorithms to solve (9) and the combinatorial search problem in Algorithm 2. Section 4.1 introduces MIP formulations, and Section 4.2 presents a new BnB algorithm for solving the corresponding problems to optimality.

4.1 MIP Formulations

4.1.1 Formulations for Problem (9)

Below we present two MIP-formulations for (9).

Big-M Formulation: We first present a Big-M based MIP formulation for Problem (9):

$$\min_{\boldsymbol{\theta},\mathbf{z}} \quad \ell(\boldsymbol{\theta}) + \lambda_0 \sum_{g=1}^{q} z_g + \lambda_1 \sum_{g=1}^{q} \|\boldsymbol{\theta}_g\|_2 \tag{24a}$$

$$\text{s.t.} \quad \|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}} z_g, \quad g \in [q] \tag{24b}$$

$$z_g \in \{0,1\}, \quad g \in [q] \tag{24c}$$

where, the optimization variables are $\boldsymbol{\theta}$ (continuous) and $\mathbf{z}$ (binary). Above, $\mathcal{M}_{\text{U}}$ is an a-priori specified constant (leading to the name "Big-M") such that some optimal solution, say $\boldsymbol{\theta}^*$, to (9) satisfies $\max_{g\in[q]} \|\boldsymbol{\theta}_g^*\|_2 \leq \mathcal{M}_{\text{U}}$. In (24), the binary variable z_g controls whether all the regression coefficients in group g are zero or not: $z_g = 0$ implies that $\boldsymbol{\theta}_g = \mathbf{0}$, and $z_g = 1$ implies that $\|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}}$. Such Big-M formulations are commonly used in mixed integer programming to model relations between discrete and continuous variables, and have been recently used in ℓ_0-regularized regression [8, 58] (for example). Various techniques have been proposed to estimate the constant $\mathcal{M}_{\text{U}}$ in practice; see [8] for a discussion on estimating the Big-M in the context of linear regression. The constraints in (24b) are second order cones [13]. Moreover, the objective function in (24) can be written as a linear function, with additional second order cone constraints to express the quadratic function $\ell(\boldsymbol{\theta})$ and the terms $\|\boldsymbol{\theta}_g\|_2,\ g \in [q]$. Thus, Problem (24) can be reformulated as a Mixed Integer Second Order Cone Program (MISOCP), which can be modeled and solved (for small/moderate problem instances) with commercial MIP solvers such as Gurobi, CPLEX, and MOSEK. We present an efficient, standalone BnB algorithm for (24) in Section 4.2.

Perspective reformulation: Recall that Problem (9) contains a ridge term in its objective. The ridge term can be used to derive stronger MIP formulations for (9) based on the perspective formulation [19, 21]. As we discuss below, the perspective-based formulation differs from the Big-M formulation (24)—when $\lambda_2 > 0$, it usually leads to tighter convex relaxations and consequently, reduced MIP runtimes. First, we rewrite (9) as

$$\min_{\boldsymbol{\theta},\mathbf{z}} \quad \tilde{\ell}(\boldsymbol{\theta}) + \lambda_0 \sum_{g=1}^{q} z_g + \lambda_1 \sum_{g=1}^{q} \|\boldsymbol{\theta}_g\|_2 + \lambda_2 \sum_{g=1}^{q} \|\boldsymbol{\theta}_g\|_2^2 \qquad \text{s.t.} \quad \text{(24b)}, \text{(24c)} \tag{25}$$

where $\ell(\boldsymbol{\theta}) = \tilde{\ell}(\boldsymbol{\theta}) + \lambda_2\|\boldsymbol{\theta}\|_2^2$. Using the perspective reformulation [19, 21, 18] for the ridge term $\sum_{g\in[q]} \|\boldsymbol{\theta}_g\|_2^2$ in the objective, we can reformulate (25) as

$$\min_{\boldsymbol{\theta},\mathbf{z},\boldsymbol{s}} \quad \tilde{\ell}(\boldsymbol{\theta}) + \lambda_0 \sum_{g=1}^{q} z_g + \lambda_1 \sum_{g=1}^{q} \|\boldsymbol{\theta}_g\|_2 + \lambda_2 \sum_{g=1}^{q} s_g, \tag{26a}$$

$$\text{s.t.} \quad \|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}} z_g,\ g \in [q] \tag{26b}$$

$$\|\boldsymbol{\theta}_g\|_2^2 \leq s_g z_g, \quad g \in [q] \tag{26c}$$

$$z_g \in \{0,1\}, s_g \geq 0, \quad g \in [q]. \tag{26d}$$

Compared to (25), formulation (26) uses additional auxiliary variables $s_g \in \mathbb{R}_{\geq 0},\ g \in [q]$ and rotated second order cone constraints: $\|\boldsymbol{\theta}_g\|_2^2 \leq s_g z_g$ for $g \in [q]$. Each s_g takes the place of the term $\|\boldsymbol{\theta}_g\|_2^2$ in the objective function in (24). Specifically, any optimal solution $(\boldsymbol{\theta}^*, \mathbf{z}^*, \boldsymbol{s}^*)$ to (26) must satisfy $s_g^* = \|\boldsymbol{\theta}_g^*\|_2^2$.

Although the MIP formulations (26) and (25) are equivalent, their continuous relaxations are generally different. The following proposition states that the relaxation of (26) is generally tighter (i.e., has a higher objective) than the relaxation of (25).

Proposition 2. *Let v_1 and v_2 be the objective values of* (25) *and* (26) *upon relaxing the binary variable z_g to $[0,1]$ for all $g \in [q]$. Let $(\boldsymbol{\theta}^*, \mathbf{z}^*, \mathbf{s}^*)$ be an optimal solution to the relaxation corresponding to v_2. Then, the following holds:*

$$v_2 - v_1 \geq \lambda_2 \sum_{g\in[q]|z_g^*>0} \|\boldsymbol{\theta}_g^*\|_2^2 \Big((z_g^*)^{-1} - 1 \Big).$$

Proposition 2 implies that using formulation (26) (over formulation (24)) can lead to tighter lower bounds for the root node relaxation; and hence tighter dual bounds for the node relaxations in the BnB tree. This can result in improved runtimes in the overall BnB solver (as we demonstrate in our experiments). Thus, in our algorithmic framework in Section 4.2, we focus on formulation (26). To be clear, our BnB procedure applies even without the presence of a ridge term (i.e., $\lambda_2 = 0$). Specifically, if $\lambda_2 = 0$ in (26), the conic constraints (26c) can be removed and formulation (26) reduces to the Big-M formulation in (24).

4.1.2 MIP formulation for local combinatorial search

We present a MIP formulation for the local search problem[4] that arises in Algorithm 2. Problem (20) can be formulated using the following Big-M based MIP:

$$\begin{aligned}
\min_{\mathbf{u},\mathbf{z},\boldsymbol{\theta}} \quad & \ell(\mathbf{u}) + \lambda_0 \sum_{g=1}^{q} z_g + \lambda_1 \sum_{g=1}^{q} \|\mathbf{u}_g\|_2 \\
\text{s.t.} \quad & \mathbf{u} = \boldsymbol{\theta}^t - \sum_{g\in S} \mathbf{U}^g \boldsymbol{\theta}^t (1 - z_g) + \sum_{g\in S^c} \mathbf{U}^g \boldsymbol{\theta} && \text{(27a)} \\
& \|\mathbf{u}_g\|_2 \leq \mathcal{M}_{\mathrm{U}} z_g, \ g \in S^c && \text{(27b)} \\
& \sum_{g\in S} z_g \geq |S| - m, \ \sum_{g\in S^c} z_g \leq m && \text{(27c)} \\
& z_g \in \{0,1\}, \quad g \in [q]. && \text{(27d)}
\end{aligned}$$

In the formulation above, we assume that $\mathcal{M}_{\mathrm{U}}$ is chosen sufficiently large so that some optimal solution to (20), say $\boldsymbol{\theta}^*$, satisfies $\|\boldsymbol{\theta}^*_g\|_2 \leq \mathcal{M}_{\mathrm{U}}, \ g \in S^c$. As we discuss below, the objective in (27) represents $h(\mathbf{u})$ with $\mathbf{u} = \boldsymbol{\theta}^t - \mathbf{U}^{S_1}\boldsymbol{\theta}^t + \mathbf{U}^{S_2}\boldsymbol{\theta}$, where $h(\mathbf{u})$, S_1 and S_2 are as defined in (20). Note that the variable $\mathbf{u}$ is an auxiliary variable introduced to simplify the presentation. The binary variables $z_g, g \in [q]$ are used to select the subsets $S_1 \subseteq S$ and $S_2 \subseteq S^c$. In particular, for $g \in S$, $z_g = 0$ iff $g \in S_1$, and this is encoded by constraint (27a). On the other hand, for $g \in S^c$, $z_g = 1$ iff $g \in S_2$, and this is encoded by constraints (27a) and (27b). Therefore, $\sum_{g=1}^{q} z_g$ is equal to $G(\mathbf{u})$. The constraints (27c) enforce $|S_1| \leq m$ and $|S_2| \leq m$.

The local search MIP-formulation (27) has a *smaller* search space compared to the full problem (24). This is due to the additional constraints appearing in (27c). Furthermore, Problem (27) effectively uses $|S^c|$-many 'free' continuous group-variables—this is in contrast to $|S| + |S^c|$ continuous group-variables appearing in the full problem. Thus, for small values of m, Problem (27) can be typically solved faster than the MIP formulation of (8). While (27) is based on a Big-M formulation, in the presence of an additional ridge regularizer, one can also derive a perspective reformulation using ideas similar to (26).

4.2 Exact optimization via a custom nonlinear Branch-and-Bound algorithm

High-performance commercial MIP solvers, such as Gurobi and CPLEX, often deliver state-of-the-art performance for a variety of MIP problems. These solvers are based on a BnB framework,

[4]We recommend the use of the MIP formulations when $m \geq 2$. When $m = 1$ a solution to the local search procedure can be computed efficiently from first principles.

which can solve MIP problems to global optimality, typically without having to explicitly enumerate all (exponentially many) solutions in the search space. These solvers are general-purpose and do not take into account the specific structure of the problems we consider here. Therefore, their performance can suffer: we have empirically observed that they may require several hours to solve (to certifiable optimality) instances of (26) with $p \sim 10^3$, and larger problems can take much longer.

To address this lack of scalability in general-purpose MIP solvers, we propose a specialized, nonlinear BnB framework for solving (26) to certifiable optimality. Our framework takes into account problem structure to achieve scalability. As we demonstrate in the experiments section, our BnB can solve instances with $p \sim 5 \times 10^6$ to certifiable optimality in minutes to hours, whereas Gurobi takes prohibitively long (at least a day) for $p \sim 10^3$. An important feature of our proposal is an open-source, standalone implementation of the BnB solver, which does not rely on sophisticated and proprietary BnB-capabilities of commercial MIP solvers (e.g., Gurobi). We first give a high-level overview of our novel nonlinear BnB framework and then dive into specific technical details.

Overview of nonlinear BnB: Nonlinear BnB is a general framework for solving mixed integer nonlinear programs [5]. This framework constructs a search tree to partition the set of feasible solutions of the given MIP (Problem (26) in our case). Instead of explicitly enumerating all the (exponentially many) feasible solutions, BnB uses intelligent enumeration and methods to prune parts of the tree by using lower bounds (dual bounds) on the optimal objective value. In what follows, we briefly describe how the tree is constructed and pruned. Starting at the root node, the algorithm solves a nonlinear convex relaxation of Problem (26), where all binary variables are relaxed to $[0, 1]$ – this is usually referred to as the *root relaxation*. Then, the algorithm chooses a *branching variable*, say z_g, and creates two child nodes (optimization subproblems): one with $z_g = 0$ and another with $z_g = 1$, where all other binary variables are relaxed to $[0, 1]$. The algorithm then proceeds recursively: for every unvisited node, it solves the corresponding optimization problem and checks if there is any fractional (i.e., non-binary) variable z_g. If there is any fractional z_g, the branching process must continue — to this end, the algorithm branches on one fractional z_g, generating two new child nodes. Thus, every node in the search tree corresponds to an optimization subproblem and every edge represents a branching decision.

While growing the search tree, BnB maintains an upper bound on the objective function (which can be obtained from any feasible solution to the problem). If the optimization subproblem at the current node leads to an objective value that exceeds the upper bound, then the node is pruned (i.e., no children are generated for this node), because none of its descendants can have a better objective value than the upper bound. Another case where BnB can safely prune a node is when the corresponding subproblem leads to an integral solution, i.e., a binary $\mathbf{z}$ (since there will be no variables to branch on). For further discussion on nonlinear BnB, see [5].

Specific details: There are many delicate details in BnB that can critically affect its scalability: for example, the choice of the algorithm for solving the continuous node subproblems, obtaining upper bounds, branching, and tree-search strategies. We discuss our choices below:

- **Subproblem solver:** The optimal solutions of the continuous optimization subproblems encountered in the course of BnB are typically sparse (see Section 4.2.1 for further discussions). To solve these subproblems, we propose an active-set algorithm, which exploits sparsity by considering a reduced problem restricted to a small subset of groups. Moreover,

we share information on the active sets across the BnB tree to speed up convergence (see Section 4.2.2).

- **Upper bounds:** Better upper bounds can lead to aggressive pruning in the search tree, which can reduce the overall runtime. We obtain the initial upper bound using the approximate algorithms of Section 3. As we demonstrate in the experiments, our approximate algorithms typically obtain optimal or near-optimal solutions, making them a good choice to initialize BnB. Moreover, at every node of BnB, we attempt to improve the upper bound by using the sparsity pattern of the solution to the current node's subproblem. More concretely, let $S \subseteq q$ denote the group support of the latter subproblem's solution. Then, we obtain a new upper bound, by restricting optimization to S, i.e., we solve:

$$\min_{\boldsymbol{\theta}} \quad \tilde{\ell}(\boldsymbol{\theta}) + \lambda_1 \sum_{g=1}^{q} \|\boldsymbol{\theta}\|_2 + \lambda_2 \|\boldsymbol{\theta}\|_2^2 \quad \text{s.t.} \quad \boldsymbol{\theta}_{S^c} = \mathbf{0},\ \|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\mathrm{U}},\ g \in [q].$$

- **Branching and search strategies:** The branching strategy selects the next variable to branch on, while the search strategy decides which unexplored node in the search tree to visit next. Many elaborate strategies for branching and search have been proposed in the literature – see [41] for a survey. When the initial upper bound is of high quality, more aggressive pruning is possible, and simple strategies tend to work relatively well in practice [for example, see the discussion in 17]. Since our approximate algorithms typically return good upper bounds, we rely on simple strategies. For branching, we use maximum fractional branching [5, 41], which branches on the factional variable z_g whose value is closest to 0.5. For search, we use breadth-first search and switch to depth-first search if memory issues are encountered.

Our approach extends our recent work [26] for the best subset selection problem (with a group size of one). We note that there are important differences as the Group ℓ_0 problem involves a different and more challenging optimization formulation. Specifically, the Big-M constraints in (26b) translate to second order cones, instead of box-constraints that appear when the group sizes are one. Furthermore, in the group setup, we have a non-smooth term $\sum_{g\in[q]} \|\boldsymbol{\theta}_g\|_2$ in the objective of (26). The conic constraints and ℓ_2 norms in our problem require special care when developing the subproblem solver (for example, when reformulating the subproblems in Section 4.2.1 and designing the active set algorithm in Section 4.2.2). It is also worth mentioning that in the simplest case where $\lambda_1 = \lambda_2 = 0$, our solver solves a MISOCP, whereas [26] solves a mixed integer quadratic program.

4.2.1 Relaxation reformulation

In this section, we study the convex relaxation arising at a node of the BnB search tree. We present a particular reformulation of this problem that leads to (i) useful insights about the sparsity in the solutions of the convex relaxation; and (ii) computational benefits. To simplify the presentation, we will first focus on the root relaxation of (26), which is obtained by relaxing all the binary variables in (26) to $[0, 1]$.

Note that the root relaxation involves the variables $(\boldsymbol{\beta}, \mathbf{z}, \mathbf{s})$. In Proposition 3, we show that the root relaxation can be reformulated in the $\boldsymbol{\beta}$ space, leading to a regularized least squares

problem. The associated regularizer can be characterized in terms of the reverse Huber penalty [46] (see also [18]), which is a function $\mathcal{H} : \mathbb{R} \to \mathbb{R}$ defined as follows:

$$\mathcal{H}(t) = \begin{cases} |t| & \text{if } |t| \leq 1 \\ (t^2+1)/2 & \text{otherwise.} \end{cases} \tag{28}$$

Proposition 3. *The root relaxation obtained by relaxing the binary variables in* (26) *to* $[0,1]$ *is equivalent to:*

$$\min_{\boldsymbol{\theta}} \quad F(\boldsymbol{\theta}) := \tilde{\ell}(\boldsymbol{\theta}) + \sum_{g=1}^{q} \Psi(\boldsymbol{\theta}_g; \boldsymbol{\lambda}, \mathcal{M}_{\text{U}}) \quad \textit{s.t.} \quad \|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}}, \ g \in [q]. \tag{29}$$

where $\boldsymbol{\lambda} = (\lambda_0, \lambda_1, \lambda_2)$ *and*

$$\Psi(\boldsymbol{\theta}_g; \boldsymbol{\lambda}, \mathcal{M}_{\text{U}}) := \begin{cases} 2\lambda_0 \mathcal{H}(\sqrt{\lambda_2/\lambda_0}\|\boldsymbol{\theta}_g\|_2) + \lambda_1\|\boldsymbol{\theta}_g\|_2 & \textit{if } \sqrt{\lambda_0/\lambda_2} \leq \mathcal{M}_{\text{U}} \\ (\lambda_0/\mathcal{M}_{\text{U}} + \lambda_1 + \lambda_2\mathcal{M}_{\text{U}})\|\boldsymbol{\theta}_g\|_2 & \textit{if } \sqrt{\lambda_0/\lambda_2} > \mathcal{M}_{\text{U}}. \end{cases}$$

The reformulation in (29) eliminates the the conic and Big-M constraints from the root relaxation, at the expense of introducing the non-smooth penalty $\sum_{g=1}^{q} \Psi(\boldsymbol{\theta}_g; \boldsymbol{\lambda}, \mathcal{M}_{\text{U}})$ which is separable across the blocks $\{\boldsymbol{\theta}_g\}_1^q$. Depending on the choices of $\boldsymbol{\lambda}$ and $\mathcal{M}_{\text{U}}$, the penalty Ψ is either the ℓ_2 norm or a combination of the reverse Huber penalty and the ℓ_2 norm. In either case, the penalty is sparsity-inducing. In essence, Problem (29) is similar to the Group Lasso problem [59], with two exceptions: (i) Problem (29) has the additional constraints: $\|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}}, \ g \in [q]$, and (ii) when $\sqrt{\lambda_0/\lambda_2} \leq \mathcal{M}_{\text{U}}$, the penalty involves the reverse Huber penalty.

Node relaxations within the BnB tree: The convex relaxation subproblem encountered at a node of the BnB search tree is similar to the root relaxation, except that some of the z_gs are fixed to 0 or 1. The fixed z_gs are determined by the branching decisions made starting from the root until reaching the node. The convex relaxation at a particular node can be reformulated in the $\boldsymbol{\beta}$-space similar to the reformulation of the root relaxation in (29), except that: (i) if $z_g = 0$ then the corresponding group should be removed from the objective function; and (ii) if $z_g = 1$, then the penalty $\Psi(\boldsymbol{\theta}_g; \boldsymbol{\lambda}, \mathcal{M}_{\text{U}})$ should be replaced with $\tilde{\Psi}(\boldsymbol{\theta}_g; \boldsymbol{\lambda}) := \lambda_1\|\boldsymbol{\theta}_g\|_2 + \lambda_2\|\boldsymbol{\theta}_g\|_2^2$. More precisely, let $\mathcal{Z}$ and $\mathcal{N}$ be the sets of indices of the z_gs that are fixed to 0 and 1, respectively. Then, the following subproblem is solved at the corresponding node:

$$\min_{\boldsymbol{\theta}} \quad \tilde{\ell}(\boldsymbol{\theta}) + \sum_{g \in \mathcal{N}^c} \Psi(\boldsymbol{\theta}_g; \boldsymbol{\lambda}, \mathcal{M}_{\text{U}}) + \sum_{g \in \mathcal{N}} \tilde{\Psi}(\boldsymbol{\theta}_g; \boldsymbol{\lambda}) \ \text{ s.t. } \ \boldsymbol{\theta}_{\mathcal{Z}} = \mathbf{0}, \|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}}, \ g \in [q]. \tag{30}$$

In the next section, we develop a scalable algorithm for solving Problem (29). The BnB subproblem (30) can be solved similarly after accounting for the fixed z_gs.

4.2.2 Active-Set subproblem solver

As discussed earlier, a solution to Problem (29) is expected to be sparse in $\boldsymbol{\theta}$ (this will be also true for the node sub-problems in the BnB tree). To exploit this sparsity, we use an active-set algorithm: We start by solving Problem (29) restricted to a small subset of groups (i.e., the *active set*). After convergence on the active set, we augment the active set with a collection of groups that violate the optimality conditions for the full problem (if any) and then resolve the problem

restricted to the augmented active set. The algorithm keeps iterating between solving a reduced optimization problem and augmenting the active set, until the optimality conditions for the full problem are satisfied. Such active-set algorithms have proven to be effective in scaling up the solvers for group Lasso-type problems [for example, see 25]—our usage differs in that we use this active-set strategy within every node of the BnB tree.

Next, we describe our active-set algorithm more formally. Let $\mathcal{A} \subseteq [q]$ be the active set. The algorithm starts by solving (29) restricted to the active set, i.e.,

$$\hat{\boldsymbol{\theta}} \in \arg\min_{\boldsymbol{\theta}} \quad F(\boldsymbol{\theta}) \;\; \text{s.t.} \;\; \|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}}, \;\; g \in [q], \;\; \boldsymbol{\theta}_{\mathcal{A}^c} = \mathbf{0}. \tag{31}$$

After solving (31), we check if $\hat{\boldsymbol{\theta}}$ satisfies the optimality condition for the full problem. Equivalently, for every group $g \in \mathcal{A}^c$, we check if the following holds

$$\mathbf{0} \in \arg\min_{\boldsymbol{\theta}_g} \quad F(\hat{\boldsymbol{\theta}}_1, \ldots, \boldsymbol{\theta}_g, \ldots, \hat{\boldsymbol{\theta}}_q) \;\; \text{s.t.} \;\; \|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}}. \tag{32}$$

Since $\boldsymbol{\theta}_g = \mathbf{0}$ is in the interior of the feasible set, condition (32) is equivalent to the zero-subgradient condition: $\mathbf{0} \in \partial_{\boldsymbol{\theta}_g} F(\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_{g-1}, \mathbf{0}, \hat{\boldsymbol{\theta}}_{g+1}, \ldots, \hat{\boldsymbol{\theta}}_q)$, and can be checked in closed form.

We repeat the procedure of solving the restricted subproblem in (31) and augmenting $\mathcal{A}$ with groups that violate (32), until there are no more violations. The algorithm is summarized below.

Algorithm 3: An Active-set Algorithm for (29)

- **Input:** Initial solution $\hat{\boldsymbol{\theta}}$ and initial active set $\mathcal{A}$.
- **Repeat** Steps 1—3 till convergence:
  1. Solve the restricted problem (31) to get a solution $\hat{\boldsymbol{\theta}}$.
  2. $\mathcal{V} \leftarrow \{g \in \mathcal{A}^c \mid (32) \text{ is violated}\}$.
  3. If $\mathcal{V}$ is empty **terminate**, otherwise[5], $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{V}$.

Algorithm 3 is guaranteed to converge to an optimal solution for Problem (29) in a finite number of steps, as there are finitely many groups.

Choice of the active set: The quality of the initial active set $\mathcal{A}$ can have a important effect on the number of iterations in Algorithm 3. Due to the choice of our branching rule, the parent and its two child nodes solve similar subproblems; the only difference between these subproblems is that a single z_g is fixed to 0 or 1 in the children. Thus, the solutions and supports of the parent and its children are unlikely to differ by much. We therefore initialize the active set of every node in the BnB tree (except the root) with the support of its parent. For the root node, we initialize the active set with the support of the warm start, obtained from the approximate algorithms that are discussed in Section 3.

[5]In some cases, $|\mathcal{V}|$ can be large, which can slow down the solver in Step 1. Thus, if $\mathcal{V}$ has more than K groups, we augment $\mathcal{A}$ with the K groups in $\mathcal{V}$ that have the largest violation (instead of $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{V}$). In our experiments we set $K = 10$. We found this helpful to keep the size of the active set manageable during the course of the algorithm.

Solving the restricted subproblem: The convex sub-problem (31) in Step 1 has a small active set and can be solved with a variety of optimization algorithms: for example, BCD, proximal gradient methods [6] or an interior point solver (as available in Gurobi). In our experiments, we use the latter due to its good performance in practice.

5 Statistical Theory

In this section we derive non-asymptotic prediction and estimation error bounds for the Group ℓ_0 estimators, and compare them to the bounds that have been established for the corresponding Group Lasso-based approaches. We focus on linear regression models in Section 5.1 and on nonparametric additive models in Section 5.2.

While the arguments used in our proofs extend naturally to the penalized case, we focus on the constrained specifications of the proposed estimators for concreteness. To simplify the presentation, we consider the setting where the model is correctly specified, so that the true regression function is a feasible solution to the corresponding optimization problem. However, our results can be generalized to allow for model misspecification.

We say that a constant is *universal* if it does not depend on other parameters, such as n, q or k. We use the notation $\gtrsim$ and $\lesssim$ to indicate that inequalities $\geq$ and $\leq$, respectively, hold up to positive universal multiplicative factors, and write $\asymp$ when the two inequalities hold simultaneously.

5.1 Linear Model

We assume that the observed data follows the model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\epsilon}$, where $\mathbf{X}$ is deterministic and the elements of $\boldsymbol{\epsilon}$ are independent $N(0, \sigma^2)$ with $\sigma > 0$. We define $k_* = G(\boldsymbol{\beta}^*)$ and refer to $n^{-1}\|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2$ as the prediction error for estimator $\widehat{\boldsymbol{\beta}}$. For simplicity of the presentation we focus on the setting where each group has the same number of T features. Thus, the total number of features, p, is equal to qT. Given $\boldsymbol{\beta} \in \mathbb{R}^p$ and $J \subseteq [q]$, we write $\boldsymbol{\beta}_J$ for the subvector of $\boldsymbol{\beta}$ indexed by $\cup_{g\in J}\mathcal{G}_g$. Consider the following definition, in which we use the notation $\|\boldsymbol{\beta}\|_{2,1} = \sum_{g=1}^q \|\boldsymbol{\beta}_g\|_2$.

Definition 1. *Given a positive integer k and a constant $c \geq 1$, let*

$$\gamma_k = \min_{\boldsymbol{\beta}\neq\mathbf{0},\, G(\boldsymbol{\beta})\leq k} \frac{\sqrt{k}\|\mathbf{X}\boldsymbol{\beta}\|_2}{\sqrt{n}\|\boldsymbol{\beta}\|_{2,1}} \quad \textit{and} \quad \kappa_{k,c} = \min_{J\subseteq[q],|J|\leq k} \left\{ \min_{\boldsymbol{\beta}\neq\mathbf{0},\, \|\boldsymbol{\beta}_{J^c}\|_{2,1}\leq c\|\boldsymbol{\beta}_J\|_{2,1}} \frac{\sqrt{k}\|\mathbf{X}\boldsymbol{\beta}\|_2}{\sqrt{n}\|\boldsymbol{\beta}_J\|_{2,1}} \right\}.$$

The above definition is most meaningful under the scaling of the features where $\|\mathbf{x}_j\|_2 \asymp \sqrt{n}$ for all j. As we discuss below, constants $\kappa_{k_*,c}^{-1}$, with $c > 1$, appear in the prediction and estimation error bounds for the Group Lasso estimator, while $\gamma_{2k_*}^{-1}$ appears in the estimation error bound for the Group ℓ_0 estimator. The following result establishes a useful relationship for these quantities.

Proposition 4. *$\gamma_{2k} \geq \kappa_{k,c}/\sqrt{2}$, for all positive integers k and all $c \geq 1$.*

We study estimator $\widehat{\boldsymbol{\beta}}$, which solves the following optimization problem:

$$\min_{\boldsymbol{\beta}} \quad \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{s.t.} \quad \sum_{g=1}^q \mathbf{1}(\boldsymbol{\beta}_g \neq \mathbf{0}) \leq k, \tag{33}$$

where k is a fixed parameter that controls the sparsity level. We note that (33) is a special case of the cardinality constrained problem considered in Section 3.3. Our first result provides the prediction error bound for $\widehat{\boldsymbol{\beta}}$, which holds without any assumptions on the design.

Theorem 3. *Let $\delta_0 \in (0,1)$ and suppose that $\widehat{\boldsymbol{\beta}}$ solves optimization problem (33) for $k \geq k_*$. Then,*

$$n^{-1}\|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2 \lesssim \sigma^2 k\Big[\frac{T+\log(q/k)}{n}\Big] + \sigma^2\Big[\frac{\log(1/\delta_0)}{n}\Big].$$

with probability at least $1-\delta_0$.

Letting $\delta_0 = (k/q)^k$ and using Definition 1, we derive the following result.

Corollary 1. *If $k = k_*$, then*

$$\begin{aligned} n^{-1}\|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2 &\lesssim \sigma^2 k_*\Big[\frac{T+\log(q/k_*)}{n}\Big] \\ \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_{2,1} &\lesssim \sigma k_*\Big[\frac{T+\log(q/k_*)}{n}\Big]^{1/2}\big[\gamma(2k_*)\big]^{-1} \end{aligned}$$

with probability at least $1-(k_*/q)^{k_*}$.

We make several observations regarding the established error bounds, comparing them to the bounds for the Group Lasso estimator, denoted by $\widehat{\boldsymbol{\beta}}_{\mathrm{GL}}$, which replaces the ℓ_0 constraint in Problem (33) with a penalty on $\|\boldsymbol{\beta}\|_{2,1}$. To simplify the comparison of the corresponding rates, we focus on the setting where $k = k_*$.

Remark 1. *The Group ℓ_0 prediction error rate provided in Corollary 1 matches the corresponding optimal prediction error rate established in [34]. The estimation error rate in Corollary 1 is also optimal provided that $\gamma_{2k_*}^{-1}$ is bounded by a universal constant under the aforementioned feature scaling $\|\mathbf{x}_j\|_2 \asymp \sqrt{n}$.*

Remark 2. *Let $\|\mathbf{x}_j\|_2 \asymp \sqrt{n}$ for all j and assume that $\kappa_{k_*,c}^{-1}$ is bounded by a universal constant for some $c > 1$. Then, the error bounds for the Group Lasso estimator [see, for example, Section 8.3 of 15] are*

$$n^{-1}\|\mathbf{X}\widehat{\boldsymbol{\beta}}_{GL} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2 \lesssim \sigma^2 k_*\Big[\frac{T+\log(q)}{n}\Big] \quad \textit{and} \quad \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_{2,1} \lesssim \sigma k_*\Big[\frac{T+\log(q)}{n}\Big]^{1/2}. \tag{34}$$

The Group ℓ_0 rates discussed in Remark 1 are better than those in display (34), because they replace the $\log(q)$ term with $\log(q/k_*)$. Moreover, in view of Proposition 4, the assumption on γ_{2k_*} in Remark 1 is weaker than the Group Lasso assumption on $\kappa_{k_*,c}$. Finally, the Group ℓ_0 prediction error bound holds without any assumptions on the design.

The last observation represents an important non-trivial advantage of ℓ_0-based approaches over Lasso-type methods. [63] provide examples of design matrices in the usual linear regression context for which the Lasso prediction error is lower-bounded by a constant multiple of $1/\sqrt{n}$, generally leading to a much larger prediction error than the one for the ℓ_0-based method.[6]

[6]The lower-bound applies to a wide class of coordinate-separable M-estimators, including local optima of non-convex regularizers such as SCAD and MCP.

Remark 3. *One advantage of estimator (33) is that tuning parameter k directly controls the sparsity of the proposed estimator. In particular, the $\widehat{\boldsymbol{\beta}}$ that achieves the bounds in Corollary 1 satisfies $G(\widehat{\boldsymbol{\beta}}) \leq k_*$. On the other hand, the $\widehat{\boldsymbol{\beta}}_{GL}$ that achieves bounds (34) is typically much more dense. The following inequality, which holds with high probability, is provided in [34]:*

$$G(\widehat{\boldsymbol{\beta}}_{GL}) \leq \Big[\frac{64\phi_{\max}}{\kappa_{k_*,3}}\Big]k_*.$$

Here, $\phi_{\max}$ is the maximum eigenvalue of $\mathbf{X}^\top\mathbf{X}/n$. Thus, the right-hand side is at least $64k_*$.

Remark 4. *Theorem 3 and Corollary 1 can also apply to approximate solutions, obtained after an early termination of the MIP solver. In such settings, the solver provides the current lower and upper bounds, LB and UB, on the value of the objective. If the corresponding optimality gap satisfies $(UB - LB)/LB \lesssim \sigma^2 k_*[T + \log(q/k_*)]/n$, then the bounds in Corollary 1 also hold for the approximate solution.*

An attractive feature of Theorem 3 is that the uncertainty parameter δ_0 is independent of the tuning parameter k. This allows us to control the expected prediction error, as we demonstrate in the following result.

Corollary 2. *Under the conditions of Theorem 3,*

$$\mathbb{E}\|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2 \lesssim \sigma^2 k\big[T + \log(q/k)\big].$$

An application of Definition 1 yields a corresponding bound on the expected estimation error.

5.2 Nonparametric Additive Model

We study the performance of the proposed approach in the deterministic design setting. We write $\|\cdot\|_{L_2}$ for the L_2 norm of a real-valued function on $[0,1]$. Using the notation in Section 2.2, we let $\mathcal{C}_j = \mathcal{C}$ for all j and focus on the case where $\mathcal{C}$ is an L_2-Sobolev space:

$$\mathcal{C} = \Big\{g : [0,1] \mapsto \mathbb{R},\ \|g\|_{L_2} + \|g^{(m)}\|_{L_2} < \infty\Big\} \quad \text{and} \quad \mathrm{Pen}(g) = \|g^{(m)}\|_{L_2}.$$

We define $\mathcal{C}_{\mathrm{gr}} = \{f : [0,1]^q \mapsto \mathbb{R},\ f(\mathbf{x}) = \sum_{j=1}^q f_j(x_j),\ f_j \in \mathcal{C}\}$ as the corresponding space of additive functions. We associate each $f \in \mathcal{C}_{\mathrm{gr}}$ with the vector $\mathbf{f} = \sum_{j=1}^q \mathbf{f}_j$, where $\mathbf{f}_j = \big(f_j(x_{1j}), ..., f_j(x_{nj})\big)$, and let

$$G(f) = \sum_{j=1}^q \mathbf{1}(\mathbf{f}_j \neq \mathbf{0}), \qquad \mathrm{Pen}_{\mathrm{gr}}(f) = \sum_{j=1}^q \mathrm{Pen}(f_j).$$

We focus on the estimator that solves the following optimization problem:

$$\min_{f \in \mathcal{C}_{\mathrm{gr}}} \ \|\mathbf{y} - \mathbf{f}\|_n^2 + \lambda_n \mathrm{Pen}_{\mathrm{gr}}(f) \quad \text{s.t.} \quad G(f) \leq k, \tag{35}$$

where $\|\cdot\|_n$ denotes the Euclidean norm divided by $\sqrt{n}$.[7] To ensure identifiability of the representation $f(\mathbf{x}) = \sum_{j=1}^q f_j(x_j)$, additional restrictions are typically imposed. For example, a popular

[7] We acknowledge the notational inconsistency when $n \leq 2$.

method is to separate out the constant term and require that $\sum_{i=1}^n f_j(x_{ij}) = 0$ for each j. Here we follow the approach of [53] and avoid specifying a particular set of restrictions. We treat every representation of f as equivalent, with the understanding that one particular representation is used when evaluating properties of the components, such as $\|\mathbf{f}_j\|_n$.

We are interested in comparing estimator (35), denoted by $\widehat{f}$, with the widely popular Group Lasso-based approach, which replaces the ℓ_0 constraint in Problem (35) with a penalty on $\sum_{j=1}^q \|\mathbf{f}_j\|_n$. Theoretical properties of the latter approach have been investigated extensively [see, for example, 39, 32, 50, 52, 60, 53, and the references therein]. To compare the error bounds for the two estimators, we need the following definition.

Definition 2. *Given a positive integer k, a constant $\xi \in (1, \infty]$ and an index set $J \subseteq [q]$, let*

$$A_{k,\xi} = \{f \in \mathcal{C}_{gr} : \sum_{j=1}^q \|\mathbf{f}_j\|_n \neq 0,\ G(f) \leq k,\ 2n^{-m/(2m+1)}\mathrm{Pen}_{\mathrm{gr}}(f) \leq (\xi - 1)\sum_{j=1}^q \|\mathbf{f}_j\|_n\}$$

$$B_{J,\xi} = \{f \in \mathcal{C}_{gr} : \sum_{j=1}^q \|\mathbf{f}_j\|_n \neq 0,\ \sum_{j \notin J} \|\mathbf{f}_j\|_n + n^{-m/(2m+1)}\mathrm{Pen}_{\mathrm{gr}}(f) \leq \xi \sum_{j \in J} \|\mathbf{f}_j\|_n\}$$

$$\psi(k,\xi) = \min_{f \in A_{k,\xi}} \frac{\sqrt{k}\|\mathbf{f}\|_n}{\sum_{j=1}^q \|\mathbf{f}_j\|_n} \quad \textit{and} \quad \phi(k,\xi) = \min_{J \subseteq [q], |J| \leq k} \left\{ \min_{f \in B_{J,\xi}} \frac{\sqrt{k}\|\mathbf{f}\|_n}{\sum_{j \in J} \|\mathbf{f}_j\|_n} \right\}.$$

As we discuss below, constants $\phi(2k,\xi)^{-1}$ appear in the error bounds for the Group Lasso-based approach, while constants $\psi(k,\xi)^{-1}$ appear in some of the bounds that we establish for $\widehat{f}$. The following result establishes a useful relationship for these quantities.

Proposition 5. *For all positive integers k and all $\xi \in (1, \infty]$, $\psi(2k,\xi) \geq \phi(k,\xi)/\sqrt{2}$.*

We assume that the observed data follows the model $\mathbf{y} = \mathbf{f}^* + \boldsymbol{\epsilon}$, where $f^* \in \mathcal{C}_{\mathrm{gr}}$, and the elements of $\boldsymbol{\epsilon}$ are independent $N(0, \sigma^2)$ with $\sigma > 0$. We refer to $\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2$ as the prediction error for estimator $\widehat{f}$. We write $r_n = n^{-m/(2m+1)}$, suppressing the dependence on m for notational simplicity, noting that r_n^2 is the optimal prediction error rate in the univariate regression setting where $f^* \in \mathcal{C}$. For example, in the case where $\mathcal{C}$ is the second order Sobolev space, which corresponds to $m = 2$, the above rate is $r_n^2 = n^{-4/5}$. We define $\alpha = 1/(4m+2)$ and note that $\alpha = 1/10$ when $m = 2$. The next result, in which we treat $m \geq 1$ as a fixed integer, establishes prediction error bounds for the proposed approach.

Theorem 4. *Let $k_* = G(f^*)$ and consider optimization Problem (35) with $k \geq k_*$. There exists a universal constant c_1, such that if $\lambda_n \geq c_1\sigma\big[k^{2\alpha}r_n^2 + k^\alpha r_n\sqrt{\log(eq/k)/n}\big]$, then*

$$\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2 \lesssim \sigma^2 k\Big[k^{2\alpha}r_n^2 + \frac{\log(eq/k)}{n}\Big] + \lambda_n \mathrm{Pen}_{\mathrm{gr}}(f^*) \tag{36}$$

with probability at least $1 - (k/q)^k$. Furthermore, for every $\xi \in (1, \infty]$, there exists a finite constant c_2, which depends only on ξ, such that if $\lambda_n \geq c_2\sigma\big[r_n^2 + r_n\sqrt{\log(q)/n}\big]$, then

$$\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2 \lesssim \sigma^2 k\Big[r_n^2 + \frac{\log(q)}{n}\Big]\big[\psi(2k,\xi)\big]^{-2} + \lambda_n \mathrm{Pen}_{\mathrm{gr}}(f^*) \tag{37}$$

with probability at least $1 - 1/q$.

We make the following observations regarding the established error bounds. To simplify the comparison of the error rates, we focus on the setting where $k = k_*$ and $\mathrm{Pen}_{\mathrm{gr}}(f^*) \asymp \sigma k_*$. The last relationship holds, for example, when the scaled roughness of each nonzero component, $\mathrm{Pen}(f_j^*)/\sigma$, is bounded above and below by positive universal constants.

Remark 5. *The expression in error bound (36) is optimized for the setting where* $\mathrm{Pen}_{\mathrm{gr}}(f^*) \asymp \sigma k$. *However, as we show in the proof, the bound can be improved when* σk *and* $\mathrm{Pen}_{\mathrm{gr}}(f^*)$ *have different orders of magnitude.*

Remark 6. *The prediction error rate provided in (37) is analogous to the rate established in [53] for the Group Lasso-based approach*[8]*, however, the latter rate replaces* $\psi(2k_*, \xi)^{-2}$ *with* $\phi(k_*, \xi)^{-2}$. *By Proposition 5, the former rate is at least as good as the latter, with a potential improvement due to the additional* ℓ_0 *group sparsity requirement in the definition of* ψ. *If for some fixed* $\xi > 1$ *quantity* $\psi(2k_*, \xi)^{-1}$ *is bounded by a universal constant, then inequality (37) yields the following prediction error rate:*

$$\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2 \lesssim \sigma^2 k_* \Big[r_n^2 + \frac{\log(q)}{n} \Big].$$

This rate matches the one established in [53] for the Group Lasso-based approach under an analogous (but somewhat stronger) assumption on $\phi(k_*, \xi)^{-1}$.

Remark 7. *Bound* (36) *yields the following error rate without imposing assumptions on the design:*

$$\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2 \lesssim \sigma^2 k_* \Big[k_*^{2\alpha} r_n^2 + \frac{\log(eq/k_*)}{n} \Big].$$

If $k_* \lesssim 1$ *or* $k_*^{2\alpha} r_n^2 \lesssim \log(eq/k_*)/n$, *then the above expression can be upper-bounded by*

$$\sigma^2 k_* \big[r_n^2 + \frac{\log(eq/k_*)}{n} \big].$$

Thus, $\widehat{f}$ *achieves the corresponding minimax lower bound on the prediction error [50, 52, 53].*

Remark 8. *When* $q = k_*$, *the prediction error rate given by bound (36) is* $k_*^{1+1/(2m+1)} r_n^2$, *which improves over the corresponding* $k_*^{1+3/(2m+1)} r_n^2$ *rate*[9] *derived in [33]. In particular, when* $m = 2$, *the former rate is* $k_*^{6/5} n^{-4/5}$, *while the latter is* $k_*^{8/5} n^{-4/5}$. *The improvement in the rate is a consequence of the more refined entropy bounds derived in our proofs.*

Remark 9. *In the special case of* $m = 2$ *and* $k_* \lesssim 1$, *bound (36) yields the prediction error rate of* $n^{-4/5} + \log(q)/n$, *which matches the optimal univariate rate of* $n^{-4/5}$ *when* $\log(q) \lesssim n^{1/5}$.

Remark 10. *If for some fixed* $\xi > 1$ *quantity* $\psi(2k_*, \xi)^{-1}$ *is bounded by a universal constant, then a direct consequence of Theorem 4 is the following estimation error rate:*

$$\sum_{j=1}^{q} \|\widehat{\mathbf{f}}_j - \mathbf{f}_j^*\|_n \lesssim \sigma k_* \Big[r_n + \sqrt{\frac{\log(q)}{n}} \Big].$$

[8] To the best of our knowledge, the bounds in [53] are overall the strongest in the literature for the Group Lasso-based approach, due to the relative weakness of the imposed conditions: see the discussion in Remark 12 of [53].

[9] Theorem 1 in [33] treats the number of predictors ($q = k_*$) as fixed and omits it from the expression for the error rate. However, an examination of the proof of their Theorem 1 and the entropy bound in their Lemma A.1, which explicitly accounts for the number of predictors, reveals the effect of the dimension k_*.

6 Experiments

We present experiments that shed light on the practical performance of our proposals compared to the state of the art. In Section 6.1, we investigate the statistical properties of our algorithms for the Group ℓ_0 problem. In Section 6.2, we present computation times of our MIP algorithm. Section 6.3 investigates nonparametric sparse additive models.

6.1 Grouped variable selection

We consider both synthetic and real datasets in our experiments, as discussed below.

Synthetic data generation. The underlying model is $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\epsilon}$, where $\boldsymbol{\beta}^* \in \mathbb{R}^p$ has q groups, all with the same size. Once we generate $\mathbf{X}$ (see below), every column is standardized to have unit ℓ_2-norm. The errors $\epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2), i = 1, \ldots, n$, are independent of $\mathbf{X}$, and σ^2 is chosen to achieve a desired signal-to-noise ratio (SNR)[10]. We note that the SNR values in our experiments are sufficiently high to make the true model support recovery possible.

Two different types of $\mathbf{X}$ are considered: (a) `example=1`: We first generate group representatives $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_q \sim \text{MVN}_q(0, \boldsymbol{\Sigma})$, where, $\boldsymbol{\Sigma}_{q\times q} = ((\sigma_{ij}))$, with $\sigma_{ij} = \rho^{|i-j|}$. Given a $\boldsymbol{\gamma}_g$, the covariates $\mathbf{x}_j, j \in \mathcal{G}_g$ are generated by adding independent Gaussian noise to a scalar multiple of $\boldsymbol{\gamma}_g$, to achieve pairwise correlation of 0.9 within the group. (b) `example=2`: Here we take $\mathbf{X} \sim \text{MVN}_p(0, \boldsymbol{\Sigma})$, where $\sigma_{ij} = \rho$, for all $i \neq j$, with $\sigma_{jj} = 1$ for all j.

To generate the true population regression coefficients, the k_* nonzero groups are taken to be equally spaced in $\{1, \ldots, q\}$. All the nonzero entries of $\boldsymbol{\beta}^*$ are drawn independently from a standard Gaussian distribution.

Competing algorithms and tuning. In the experiments of this section, we focus on the Group ℓ_0 problem defined in (1), and study the performance of our algorithms. We compare against the following state-of-the-art grouped variable selection methods: Group Lasso (based on $\ell_{2,1}$ regularization), Group MCP, and Group SCAD – these estimators are computed by using the R package `grpreg` [14]. For synthetic data, we construct a separate validation set with a fixed design. We tune the parameters of the different problems to minimize the prediction error on the validation set. Specifically, for each of Group ℓ_0 and Group Lasso, we tune the regularization parameter over a (one-dimensional) grid with 100 values. For MCP and SCAD, we tune the first parameter λ over a grid with 100 values, and leave the second parameter γ to its default value in `grpreg`.

Performance measures. Given an estimator $\hat{\boldsymbol{\beta}}$, we consider the following performance measures:

- **True Positives (TP)**: The number of nonzero groups that are in both $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}^*$.
- **False Positives (FP)**: The number of nonzero groups in $\hat{\boldsymbol{\beta}}$ but not in $\boldsymbol{\beta}^*$
- **Recovery F1 Score**: The harmonic mean of precision and recall, i.e., F1 Score $= 2PR/(P+R)$, where $P = \text{TP}/(\text{TP} + \text{FP})$ is precision and $R = \text{TP}/k_*$ is recall. We note that an F1 Score of 1 implies perfect support recovery.
- **Test MSE**: This is defined as $\frac{1}{n}\|\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2$.

[10] For a generative model of the form $y_i = \mu_i + \epsilon_i$, we define $\text{SNR} = \text{Var}(\mu)/\text{Var}(\epsilon)$.

6.1.1 Statistical performance for varying number of observations

In this experiment, we study the effect of varying the number of observations n on the performance of Group ℓ_0 and other state-of-the-art group regularizers (Group Lasso, MCP, and SCAD). We obtain approximate estimators to the Group ℓ_0 problem using Algorithms 1 and 2 (with $m = 1$). We generate 10 datasets having exponentially decaying correlation (i.e., under `example=1`) with a correlation parameter $\rho = 0.9$, $p = 5000$, a group size of 4, number of nonzero groups $k_* = 25$, and SNR $= 10$. This setting is relatively difficult for recovery as each group is highly correlated with a few others. We report the average performance measures over the 10 datasets in Figure 1.

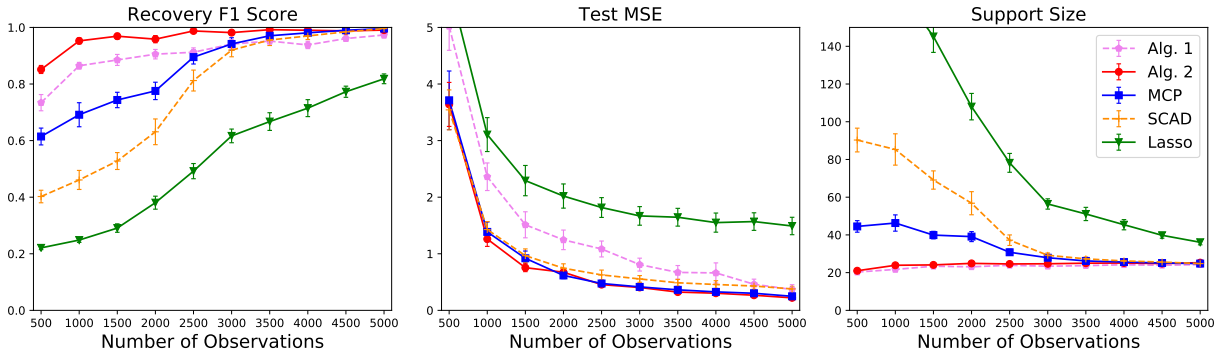

Figure 1: Performance measures for varying number of observations on a synthetic dataset with highly correlated features. Alg. 1 and Alg. 2 are our proposed algorithms. Here, "Lasso" is a shorthand for Group Lasso, we use the same convention for SCAD, MCP.

Figure 1 shows that Algorithm 2 notably outperforms the other methods in terms of variable selection; it perfectly recovers the support for $n \approx 2000$. Group MCP and SCAD require roughly 4500 observations to recover the true support, whereas Group Lasso does not recover the support even when $n = p$. Moreover, Algorithms 1 and 2 attain the smallest support sizes for any n, whereas the other methods require much larger supports, especially for small n. Algorithm 2 has the lowest test MSE for all n. The MSE of MCP matches that of Algorithm 2 in most of the cases, while the other methods lag behind. We also note that there is a gap between the MSE of Algorithms 1 and 2. This difference is likely due to Algorithm 2 doing a better job in optimization.

6.1.2 Statistical performance on high-dimensional instances

We compare the performance of the different methods under two high-dimensional settings. In both settings, we generate data with constant correlation (i.e., under `example=2`) and SNR $= 10$. Below is a description of the settings:

- Setting 1: $\rho = 0.9, n = 1000, p = 100,000, k = 10$, and a group size of 10.
- Setting 2: $\rho = 0.3, n = 1000, p = 100,000, k = 20$, and a group size of 4.

For each setting, we generate 10 random training and validation datasets, on which we train and tune the algorithms. To ensure a fair comparison in terms of running time, we solve the Group

ℓ_0 problem approximately using Algorithm 2 (with $m = 1$), which typically has the same order of running time (seconds in this case) as the other group selection methods considered here. We report the averaged results for Settings 1 and 2 in Table 1.

Table 1: Performance measures for Setting 1 (top panel) and Setting 2 (bottom panel). Means are reported along with their standard errors.

	Algorithm	$\|\hat{\boldsymbol{\beta}}\|_0$	TP	FP	MSE	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_\infty$
Setting 1	Group ℓ_0	98.0 (2.5)	9.7 (0.2)	0.1 (0.1)	7.8 (1.7)	0.8 (0.1)
	Group Lasso	2108 (222.6)	10.0 (0.0)	200.8 (22.3)	19.8 (3.4)	1.4 (0.12)
	Group MCP	294 (44.3)	10.0 (0.0)	19.4 (4.4)	11.7 (3.2)	0.95 (0.17)
	Group SCAD	637 (98.6)	10.0 (0.0)	53.7 (9.9)	18.4 (5.4)	1.2 (0.22)
Setting 2	Group ℓ_0	79.2 (1.3)	19.6 (0.2)	0.2 (0.1)	0.97 (0.08)	0.35 (0.03)
	Group Lasso	1139.2 (63.3)	19.9 (0.1)	264.9 (15.7)	4.42 (0.29)	0.67 (0.03)
	Group MCP	146.0 (15.6)	19.8 (0.1)	16.7 (3.9)	1.07 (0.07)	0.38 (0.03)
	Group SCAD	300.0 (36.3)	20.0 (0.0)	55.0 (9.1)	1.26 (0.10)	0.45 (0.05)

Under both settings, Group ℓ_0 selects significantly smaller support sizes and false positives than other methods, and is more consistent across the replications (as evidenced by the small standard error). For example, in Table 1 (top), Group ℓ_0 has a support size which is roughly 20 times smaller than the one for the Lasso and 3 times smaller than one for MCP. For few of the instances, one true positive is missed in Group ℓ_0, but the difference with the other methods is marginal. In terms of MSE and the estimation error (i.e., $\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_\infty$), Group ℓ_0 appears to outperform the other methods, with the differences being most pronounced in the high correlation setting of Table 1 (top). This aligns with the results in Figure 1, where we saw that Group ℓ_0 leads to important improvements when features are highly correlated and n is small.

6.1.3 Real data

We study the performance of the different methods on the Amazon Reviews dataset [24]. After preprocessing, the dataset consists of 3482 predictors divided into 100 groups. We use 3500 and 2368 observations for training and testing, respectively. Additional details on the dataset and preprocessing are discussed in the Supplement D. On this dataset, we fit regularization paths for Group ℓ_0, Lasso, and SCAD[11]. For Group ℓ_0, we use an additional ridge regularization term[12] and consider $\lambda_2 \in \{0.5, 1, 2\}$. In Figure 2, we plot the test MSE at different sparsity levels. The results indicate that the lowest MSE is roughly the same for Group ℓ_0 ($\lambda_2 = 1$), Lasso, and SCAD; with Group ℓ_0 having a clear advantage in terms of the support size. Specifically, Group ℓ_0 with $\lambda_2 = 1$ attains the lowest MSE at 5 groups whereas Group Lasso and SCAD require around 60 groups to achieve a similar MSE performance.

In the Supplement C.1, we report results on another real dataset; and our conclusions are qualitatively similar to the example in Figure 2.

[11] We also tried group MCP, but the solver faced numerical problems—hence, their results are not reported.

[12] This is found to be useful here due to high feature correlations within a group.

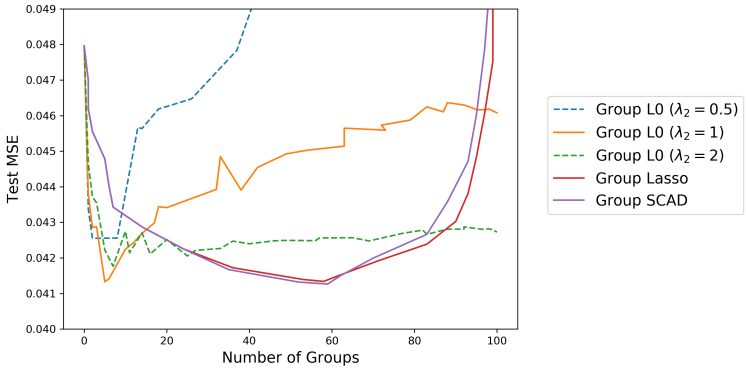

Figure 2: Test MSE on the Amazon Reviews dataset ($n = 3500$, $p = 3368$, and $q = 100$). For Group ℓ_0, we consider additional ridge regularization and vary the corresponding regularization parameter $\lambda_2 \in \{0.5, 1, 2\}$.

6.2 MIP-based global optimality certificates: Timing comparisons

Here, we compare the running time of our BnB solver with Gurobi for obtaining globally optimal solutions (we note that Algorithms 1, 2 presented earlier are approximate algorithms.) We generate synthetic data under `example=2`, and we study the effect of the number of predictors p on the running time. Specifically, we vary $p \in \{10^3, 10^4, 10^5, 10^6, 5 \times 10^6\}$ and fix the other data generation parameters as follows: group size of 10, $n = 10^3$, $\rho = 0.1$, $k_* = 5$, SNR $= 10$, and set all nonzero coefficients in $\boldsymbol{\beta}^*$ to 1. We solve the MIP in (26) to optimality, for two cases: **(i)** with ridge regularization ($\lambda_2 > 0$) and **(ii)** without ridge regularization ($\lambda_2 = 0$). In both cases, we fix $\lambda_1 = 0$. For case (i), we choose (λ_0, λ_2) so that the solution obtained has k_* nonzero groups and minimizes the ℓ_2 estimation error. More formally, for a fixed choice of (λ_0, λ_2), let $\boldsymbol{\theta}(\lambda_0, \lambda_2)$ denote a solution of (26). Then, we choose the parameters of case (ii) as follows:

$$(\lambda_0^*, \lambda_2^*) \in \underset{(\lambda_0, \lambda_2)}{\arg\min} \|\boldsymbol{\theta}(\lambda_0, \lambda_2) - \boldsymbol{\beta}^*\|_2 \;\; \text{s.t.} \;\; G(\boldsymbol{\theta}(\lambda_0, \lambda_2)) = k_*.$$

We estimate $(\lambda_0^*, \lambda_2^*)$ by running Algorithm 2 on a two-dimensional grid with $\lambda_0 \in \{10^3, 2 \times 10^3, \ldots, 10^4\}$ and $\lambda_2 \in \{10^{-5}, 10^{-4}, \ldots, 10^5\}$. For case (ii), we choose λ_0 so that the corresponding solution has k_* nonzero groups. Let S^* be the support of the true solution $\boldsymbol{\beta}^*$, and let $\hat{\boldsymbol{\beta}}$ be the solution obtained by solving $\min_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta})$ s.t. $\boldsymbol{\beta}_{(S^*)^c} = 0$. Then, in both cases, we set $\mathcal{M}_{\text{U}}$ to $\max_{g \in [q]} \|\hat{\boldsymbol{\beta}}_g\|_2$. For the two solvers, we set the optimality gap[13] to 1% and use a warm start obtained from Algorithm 2. The running times were measured on a cluster with CentOS 7. Each job (i.e., a single run of a solver over one dataset) was allocated 4 cores of an Intel Xeon Gold 6130 CPU @ 2.10GHz processor and up to 120 GB of RAM. For each job, we set a time limit of 24 hours.

In Table 2, we report the running time (in seconds) for cases (i) and (ii). In both cases, the results indicate that our BnB can solve instances with $p = 5 \times 10^6$ in the order of minutes to hours, whereas Gurobi cannot solve the problem beyond $p = 10^3$ within the 24-hour time limit. Specifically, for $p \geq 10^4$, Gurobi's optimality gap is 100%. The reason behind this large gap is

[13]Given an upper bound UB and a lower bound LB, the optimality gap is defined as (UB-LB)/UB.

Table 2: Running time in seconds for solving Problem (26) to optimality. A dash (-) indicates that Gurobi cannot solve the problem in 24 hours and has an optimality gap of 100% upon termination.

p	Case (i): $\lambda_2 = \lambda_2^*$		Case (ii): $\lambda_2 = 0$	
	Ours	Gurobi	Ours	Gurobi
10^3	96	24223	373	8737
10^4	199	-	466	-
10^5	231	-	1136	-
10^6	386	-	1628	-
5×10^6	1922	-	11627	-

that Gurobi cannot solve the root relaxation in the 24-hour time limit, so the best lower bound upon termination is 0. The running times for our BnB solver in case (i) are lower than case (ii), and this can be attributed the perspective reformulation which exploits the presence of the ridge regularizer to speed up computation. It is also worth mentioning that our implementation of BnB is a prototype that does not exploit parallelism (commercial solvers like Gurobi exploit parallelism). Parallelizing our BnB implementation is expected to make it faster, especially on difficult instances where the search tree is large. In the Supplement C.2, we report the running times of our BnB and Gurobi for different choices of $\mathcal{M}_{\text{U}}$.

6.3 Nonparametric Additive Models

We study an expanded version of the popular Boston Housing dataset[14] as an application of our MIP framework to ℓ_0-sparse additive modeling. The dataset consists of 13 covariates. To get a better idea about the performance in the presence of irrelevant covariates, we augmented the data with 50 irrelevant covariates. Specifically, we selected 5 covariates uniformly at random. For each selected covariate, we randomly permuted the entries of the covariate vector and augmented the data with the permuted vector—we repeated this step 10 times. This led to 63 covariates in total. We randomly sampled 406 observations for training and 50 observations for validation, and we standardized the response and the covariates. We predict house price using the 63 covariates.

We compare the performance of sparse additive models based on Group ℓ_0 and Group Lasso. In both approaches, we used B-splines of degree 3 for the basis functions, with 10 knots equi-spaced in the covariates. For the Group ℓ_0-based approach, we used formulation (6) and tuned λ over a grid of 100 values between 10^{-5} and 10^{-2} (equi-spaced on a logarithmic scale). We obtained the Group Lasso-based approach by relaxing all the binary variables in the MIP formulation of (6) to the interval $[0, 1]$, and we tuned λ over a grid of 100 values ranging from 10^{-4} to 1 (equi-spaced on a logarithmic scale). In Figure 3, we plot the test MSE versus the number of nonzeros, for each of the two models. The results indicate that the Group ℓ_0-based approach achieves the minimum test MSE at 7 nonzeros, whereas the Group Lasso-based method achieves its minimum MSE at around 60 nonzeros (without matching the performance of Group L0).

[14]The dataset was downloaded from `https://archive.ics.uci.edu/ml/datasets/Housing`.

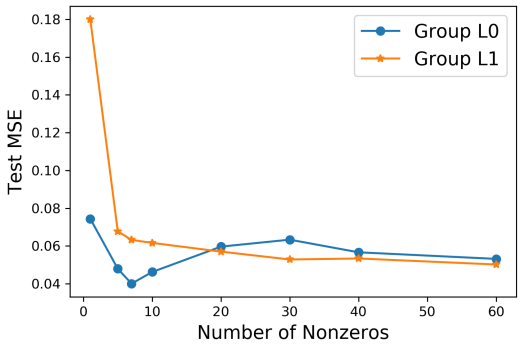

Figure 3: Test MSE versus the number of nonzeros on the Boston Housing dataset (with additional noisy covariates).

Acknowledgements

We thank Shibal Ibrahim for his help with the Boston Housing dataset experiment. The research was partially supported by the Office of Naval Research (ONR-N000141512342, ONR-N000141812298), National Science Foundation (NSF-IIS-1718258).

References

[1] S Agmon. *Lectures on Elliptic Boundary Value Problems*. Van Nostrand, Princeton, NJ, 1965.

[2] F.R. Bach. Consistency of the group lasso and multiple learning kernel. *Journal of Machine Learning Research*, 9:1179–1225, 2008.

[3] Amir Beck and Yonina C. Eldar. Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. *SIAM Journal on Optimization*, 23(3):1480–1509, 2013.

[4] Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

[5] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22, 05 2013. doi: 10.1017/S0962492913000032.

[6] D.P. Bertsekas. *Nonlinear Programming*. Athena scientific optimization and computation series. Athena Scientific, 2016. ISBN 9781886529052. URL `https://books.google.com/books?id=TwOujgEACAAJ`.

[7] Dimitris Bertsimas and Bart Van Parys. Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *The Annals of Statistics*, 48(1):300–323, 2020.

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

[9] M. S. Birman and M. Z. Solomjak. Piecewise-polynomial approximations of functions of the classes w_p^α. *Math. USSR-Sbornik*, 2(3):295–317, 1967.

[10] Robert E Bixby. A brief history of linear and mixed-integer programming computation. *Documenta Mathematica, Extra Volume: Optimization Stories*, pages 107–121, 2012.

[11] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[12] Thomas Blumensath and Mike Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5-6):629–654, 2008.

[13] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

[14] Patrick Breheny and Jian Huang. Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. *Statistics and computing*, 25(2): 173–187, 2015.

[15] P. Bühlmann and S. Van de Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer, 2011.

[16] C. Chesneau and M. Hebiri. Some theoretical results on the grouped variables lasso. *Mathematical Methods of Statistics*, 17:317–326, 2008.

[17] Jens Clausen and Michael Perregaard. On the best search strategy in parallel branch-and-bound: Best-first search versus lazy depth-first search. *Annals of Operations Research*, 90: 1–17, 1999.

[18] H. Dong, K. Chen, and J. Linderoth. Regularization vs. Relaxation: A conic optimization perspective of statistical variable selection. *ArXiv e-prints*, October 2015.

[19] Antonio Frangioni and Claudio Gentile. Perspective cuts for a class of convex 0–1 mixed integer programs. *Mathematical Programming*, 106(2):225–236, 2006.

[20] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. URL `http://www.jstatsoft.org/v33/i01/`.

[21] Oktay Günlük and Jeff Linderoth. Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Mathematical programming*, 124(1-2):183–205, 2010.

[22] T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman and Hall, London, 1990.

[23] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, FL, 2015.

[24] Hussein Hazimeh and Rahul Mazumder. Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms. *Operations Research*, 68(5):1517–1537, 2020.

[25] Hussein Hazimeh and Rahul Mazumder. Learning hierarchical interactions at scale: A convex optimization approach. In *International Conference on Artificial Intelligence and Statistics*, pages 1833–1843, 2020.

[26] Hussein Hazimeh, Rahul Mazumder, and Ali Saab. Sparse regression at scale: Branch-and-bound rooted in first-order optimization. *arXiv preprint arXiv:2004.06152*, 2020.

[27] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 507–517, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee. ISBN 9781450341431. doi: 10.1145/2872427.2883037. URL `https://doi.org/10.1145/2872427.2883037`.

[28] J. Huang and T. Zhang. The benefit of group sparsity. *The Annals of Statistics*, 38:1978–2004, 2010.

[29] J. Huang, J.L. Horowitz, and F. Wei. Variable selection in nonparametric additive models. *The Annals of Statistics*, 38:2282–2313, 2010.

[30] J. Huang, B. Breheny, and S. Ma. A selective review of group selection in high-dimensional models. *Statistical Science*, 27:481–499, 2012.

[31] Michael Jünger, Thomas M Liebling, Denis Naddef, George L Nemhauser, William R Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A Wolsey. *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-art*. Springer Science & Business Media, 2009.

[32] Vladimir Koltchinskii and Ming Yuan. Sparsity in multiple kernel learning. *The Annals of Statistics*, 38(6):3660–3695, 2010.

[33] Y. Lin and H. H. Zhang. Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics*, 34:2272–2297, 2006.

[34] K. Lounici, M. Pontil, S. van de Geer, and A. Tsybakov. Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics*, 39(4):2164–2204, 2011.

[35] Zhaosong Lu. Iterative hard thresholding methods for l0 regularized convex cone programming. *Mathematical Programming*, 147(1):125–154, Oct 2014. ISSN 1436-4646. doi: 10.1007/s10107-013-0714-4. URL `https://doi.org/10.1007/s10107-013-0714-4`.

[36] Rahul Mazumder and Peter Radchenko. The Discrete Dantzig Selector: Estimating sparse linear models via mixed integer linear optimization. *IEEE Transactions on Information Theory*, 63 (5):3053 – 3075, 2017.

[37] Rahul Mazumder, Jerome Friedman, and Trevor Hastie. Sparsenet: Coordinate descent with non-convex penalties. *Journal of the American Statistical Association*, 117(495):1125–1138, 2011.

[38] Rahul Mazumder, Peter Radchenko, and Antoine Dedieu. Subset selection with shrinkage: Sparse linear modeling when the snr is low. *arXiv preprint arXiv:1708.03288*, 2017.

[39] L Meier, S. van de Geer, and P. Bühlmann. High-dimensional additive modeling. *The Annals of Statistics*, 37:3779–3821, 2009.

[40] Alan Miller. *Subset selection in regression*. CRC Press Washington, 2002.

[41] David R Morrison, Sheldon H Jacobson, Jason J Sauppe, and Edward C Sewell. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19:79–102, 2016.

[42] Y. Nardi and A. Rinaldo. On the asymptotic properties of the group lasso estimator for linear models. *Electronic Journal of Statistics*, 2:605–633, 2008.

[43] Balas Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[44] G. Obozinski, M. J. Wainwright, and M. I. Jordan. Support and union recovery in high-dimensional multivariate regression. *The Annals of Statistics*, 39:1–47, 2011.

[45] John Tinsley Oden and Junuthula Narasimha Reddy. *An introduction to the mathematical theory of finite elements*. Wiley, New York, 1976.

[46] Art B Owen. A robust hybrid of lasso and ridge regression. *Contemporary Mathematics*, 443 (7):59–72, 2007.

[47] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[48] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12: 2825–2830, 2011.

[49] P. Radchenko and G. M. James. Variable selection using adaptive nonlinear interaction structures in high dimensions. *Journal of the American Statistical Association*, 105:1541–1553, 2010.

[50] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *Journal of Machine Learning Research*, 13(Feb):389–427, 2012.

[51] P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman. Sparse additive models. *Journal of the Royal Statistical Society, B.*, 71:1009–1030, 2009.

[52] T. Suzuki and M. Sugiyama. Fast learning rate of multiple kernel learning: Trade-off between sparsity and smoothness. *Annals of Statistics*, 41:1381–1405, 2013.

[53] Zhiqiang Tan and Cun-Hui Zhang. Doubly penalized estimation in additive regression with high-dimensional data. *The Annals of Statistics*, 47(5):2567–2600, 2019.

[54] R Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[55] Sara Van de Geer. *Empirical Processes in M-Estimation*. Cambridge University Press, Cambridge, 2000.

[56] G. Wahba. *Spline Models for Observational Data*. SIAM, Philadelphia, 1990.

[57] F. Wei and J. Huang. Consistent group selection in high-dimensional linear regression. *Bernoulli*, 16:1369–1384, 2010.

[58] Weijun Xie and Xinwei Deng. Scalable algorithms for the sparse ridge regression. *SIAM Journal on Optimization*, 30(4):3359–3386, 2020.

[59] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

[60] Ming Yuan and Ding-Xuan Zhou. Minimax optimal rates of estimation in high dimensional additive models. *The Annals of Statistics*, 44(6):2564–2593, 2016.

[61] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942, 2010.

[62] Cun-Hui Zhang and Jian Huang. The sparsity and bias of the lasso selection in high-dimensional linear regression. *Annals of Statistics*, 36(4):1567–1594, 2008.

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

A Convex Relaxation of Problem (7)

Consider Problem (7) and suppose that the solution to this problem is bounded. Moreover, we assume that the ℓ_2-norms of every group $\mathbf{f}_j$ satisfies: $\|\mathbf{f}_j\|_2 \leq \mathcal{M}_{\mathrm{U}}$. Then it follows that the problem is equivalent to:

$$\min \quad \|\mathbf{y} - \sum_{j=1}^{q} \mathbf{f}_j\|_2^2 + \lambda_0 \sum_{j\in[q]} z_j + \lambda \sum_{j=1}^{q} \|\mathbf{f}_j\|_{C_j} \quad \text{s.t.} \quad \|\mathbf{f}_j\|_2 \leq \mathcal{M}_{\mathrm{U}} z_j, z_j \in \{0,1\}, j \in [q]. \tag{A.1}$$

Relaxing the z_j's in the above to $[0,1]$, leads to the following formulation:

$$\min \quad \|\mathbf{y} - \sum_{j=1}^{q} \mathbf{f}_j\|_2^2 + \lambda \sum_{j=1}^{q} \|\mathbf{f}_j\|_{C_j} + \lambda_1 \sum_{j=1}^{q} \|\mathbf{f}_j\|_2 \quad \text{s.t.} \quad \|\mathbf{f}_j\|_2 \leq \mathcal{M}_{\mathrm{U}} \tag{A.2}$$

where $\lambda_1 := \frac{\lambda_0}{\mathcal{M}_{\mathrm{U}}}$. Next, we (i) drop the constraints in the above, and (ii) rewrite the resulting problem as follows:

$$\Gamma_1 := \min \quad \|\mathbf{y} - \sum_{j=1}^{q} \mathbf{f}_j\|_2^2 + \lambda(\sum_{j=1}^{q} \|\mathbf{f}_j\|_{C_j} + \frac{\lambda_1}{\lambda} \sum_{j=1}^{q} \|\mathbf{f}_j\|_2). \tag{A.3}$$

Note that (A.3) is a relaxation of (A.1) (and consequently of (7)). Now, using the fact that

$$(\|\mathbf{f}_j\|_{C_j} + \frac{\lambda_1}{\lambda}\|\mathbf{f}_j\|_2)/\sqrt{2} \leq \sqrt{\|\mathbf{f}_j\|_{C_j}^2 + \left(\frac{\lambda_1}{\lambda}\right)^2 \|\mathbf{f}_j\|_2^2},$$

it follows that the following

$$\Gamma_2 := \min \quad \|\mathbf{y} - \sum_{j=1}^{q} \mathbf{f}_j\|_2^2 + \sqrt{2}\lambda \sum_{j=1}^{q} \left(\sqrt{\|\mathbf{f}_j\|_{C_j}^2 + \left(\frac{\lambda_1}{\lambda}\right)^2 \|\mathbf{f}_j\|_2^2} \right), \tag{A.4}$$

is an upper bound to Problem (A.3) (with the tuning parameters kept fixed). Note that Problem (A.4) is indeed the penalty considered in [39], with the choice of $\text{Pen}(f_j) = \sqrt{\|\mathbf{f}_j\|_{C_j}^2 + \lambda' \|\mathbf{f}_j\|_2^2}$, where λ' is appropriately chosen to match (A.4).

We note that the penalty chosen in formulation (A.3) is similar to the penalty considered in [50], wherein the authors consider an RKHS framework with penalization:

$$\lambda \sum_{j=1}^{q} \sqrt{\boldsymbol{\beta}_j' \mathbf{K}^j \boldsymbol{\beta}_j} + \lambda' \sum_{j=1}^{q} \|\mathbf{K}^j \boldsymbol{\beta}_j\|_2,$$

where $\mathbf{K}^j$ indicates the kernel basis matrix for the jth coordinate.

B Proofs

B.1 Proof of Theorem 1

The following lemma shows that there is a sufficient decrease in the objective after every group update in Algorithm 1. The result of this lemma will be used in the proof of Theorem 1.

Lemma 1. *(Sufficient Decrease) The sequence of iterates $\{\boldsymbol{\theta}^l\}$ in Algorithm 1 satisfies the following for every l and $g = 1 + (l \mod q)$:*

$$h(\boldsymbol{\theta}^l) - h(\boldsymbol{\theta}^{l+1}) \geq \frac{\hat{L}_g - L_g}{2} \|\boldsymbol{\theta}_g^l - \boldsymbol{\theta}_g^{l+1}\|_2^2. \tag{B.5}$$

Proof of Lemma 1. Fix some $l \geq 0$ and let $g = 1 + (l \mod q)$. Applying (11) to $(\boldsymbol{\theta}^{l+1}, \boldsymbol{\theta}^l)$ and adding $\Omega(\boldsymbol{\theta}^{l+1})$ to both sides, we get:

$$h(\boldsymbol{\theta}^{l+1}) \leq \ell(\boldsymbol{\theta}^l) + \langle \nabla_{\boldsymbol{\theta}_g} \ell(\boldsymbol{\theta}^l), \boldsymbol{\theta}_g^{l+1} - \boldsymbol{\theta}_g^l \rangle + \frac{L_g}{2} \|\boldsymbol{\theta}_g^{l+1} - \boldsymbol{\theta}_g^l\|_2^2 + \Omega(\boldsymbol{\theta}^{l+1}). \tag{B.6}$$

By rewriting the term $\frac{L_g}{2}\|\boldsymbol{\theta}_g^{l+1} - \boldsymbol{\theta}_g^l\|_2^2$ in the above as $\frac{L_g - \hat{L}_g}{2}\|\boldsymbol{\theta}_g^{l+1} - \boldsymbol{\theta}_g^l\|_2^2 + \frac{\hat{L}_g}{2}\|\boldsymbol{\theta}_g^{l+1} - \boldsymbol{\theta}_g^l\|_2^2$ and regrouping terms, we get:

$$h(\boldsymbol{\theta}^{l+1}) \leq \tilde{g}(\boldsymbol{\theta}^{l+1}; \boldsymbol{\theta}^l) + \frac{L_g - \hat{L}_g}{2} \|\boldsymbol{\theta}_g^{l+1} - \boldsymbol{\theta}_g^l\|_2^2. \tag{B.7}$$

But $\tilde{g}(\boldsymbol{\theta}^{l+1}; \boldsymbol{\theta}^l) \leq \tilde{g}(\boldsymbol{\theta}^l; \boldsymbol{\theta}^l)$ (by the definition of $\boldsymbol{\theta}^{l+1}$ in (13)). Moreover, $\tilde{g}(\boldsymbol{\theta}^l; \boldsymbol{\theta}^l) = h(\boldsymbol{\theta}^l)$, which implies $\tilde{g}(\boldsymbol{\theta}^{l+1}; \boldsymbol{\theta}^l) \leq h(\boldsymbol{\theta}^l)$. Using the latter bound in (B.7), we arrive to the result of the lemma.

Proof of the theorem. In the rest of this proof, we utilize the following definition: $E(\boldsymbol{\theta}_S) := \ell(\boldsymbol{\theta}_S) + \lambda_1 \sum_{g \in S} \|\boldsymbol{\theta}_g\|_2$.

- **Part 1.** We will show that the event $\text{Supp}(\boldsymbol{\theta}^l) \neq \text{Supp}(\boldsymbol{\theta}^{l+1})$ cannot happen infinitely often. Suppose that $\text{Supp}(\boldsymbol{\theta}^l) \neq \text{Supp}(\boldsymbol{\theta}^{l+1})$ holds for some l. Then, either one of the following cases must hold for $g = 1 + (l \mod q)$: (I) $\boldsymbol{\theta}_g^l = 0 \neq \boldsymbol{\theta}_g^{l+1}$ or (II) $\boldsymbol{\theta}_g^l \neq 0 = \boldsymbol{\theta}_g^{l+1}$. Next, we will consider Case (I). Since $\boldsymbol{\theta}_g^{l+1} \neq 0$, then from the definition of the thresholding operator in (14), we have $\|\boldsymbol{\theta}^{l+1}\|_2 > \sqrt{\frac{2\lambda_0}{\hat{L}_g}}$. Plugging the latter inequality into Lemma 1, we get:

$$h(\boldsymbol{\theta}^l) - h(\boldsymbol{\theta}^{l+1}) \geq \frac{\hat{L}_g - L_g}{\hat{L}_g}\lambda_0. \tag{B.8}$$

  The same result in (B.8) applies for Case (II) as well. Thus, whenever the support changes, the objective improves by a positive constant (defined in the r.h.s of (B.8)), which combined with the fact that $h(\boldsymbol{\theta}) \geq 0$, implies that the support cannot change infinitely often.

- **Part 2.** First, we will show that the function $E(\boldsymbol{\theta}_S)$ is strongly convex. This trivially holds under Assumption 1(a). Next, we will assume that only Assumption 1(b) is satisfied. In this case, we have $h(\boldsymbol{\theta}^0) \leq h(\hat{\boldsymbol{\theta}})$ (where $\hat{\boldsymbol{\theta}}$ is defined in Assumption 1(b)). Since Algorithm 1 is a descent algorithm, we have $h(\boldsymbol{\theta}^l) \leq h(\hat{\boldsymbol{\theta}})$ for all $l \geq 0$. Thus, $E(\boldsymbol{\theta}^l) + \lambda_0 G(\boldsymbol{\theta}^l) \leq E(\hat{\boldsymbol{\theta}}) + \lambda_0 G(\hat{\boldsymbol{\theta}})$, which combined with the fact that $E(\boldsymbol{\theta}^l) \geq E(\hat{\boldsymbol{\theta}})$, implies that $G(\boldsymbol{\theta}^l) \leq G(\hat{\boldsymbol{\theta}})$ for all l. Thus, by the definition of k in the assumption, we have $\|\boldsymbol{\theta}^l\|_0 \leq k$ for all l. But since every k columns in $\mathbf{W}$ are linearly independent, we conclude that $E(\boldsymbol{\theta}_S)$ is strongly convex.

  After the support stabilizes (by Part 1), Algorithm 1 becomes equivalent to minimizing the strongly convex function $E(\boldsymbol{\theta}_S)$ using cyclic CD. By standard results on CD (e.g., see [6]), this is guaranteed to converge to a stationary solution $\boldsymbol{\theta}^*$ of $E(\boldsymbol{\theta}_S)$. This establishes (15).

  Finally, we will show that (16) and (17) hold. By the definition of the thresholding operator in (14), we have

$$\|\boldsymbol{\theta}_g^l\|_2 > \sqrt{\frac{2\lambda_0}{\hat{L}_g}}, \quad \forall g \in S. \tag{B.9}$$

  Taking the limit as $l \to \infty$, we arrive to (16). Similarly, we have

$$\|\nabla_{\boldsymbol{\theta}_g} \ell(\boldsymbol{\theta}^l)\|_2 \leq \sqrt{2\lambda_0 \hat{L}_g} + \lambda_1, \quad \forall g \in S^c. \tag{B.10}$$

  Taking the limit $l \to \infty$ leads to (17).

- **Part 3.** After support stabilization, Algorithm 1 is equivalent to performing cyclic CD to minimize the function $E(\boldsymbol{\theta}_S)$. Moreover, every iterate of the algorithm after support stabilization, i.e., $\boldsymbol{\theta}_S^l$ for $l \geq K$, belongs to the set $D := \{\boldsymbol{\theta}_S \mid \|\boldsymbol{\theta}_S\|_2 \geq \sqrt{\frac{2\lambda_0}{\hat{L}_g}}\}$ (this follows from (14)). Note that $\nabla_{\boldsymbol{\theta}_S} E(\boldsymbol{\theta}_S)$ is group-wise Lipschitz continuous over D, i.e., the following holds for every $g \in [q]$:

$$\|\nabla_{\boldsymbol{\theta}_S} E(\boldsymbol{\theta}_S^1) - \nabla_{\boldsymbol{\theta}_S} E(\boldsymbol{\theta}_S^2)\|_2 \leq \tilde{L}_g \|\boldsymbol{\theta}_S^1 - \boldsymbol{\theta}_S^2\|_2, \;\; \forall \boldsymbol{\theta}_S^1, \boldsymbol{\theta}_S^2 \in D \text{ s.t. } \boldsymbol{\theta}_i^1 = \boldsymbol{\theta}_i^2 \; \forall i \neq g$$

  where $\tilde{L}_g = \hat{L}_g + 2\lambda_1$. Similarly, $\nabla_{\boldsymbol{\theta}_S} E(\boldsymbol{\theta}_S)$ has a (global) Lipschitz constant of $L_S + 2|S|\lambda_1$, over D.

Lemma 3.3 of [4] bounds the objective values of cyclic CD after one full cycle. Their result holds for continuously differentiable functions whose gradient is Lipschitz over $\mathbb{R}^n$. Our function's gradient is Lipschitz over D, but we note that [4]'s result can be easily extended to D, leading to the following bound:

$$E(\boldsymbol{\theta}_S^{lq}) - E(\boldsymbol{\theta}_S^{(l+1)q}) \geq \frac{1}{2\eta}\|\nabla_{\boldsymbol{\theta}_S} E(\boldsymbol{\theta}_S^{lq})\|_2^2, \quad \forall\, l \geq K, \tag{B.11}$$

where η is defined in the statement of the theorem. In part 2, we have shown that $E(\boldsymbol{\theta}_S)$ is strongly convex. Thus, the following holds:

$$E(\boldsymbol{\alpha}_S) \geq E(\boldsymbol{\theta}_S) + \langle \nabla E(\boldsymbol{\theta}_S), \boldsymbol{\alpha}_S - \boldsymbol{\theta}_S \rangle + \frac{\sigma_S}{2}\|\boldsymbol{\alpha}_S - \boldsymbol{\theta}_S\|_2^2, \quad \forall \boldsymbol{\alpha}_S, \boldsymbol{\theta}_S. \tag{B.12}$$

Minimizing both sides in (B.12) w.r.t. $\boldsymbol{\alpha}_S$ and rearranging terms, we get

$$E(\boldsymbol{\theta}_S) - E(\boldsymbol{\theta}_S^*) \leq \frac{1}{2\sigma_S}\|\nabla_{\boldsymbol{\theta}_S} E(\boldsymbol{\theta}_S)\|_2^2, \quad \forall \boldsymbol{\theta}_S. \tag{B.13}$$

Inequalities (B.11) and (B.13) lead to:

$$(E(\boldsymbol{\theta}_S^{lq}) - E(\boldsymbol{\theta}_S^*)) - (E(\boldsymbol{\theta}_S^{(l+1)q}) - E(\boldsymbol{\theta}_S^*)) \geq \frac{1}{2\eta}\|\nabla_{\boldsymbol{\theta}_S} E(\boldsymbol{\theta}_S^{lq})\|_2^2 \tag{B.14}$$

$$\geq \frac{\sigma_S}{\eta}(E(\boldsymbol{\theta}_S^{lq}) - E(\boldsymbol{\theta}_S^*)). \tag{B.15}$$

Rearranging the terms in the above yields:

$$E(\boldsymbol{\theta}^{(l+1)q}) - E(\boldsymbol{\theta}^*) \leq \left(1 - \frac{\sigma_S}{\eta}\right)\Big(E(\boldsymbol{\theta}^{lq}) - E(\boldsymbol{\theta}^*)\Big). \tag{B.16}$$

Finally, we note that the function E in the above can be replaced by h (because of support stabilization), which establishes part 3.

B.2 Proof of Theorem 2

By Theorem 1, the support of the iterates in Algorithm 1 stabilizes, say on a support S, and converges to a solution of $\min_{\boldsymbol{\theta}, \mathrm{Supp}(\theta)=S} h(\boldsymbol{\theta})$. The latter observation along with the fact that Step 2 of Algorithm 2 ensures strict descent, imply that the sequence of solutions $\boldsymbol{\theta}^t$ in Algorithm 2 must have distinct supports. Therefore, the algorithm terminates in a finite number of iterations. Note that $\boldsymbol{\theta}^\dagger$ is the output of Algorithm 1 so it must satisfy the characterization given in part 2 of Theorem 1. Moreover, the search in Step 2 must fail at $\boldsymbol{\theta}^\dagger$, and thus (21) holds.

B.3 Proof of Proposition 2

Let $F_1(\boldsymbol{\theta}, \boldsymbol{z})$ and $F_2(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{s})$ be the objective functions in (25) and (26), respectively. Note that by definition, $v_2 = F_2(\boldsymbol{\theta}^*, \boldsymbol{z}^*, \boldsymbol{s}^*)$. Since $(\boldsymbol{\theta}^*, \boldsymbol{z}^*)$ is feasible for the problem corresponding to v_1, we have:

$$v_2 - v_1 \geq F_2(\boldsymbol{\theta}^*, \boldsymbol{z}^*, \boldsymbol{s}^*) - F_1(\boldsymbol{\theta}^*, \boldsymbol{z}^*) \tag{B.17}$$

Since $(\boldsymbol{\theta}^*, \boldsymbol{z}^*, \boldsymbol{s}^*)$ is optimal for the problem of v_2, it must satisfy $s_g^* = 0$ if $z_g^* = 0$ and $s_g^* = \frac{\|\boldsymbol{\theta}_g^*\|_2^2}{z_g^*}$ otherwise (because this is the smallest value of s_g, which satisfies (26c)). Plugging s_g^* into the term $F_2(\boldsymbol{\theta}^*, \boldsymbol{z}^*, \boldsymbol{s}^*)$ in (B.17) and simplifying, leads to the result of the proposition.

B.4 Proof of Proposition 3

The root relaxation of (26) can be written as:

$$\min_{\boldsymbol{\theta}} \quad \left\{ \tilde{\ell}(\boldsymbol{\theta}) + \lambda_1 \sum_{g=1}^{q} \|\boldsymbol{\theta}_g\|_2 + \sum_{g=1}^{q} \min_{z_g, s_g} (\lambda_0 z_g + \lambda_2 s_g) \right\} \tag{B.18}$$

$$\text{s.t.} \quad \|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}} z_g, \ g \in [q] \tag{B.19}$$

$$s_g z_g \geq \|\boldsymbol{\theta}_g\|_2^2, \quad g \in [q] \tag{B.20}$$

$$z_g \in [0,1], s_g \geq 0, \quad g \in [q] \tag{B.21}$$

Define

$$\omega(\boldsymbol{\theta}_g; \boldsymbol{\lambda}, \mathcal{M}_{\text{U}}) = \min_{z_g, s_g} (\lambda_0 z_g + \lambda_2 s_g) \quad \text{s.t.} \quad \text{(B.19)}, \text{(B.20)}, \text{(B.21)}. \tag{B.22}$$

Note that the above optimization problem appears inside the second summation of (B.18). Next, we will derive a closed form expression for (B.22). Let $(\boldsymbol{\theta}_g, z_g, s_g)$ be some feasible solution. Then, the solution $(\boldsymbol{\theta}_g, \hat{z}_g, s_g)$, where $\hat{z}_g = \max\{\frac{\|\boldsymbol{\theta}_g\|_2^2}{s_g}, \frac{\|\boldsymbol{\theta}_g\|_2}{\mathcal{M}_{\text{U}}}\}$, has an objective value which is less than or equal to that of $(\boldsymbol{\theta}_g, z_g, s_g)$ (since $\hat{z}_g$ is the smallest possible choice of z_g which satisfies all the constraints)—if $\boldsymbol{\theta}_g = \mathbf{0}$ and $s_g = 0$, we assume that $\frac{\|\boldsymbol{\theta}_g\|_2^2}{s_g} = 0$, which leads to $\hat{z}_g = 0$. Thus, replacing constraints (B.19) and (B.20) with the constraint $z = \max\{\frac{\|\boldsymbol{\theta}_g\|_2^2}{s_g}, \frac{\|\boldsymbol{\theta}_g\|_2}{\mathcal{M}_{\text{U}}}\}$ does not change the optimal objective of the problem. This replacement leads to the following equivalent problem:

$$\omega(\boldsymbol{\theta}_g; \boldsymbol{\lambda}, \mathcal{M}_{\text{U}}) = \min_{z_g, s_g} (\lambda_0 z_g + \lambda_2 s_g) \quad \text{s.t.} \quad z_g = \max\left\{ \frac{\|\boldsymbol{\theta}_g\|_2^2}{s_g}, \frac{\|\boldsymbol{\theta}_g\|_2}{\mathcal{M}_{\text{U}}} \right\}, z_g \in [0,1], s_g \geq 0. \tag{B.23}$$

In the above, we can eliminate z_g by plugging its expression into the the objective and the constraint $z_g \in [0,1]$, which leads to the following equivalent formulation:

$$\omega(\boldsymbol{\theta}_g; \boldsymbol{\lambda}, \mathcal{M}_{\text{U}}) = \min_{s_g} \max \left\{ \underbrace{\frac{\lambda_0 \|\boldsymbol{\theta}_g\|_2^2}{s_g} + \lambda_2 s_g}_{\text{Term 1}}, \underbrace{\frac{\lambda_0 \|\boldsymbol{\theta}_g\|_2}{\mathcal{M}_{\text{U}}} + \lambda_2 s_g}_{\text{Term 2}} \right\} \ \ s.t. \ \ s_g \geq \|\boldsymbol{\theta}_g\|_2^2, \|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}}. \tag{B.24}$$

Suppose that Term 1 in (B.24) attains the maximum. This holds iff Term 1 $\geq$ Term 2, which simplifies to: $s_g \leq \mathcal{M}_{\text{U}} \|\boldsymbol{\theta}_g\|_2$. Term 1 is convex in s_g, so the solution of (B.24) (obtained via solving the first order optimality condition, assuming $s_g \leq \mathcal{M}_{\text{U}} \|\boldsymbol{\theta}_g\|_2$) is given $s_g^* = \sqrt{\lambda_0/\lambda_2} \|\boldsymbol{\theta}_g\|_2$ if $\|\boldsymbol{\theta}_g\|_2 \leq \sqrt{\lambda_0/\lambda_2} \leq M$, and $s_g^* = \|\boldsymbol{\theta}_g\|_2^2$ if $\sqrt{\lambda_0/\lambda_2} \leq \|\boldsymbol{\theta}_g\|_2 \leq M$. Plugging s_g^* into (B.24), leads to $\omega(\boldsymbol{\theta}_g; \boldsymbol{\lambda}, \mathcal{M}_{\text{U}}) = 2\lambda_0 \mathcal{H}(\sqrt{\lambda_2/\lambda_0} \|\boldsymbol{\theta}_g\|_2)$, for $\sqrt{\lambda_0/\lambda_2} \leq \|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}}$.

Now suppose Term 2 attains the maximum in (B.24). There are two lower bounds on s_g in this case: $s_g \geq \mathcal{M}_{\text{U}} \|\boldsymbol{\theta}_g\|_2$ (from Term 1 $\leq$ Term 2) and $s_g \geq \|\boldsymbol{\theta}_g\|_2^2$ (from the feasible set in (B.24)). Since $\|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}}$, we have $\mathcal{M}_{\text{U}} \|\boldsymbol{\theta}_g\|_2 \geq \|\boldsymbol{\theta}_g\|_2^2$, which implies that $s_g \geq \mathcal{M}_{\text{U}} \|\boldsymbol{\theta}_g\|_2$ is the only lower bound needed. Thus, we can simplify (B.24) to:

$$\omega(\boldsymbol{\theta}_g; \boldsymbol{\lambda}, \mathcal{M}_{\text{U}}) = \min_{s_g} \frac{\lambda_0 \|\boldsymbol{\theta}_g\|_2}{\mathcal{M}_{\text{U}}} + \lambda_2 s_g \ \ s.t. \ \ s_g \geq \mathcal{M}_{\text{U}} \|\boldsymbol{\theta}_g\|_2, \|\boldsymbol{\theta}_g\|_2 \leq \mathcal{M}_{\text{U}}.$$

The optimal solution of the above is given by $s_g^* = \mathcal{M}_{\text{U}}\|\boldsymbol{\theta}_g\|_2$, and this holds for $\sqrt{\lambda_0/\lambda_2} \geq \mathcal{M}_{\text{U}}$. Plugging s_g^* into (B.24) leads to $\omega(\boldsymbol{\theta}_g; \boldsymbol{\lambda}, \mathcal{M}_{\text{U}}) = (\lambda_0/\mathcal{M}_{\text{U}} + \lambda_2\mathcal{M}_{\text{U}})\|\boldsymbol{\theta}_g\|_2$, for $\sqrt{\lambda_0/\lambda_2} \geq \mathcal{M}_{\text{U}}$. Finally, we replace the inner minimization in (B.18) by the closed form expression of $\omega(\boldsymbol{\theta}_g; \boldsymbol{\lambda}, \mathcal{M}_{\text{U}})$, which leads to the result of the proposition.

B.5 Proof of Proposition 4

Because $\kappa_{k,c} \geq \kappa_{k,1}$ for $c \geq 1$, it is sufficient to derive the stated inequality for $c = 1$.

Consider an arbitrary $\boldsymbol{\beta}$ satisfying $\boldsymbol{\beta} \neq \mathbf{0}$ and $G(\boldsymbol{\beta}) \leq 2k$. We let $J_0 \subseteq [q]$ index the k largest values in the set $\{\|\boldsymbol{\beta}_g\|_{2,1}\}_{g\in[q]}$, noting that $|J_0| = k$ and $\|\boldsymbol{\beta}_{J_0^c}\|_{2,1} \leq \|\boldsymbol{\beta}_{J_0}\|_{2,1}$. The stated inequality follows from an observation that

$$\frac{\sqrt{2k}\|\mathbf{X}\boldsymbol{\beta}\|_2}{\sqrt{n}\|\boldsymbol{\beta}\|_{2,1}} \geq \frac{\sqrt{2k}\|\mathbf{X}\boldsymbol{\beta}\|_2}{2\sqrt{n}\|\boldsymbol{\beta}_{J_0}\|_{2,1}} \geq \frac{\kappa_{k,1}}{\sqrt{2}}.$$

□

B.6 Proof of Theorem 3

Optimality of $\widehat{\boldsymbol{\beta}}$ and feasibility of $\boldsymbol{\beta}^*$ imply $\|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|_2^2 \leq \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2$, which leads to

$$\|\mathbf{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|_2^2 \leq 2\boldsymbol{\epsilon}^\top\mathbf{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*). \tag{B.25}$$

We will derive a bound for the right hand side of inequality (B.25).

First, consider a fixed subset $J \subseteq [q]$ such that $|J| = 2k$. We define $I_J = \cup_{g\in J}\mathcal{G}_g$ and $s = Tk$, noting that $|I_J| = 2s$. We choose an orthonormal basis $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_{2s}]$, such that the corresponding linear space contains the one spanned by features $\{\mathbf{x}_j\}_{j\in I_J}$. Then, $\|\boldsymbol{\Phi}^\top\boldsymbol{\epsilon}\|_2^2/\sigma^2$ has chi-square distribution with $2s$ degrees of freedom, and

$$\boldsymbol{\epsilon}^\top\mathbf{X}\boldsymbol{\theta} \leq \|\boldsymbol{\Phi}^\top\boldsymbol{\epsilon}\|_2\|\mathbf{X}\boldsymbol{\theta}\|_2,$$

for all $\boldsymbol{\theta} \in \mathbb{R}^p$ with $\text{supp}(\boldsymbol{\theta}) \subseteq I_J$. Applying a chi-square tail bound (for example, the one in Section 8.3.2 of [15]), we derive that $|\boldsymbol{\Phi}^\top\boldsymbol{\epsilon}|^2 \lesssim \sigma^2 s(1+a)$ with probability at least $1 - \exp(-2sa)$. Consequently, with probability at least $1 - \exp(-2sa)$, inequality

$$\boldsymbol{\epsilon}^T\mathbf{X}\boldsymbol{\theta} \lesssim \Big[\sigma^2 s(1+a)\Big]^{1/2}\|\mathbf{X}\boldsymbol{\theta}\|_2 \tag{B.26}$$

holds uniformly for all $\boldsymbol{\theta} \in \mathbb{R}^p$ with $\text{supp}(\boldsymbol{\theta}) \subseteq I_J$.

We now extend this bound to *all* subsets $J \subseteq [q]$ that have size $2k$. Note that the number of such subsets is bounded by $(qe/2k)^{2k}$. Applying the union bound, we deduce that inequality (B.26) holds uniformly over both such J and $\boldsymbol{\theta}$ with probability at least $1 - \exp(-2sa + 2k\log(qe/2k))$. We note that $G(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) \leq 2k$ and take $a = T^{-1}\log(eq/2k) + [2s]^{-1}\log(1/\delta_0)$. It follows that

$$\boldsymbol{\epsilon}^T\mathbf{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) \lesssim \Big[\sigma^2 k[T + \log(eq/k)] + \sigma^2\log(1/\delta_0)\Big]^{1/2}\|\mathbf{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|_2,$$

with probability at least $1 - \delta_0$. We complete the proof by combining the above bound with inequality (B.25). □

B.7 Proof of Corollary 2

We let c_0 be the universal constant from the error bound in Theorem 3 and define

$$W = \|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2 - c_0\sigma^2 k\big[T + \log(q/k)\big].$$

By Theorem 3 we have $W \leq c_0\sigma^2 \log(1/\delta_0)$ with probability at least $1 - \delta_0$. Hence,

$$\mathbb{P}\big(W > w\big) \leq e^{-w/[c_0\sigma^2]},$$

for every non-negative w. Consequently,

$$\mathbb{E}W \leq \int_0^\infty \mathbb{P}\big(W > w\big)dw \leq \int_0^\infty e^{-w/[c_0\sigma^2]}dw \leq c_0\sigma^2.$$

Thus, by the definition of W, we have

$$\mathbb{E}\|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2 \leq c_0\sigma^2 k\big[T + \log(q/k)\big] + c_0\sigma^2.$$

□

B.8 Proof of Proposition 5

Consider an arbitrary $f \in A_{2k,\xi}$. Let J_0 be the index set corresponding to the k components f_j with the largest $\|\cdot\|_n$ norm. Write r_n for $n^{-m/(2m+1)}$. Note that

$$r_n \mathrm{Pen}_g(f) \leq (\xi/2 - 1/2)\sum\nolimits_{j=1}^q \|\mathbf{f}_j\|_n \leq (\xi - 1)\sum_{j\in J_0}\|\mathbf{f}_j\|_n,$$

and hence

$$\sum_{j\notin J_0}\|\mathbf{f}_j\|_n + r_n\mathrm{Pen}_{\mathrm{gr}}(f) \leq \xi\sum_{j\in J_0}\|\mathbf{f}_j\|_n.$$

Consequently, $f \in B(J_0, \xi)$. To complete the proof, we note that

$$\frac{\sqrt{2k}\|\mathbf{f}\|_n}{\sum_{j=1}^q \|\mathbf{f}_j\|_n} \geq \frac{\sqrt{2k}\|\mathbf{f}\|_n}{2\sum_{j\in J_0}\|\mathbf{f}_j\|_n} \geq \phi(k,\xi)/\sqrt{2}.$$

□

B.9 Proof of Theorem 4

By analogy with the $\|\cdot\|_n$ notation, we define $(\boldsymbol{\epsilon}, \mathbf{v})_n = (1/n)\sum_{i=1}^n \epsilon_i v_i$, for each $\mathbf{v} \in \mathbb{R}^n$. The global optimality of $\widehat{f}$, together with the feasibility of f^*, implies the following inequality:

$$\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2 + \lambda_n\mathrm{Pen}_{\mathrm{gr}}(\widehat{f}) \leq 2(\boldsymbol{\epsilon}, \widehat{\mathbf{f}} - \mathbf{f}^*)_n + \lambda_n\mathrm{Pen}_{\mathrm{gr}}(f^*). \tag{B.27}$$

To control the term $(\boldsymbol{\epsilon}, \widehat{\mathbf{f}} - \mathbf{f}^*)_n$ we need the following result, which is proved in Section B.10.

Lemma 2. *Let $\mathcal{F}_s = \{f : f \in \mathcal{C}_{\rm gr},\ G(f) \le s\}$. Then, with probability at least $1-\epsilon$, inequality*

$$\begin{aligned}(\boldsymbol{\epsilon}/\sigma, \mathbf{f})_n &\lesssim \Big[s^{1/2+\gamma/(2m)} r_n + \sqrt{\frac{s\log(eq/s)}{n}} + \sqrt{\frac{\log(1/\epsilon)}{n}}\Big]\|\mathbf{f}\|_n \\ &\quad + \Big[s^{1/2-\gamma(2m-1)/(2m)} r_n^2 + s^{-\gamma} r_n \sqrt{\frac{s\log(eq/s)}{n}} + s^{-\gamma} r_n \sqrt{\frac{\log(1/\epsilon)}{n}}\Big]\mathrm{Pen}_{\rm gr}(f)\end{aligned}$$

holds uniformly over $f \in \mathcal{F}_s$.

We now prove inequalities (36) and (37) in the statement of Theorem 4.

Proof of inequality (36). Note that $G(\widehat{f} - f^*) \le 2k$. Applying Lemma (2) with $f = \widehat{f} - f^*$, $s = 2k$ and $\epsilon = (k/q)^k$, we conclude that, with probability at least $1 - (k/q)^k$,

$$\begin{aligned}(\boldsymbol{\epsilon}/\sigma, \widehat{\mathbf{f}} - \mathbf{f}^*)_n &\le \tilde{c}_1 k^{1/2}\Big[k^{\gamma/(2m)} r_n + \sqrt{\frac{\log(eq/k)}{n}}\Big]\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n \\ &\quad + (c_1/4)\Big[k^{1/2-\gamma(2m-1)/(2m)} r_n^2 + k^{1/2-\gamma} r_n \sqrt{\frac{\log(eq/k)}{n}}\Big]\mathrm{Pen}_{\rm gr}(\widehat{f} - f^*) \quad \text{(B.28)}\end{aligned}$$

for some universal constants $\tilde{c}_1$ and c_1.

For the remainder of the proof we restrict our attention to the random event on which (B.28) holds. We will establish a general prediction error bound, from which inequality (36) will follow by setting $\gamma = m/(2m+1)$. We let

$$\begin{aligned}\tau_n &:= 2\tilde{c}_1 \sigma k^{1/2}\Big[k^{\gamma/(2m)} r_n + \sqrt{\frac{\log(eq/k)}{n}}\Big] \quad \text{and} \\ \lambda_n &\ge c_1 \sigma\Big[k^{1/2-\gamma(2m-1)/(2m)} r_n^2 + k^{1/2-\gamma} r_n \sqrt{\frac{\log(eq/k)}{n}}\Big],\end{aligned}$$

noting that when $\gamma = m/(2m+1)$, the last inequality matches the corresponding lower-bound on λ_n in the statement of Theorem 4. Multiplying inequality (B.27) by two and then applying (B.28) with $f = \widehat{f} - f^*$, we derive

$$\begin{aligned}2\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2 + \lambda_n \mathrm{Pen}_{\rm gr}(\widehat{f}) &\le 2\tau_n \|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n + 3\lambda_n \mathrm{Pen}_{\rm gr}(f^*) \\ &\le \|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2 + \tau_n^2 + 3\lambda_n \mathrm{Pen}_{\rm gr}(f^*).\end{aligned}$$

Consequently,

$$\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2 \lesssim \sigma^2 k\Big[k^{\gamma/m} r_n + \frac{\log(eq/k)}{n}\Big] + \lambda_n \mathrm{Pen}_{\rm gr}(f^*).$$

Inequality (36) then follows from the above bound by letting $\gamma = m/(2m+1)$. We note that this choice of γ optimizes the prediction error rate in the setting where $\mathrm{Pen}_{\rm gr}(f^*) \asymp \sigma k$, however, the rate can be improved when $\mathrm{Pen}_{\rm gr}(f^*)$ and σk have different orders of magnitude.

Proof of inequality (37). Applying Lemma (2) with $s = 1$ and $\epsilon = 1/q$, we deduce that with probability at least $1 - 1/q$, inequality

$$(\boldsymbol{\epsilon}/\sigma, \mathbf{f}_j)_n \lesssim \Big[r_n + \sqrt{\frac{\log(q)}{n}}\Big]\Big[\|\mathbf{f}_j\|_n + r_n \mathrm{Pen}(f_j)\Big]$$

holds uniformly over $f \in \mathcal{C}_{\text{gr}}$ and $j \in [q]$. The above bound implies that there exists a universal constant c_0, such that

$$(\boldsymbol{\epsilon}/\sigma, \mathbf{f})_n = \sum_{j=1}^{q} (\boldsymbol{\epsilon}/\sigma, \mathbf{f}_j)_n \leq c_0 \Big[r_n + \sqrt{\frac{\log(q)}{n}} \Big] \Big[\sum_{j=1}^{q} \|\mathbf{f}_j\|_n + r_n \text{Pen}_{\text{gr}}(f) \Big].$$

Letting $f = \widehat{f} - f^*$, we conclude that

$$(\boldsymbol{\epsilon}/\sigma, \widehat{\mathbf{f}} - \mathbf{f}^*)_n \leq c_0 \Big[r_n + \sqrt{\frac{\log(q)}{n}} \Big] \Big[\sum_{j=1}^{q} \|\widehat{\mathbf{f}}_j - \mathbf{f}_j^*\|_n + r_n \text{Pen}_{\text{gr}}(\widehat{f} - f^*) \Big] \tag{B.29}$$

with probability at least $1 - 1/q$.

For the remainder of the proof we restrict our attention to the random event on which (B.29) holds. We define $\mu_n = 4c_0\sigma \big[r_n + \sqrt{\log(q)/n} \big]$ and let $\lambda_n \geq 4\mu_n r_n \xi/(\xi - 1)$. Applying inequality (B.29), we rewrite inequality (B.27) as follows:

$$2\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2 + \lambda_n \text{Pen}_{\text{gr}}(\widehat{f} - f^*) \leq \mu_n \sum_{j=1}^{q} \|\widehat{\mathbf{f}}_j - \mathbf{f}_j^*\|_n + 3\lambda_n \text{Pen}_{\text{gr}}(f^*). \tag{B.30}$$

We now consider two possible cases.

Case i): $\mu_n \sum_{j=1}^{q} \|\widehat{\mathbf{f}}_j - \mathbf{f}_j^*\|_n \geq 3\lambda_n \text{Pen}_{\text{gr}}(f^*)$. It follows that

$$2\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2 + \lambda_n \text{Pen}_{\text{gr}}(\widehat{f} - f^*) \leq 2\mu_n \sum_{j=1}^{q} \|\widehat{\mathbf{f}}_j - \mathbf{f}_j^*\|_n, \tag{B.31}$$

and, consequently, $2r_n \text{Pen}_{\text{gr}}(\widehat{f} - f^*) \leq 4(\mu_n r_n/\lambda_n) \sum_{j=1}^{q} \|\widehat{\mathbf{f}}_j - \mathbf{f}_j^*\|_n \leq (\xi - 1) \sum_{j=1}^{q} \|\widehat{\mathbf{f}}_j - \mathbf{f}_j^*\|_n$. Taking into account inequality $G(\widehat{f} - f^*) \leq 2k$ and Definition 2, we then derive

$$\sum_{j \leq q} \|\widehat{\mathbf{f}}_j - \mathbf{f}_j^*\|_n \leq [2k]^{1/2} [\psi(2k, \xi)]^{-1} \|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n. \tag{B.32}$$

Combining this bound with inequality (B.31), we colclude

$$\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2 \leq \mu_n [2k]^{1/2} [\psi(2k, \xi)]^{-1} \|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n,$$

which implies the stated prediction error bound.

Case ii): $\mu_n \sum_{j=1}^{q} \|\widehat{\mathbf{f}}_j - \mathbf{f}_j^*\|_n < 3\lambda_n \text{Pen}_{\text{gr}}(f^*)$. Going back to inequality (B.30), we derive

$$2\|\widehat{\mathbf{f}} - \mathbf{f}^*\|_n^2 + \lambda_n \text{Pen}_{\text{gr}}(\widehat{f} - f^*) \leq 6\lambda_n \text{Pen}_{\text{gr}}(f^*),$$

which implies the stated prediction error bound. □

B.10 Proof of Lemma 2

Given $J \subseteq [q]$, we define a functional class $\mathcal{F}(J) = \{f : \ f(\mathbf{x}) = \sum_{j \in J} f_j(x_j), \ f_j \in \mathcal{C}\}$. We will need the following result, which is proved in Section B.11.

Lemma 3. *Let $J \subseteq [q]$. Then, with probability at least $1 - e^{-t}$, inequality*

$$(\boldsymbol{\epsilon}/\sigma, \mathbf{f})_n \lesssim \Big[|J|^{1/2+\gamma/(2m)} r_n + \sqrt{t/n}\Big] \|\mathbf{f}\|_n + \Big[|J|^{1/2-\gamma(2m-1)/(2m)} r_n^2 + |J|^{-\gamma} r_n \sqrt{t/n}\Big] Pen_{\mathrm{gr}}(f)$$

holds uniformly over $f \in \mathcal{F}(J)$.

Let M_s denote the number of distinct subsets of $[q]$ that have size s. We note that $\log(M_s) \leq s \log(eq/s)$ and, thus, $M_s e^{-t} \leq e^{s\log(eq/s)-t}$. Applying Lemma 3 together with the union bound, we derive that, with probability at least $1 - e^{s\log(eq/s)-t}$, inequality

$$(\boldsymbol{\epsilon}/\sigma, \mathbf{f})_n \lesssim \Big[s^{1/2+\gamma/(2m)} r_n + \sqrt{t/n}\Big] \|\mathbf{f}\|_n + \Big[s^{1/2-\gamma(2m-1)/(2m)} r_n^2 + s^{-\gamma} r_n \sqrt{t/n}\Big] \mathrm{Pen}_{\mathrm{gr}}(f)$$

holds uniformly over $f \in \mathcal{F}_s$. We complete the proof by noting that for $t = s\log(eq/s) + \log(1/\epsilon)$ the above inequality becomes

$$\begin{aligned}(\boldsymbol{\epsilon}/\sigma, \mathbf{f})_n \quad \lesssim \quad & \Big[s^{1/2+\gamma/(2m)} r_n + \sqrt{\frac{s\log(eq/s)}{n}} + \sqrt{\frac{\log(1/\epsilon)}{n}}\Big] \|\mathbf{f}\|_n \\ & + \Big[s^{1/2-\gamma(2m-1)/(2m)} r_n^2 + s^{-\gamma} r_n \sqrt{\frac{s\log(eq/s)}{n}} + s^{-\gamma} r_n \sqrt{\frac{\log(1/\epsilon)}{n}}\Big] \mathrm{Pen}_{\mathrm{gr}}(f),\end{aligned}$$

and the corresponding lower-bound on the probability simplifies to $1 - \epsilon$. □

B.11 Proof of Lemma 3

Given a positive constant δ and a metric space $\mathcal{H}$ endowed with the norm $\|\cdot\|$, we use the standard notation and write $H(\delta, \mathcal{H}, \|\cdot\|)$ for the δ-entropy of $\mathcal{H}$ with respect to $\|\cdot\|$. More specifically, $H(\delta, \mathcal{H}, \|\cdot\|)$ is the natural logarithm of the smallest number of balls with radius δ needed to cover $\mathcal{H}$.

With a slight abuse of notation, we extend the domain of $\|\cdot\|_n$ from vectors in $\mathbb{R}^n$ to real-valued functions on $[0,1]^q$ by letting $\|\cdot\|_n$ be the empirical L_2-norm. Thus, given a function h, we let $\|h\|_n = [\sum_{i=1}^n h(\mathbf{x}_i)^2/n]^{1/2}$. This extension is consistent in the sense that $\|f\|_n = \|\mathbf{f}\|_n$ and $\|f_j\|_n = \|\mathbf{f}_j\|_n$ for $f \in \mathcal{C}_{\mathrm{gr}}$, $j \in [q]$.

We let $\mathcal{H}(J) = \{h : \ h \in \mathcal{F}(J),\ \|h\|_n/(r_n|J|^{-\gamma}) + \mathrm{Pen}_{\mathrm{gr}}(h) \leq 1\}$, noting that $\|h\|_n \leq r_n|J|^{-\gamma}$ and $\mathrm{Pen}_{\mathrm{gr}}(h) \leq 1$ for every $h \in \mathcal{H}(J)$. By Corollary 8.3 in [55] (cf. Lemma 12 in the supplementary material for [53]),

$$\sup_{h \in \mathcal{H}(J)} (\boldsymbol{\epsilon}/\sigma, \mathbf{h})_n \lesssim n^{-1/2} \int_0^{r_n|J|^{-\gamma}} \sqrt{H(u, \mathcal{H}(J), \|\cdot\|_n)}\, du + r_n|J|^{-\gamma}\sqrt{t/n} \tag{B.33}$$

with probability at least $1 - e^{-t}$. To bound the entropy, we will use the following result, proved in Section B.12.

Lemma 4. *$H(u, \mathcal{H}(J), \|\cdot\|_n) \lesssim |J|(1/u)^{1/m}$ for $u \in (0,1)$.*

Noting that $r_n = n^{-m/(2m+1)}$ and, thus, $n^{-1/2} = r_n^{(2m+1)/(2m)}$, we derive

$$\begin{aligned} n^{-1/2}\int_0^{r_n|J|^{-\gamma}} \sqrt{H(u,\mathcal{H}(J),\|\cdot\|_n)}du &\lesssim n^{-1/2}\int_0^{r_n|J|^{-\gamma}} |J|^{1/2}u^{-1/(2m)}du \\ &\lesssim |J|^{1/2}n^{-1/2}\Big[r_n|J|^{-\gamma}\Big]^{(2m-1)/(2m)} \\ &= r_n^{(2m+1)/(2m)+(2m-1)/(2m)}|J|^{1/2-\gamma(2m-1)/(2m)} \\ &= r_n^2|J|^{1/2-\gamma(2m-1)/(2m)}. \end{aligned}$$

Applying bound (B.33), we conclude that

$$\sup_{h\in\mathcal{H}(J)} (\boldsymbol{\epsilon}/\sigma,\mathbf{h})_n \lesssim r_n^2|J|^{1/2-\gamma(2m-1)/(2m)} + r_n|J|^{-\gamma}\sqrt{t/n}$$

with probability at least $1-e^{-t}$. The statement of the lemma is then a consequence of the fact that for every $f\in\mathcal{F}(J)$, function $f/\big[\|f\|_n/(r_n|J|^{-\gamma}) + \mathrm{Pen}_{\mathrm{gr}}(f)\big]$ falls in the class $\mathcal{H}(J)$. □

B.12 Proof of Lemma 4

We will establish the stated entropy bound for a somewhat larger functional space $\mathcal{H}'_J = \{h : h\in\mathcal{F}(J),\ \|h\|_n + \mathrm{Pen}_{\mathrm{gr}}(h)\le 1\}$. We treat m as fixed, so that universal constants in inequalities below are allowed to depend on m.

Consider an arbitrary $g\in\mathcal{C}$. By the Sobolev embedding theorem [for example, 45, Theorem 3.13], we can write g as a sum of a polynomial of degree $m-1$ and a function $\tilde{g}$ that satisfies $\|\tilde{g}\|_{L_2}\lesssim \mathrm{Pen}(g)$, where we note that $\mathrm{Pen}(g)=\mathrm{Pen}(\tilde{g})$. Applying Lemma 10.9 in [55], which builds on the interpolation inequality of [1], we derive $\|\tilde{g}\|_\infty\lesssim\mathrm{Pen}(\tilde{g})$. Thus, $\mathcal{H}'_J\subseteq\{p+\tilde{h}:\ p\in\mathcal{P}_J,\ \tilde{h}\in\tilde{\mathcal{H}}_J\}$, where

$$\begin{aligned} \mathcal{P}_J &= \{p:\ p(\mathbf{x})=\alpha_0+\sum_{j\in J}\sum_{l=1}^{m-1}\alpha_{jl}x_j^l,\ \alpha_0\in\mathbb{R},\ \alpha_{jl}\in\mathbb{R}\ \forall j,k,\ \|p\|_n\le 2\} \\ \tilde{\mathcal{H}}_J &= \{\tilde{h}:\ \tilde{h}\in\mathcal{F}(J),\ \mathrm{Pen}_{\mathrm{gr}}(\tilde{h})\le 1,\ \|\tilde{h}_j\|_\infty\lesssim\mathrm{Pen}(\tilde{h}_j)\ \forall j\in J\}. \end{aligned}$$

We are able to impose the bound $\|p\|_n\le 2$ in the definition of $\mathcal{P}_J$, because if $h=p+\tilde{h}$ for $h\in\mathcal{H}'_J$ and $\tilde{h}\in\tilde{\mathcal{H}}_J$, then $\|p+\tilde{h}\|_n\le 1$ and $\|\tilde{h}\|_n\le\mathrm{Pen}_{\mathrm{gr}}(\tilde{h})\le 1$. Consequently,

$$H(u,\mathcal{H}(J),\|\cdot\|_n)\le H(u,\mathcal{H}'_J,\|\cdot\|_n)\le H(u/2,\mathcal{P}_J,\|\cdot\|_n)+H(u/2,\tilde{\mathcal{H}}_J,\|\cdot\|_\infty), \tag{B.34}$$

where we used the fact that the unit ball with respect to the $\|\cdot\|_\infty$-norm is contained within the corresponding ball with respect to the $\|\cdot\|_n$-norm. We note that $\mathcal{P}_J$ is a ball of radis 2, with respect to the $\|\cdot\|_n$-norm, in a linear functional space of dimension $|J|(m-1)+1$. Hence, $H(u/2,\mathcal{P}_J,\|\cdot\|_n)\lesssim |J|+|J|\log(1/u)$ by, for example, Corollary 2.6 in [55]. Thus, the result of Lemma 4 follows from B.34 if we also establish that $H(\delta,\tilde{\mathcal{H}}_J,\|\cdot\|_\infty)\lesssim|J|(1/\delta)^{1/m}$ for $\delta\in(0,1)$.

It is only left to derive the stated bound on $H(\delta,\tilde{\mathcal{H}}_J,\|\cdot\|_\infty)$. Note that we can represent functional class $\tilde{\mathcal{H}}_J$ as follows:

$$\tilde{\mathcal{H}}_J=\left\{\tilde{h}:\ \tilde{h}(\mathbf{x})=\sum_{j\in J}\lambda_j g_j(x_j),\ \sum_{j\in J}|\lambda_j|\le 1\ g_j\in\mathcal{C},\ \mathrm{Pen}(g_j)\le 1,\ \|g_j\|_\infty\le 1\ \forall j\in J\right\}.$$

Given functions $\tilde{h}(\mathbf{x}) = \sum_{j\in J} \lambda_j g_j(x_j)$ and $\tilde{h}'(\mathbf{x}) = \sum_{j\in J} \lambda'_j g'_j(x_j)$ in $\tilde{\mathcal{H}}_J$, we have

$$\begin{aligned}
\|\tilde{h} - \tilde{h}'\|_\infty &\leq \|\sum_{j\in J} \lambda_j g_j - \sum_{j\in J} \lambda_j g'_j\|_\infty + \|\sum_{j\in J} \lambda_j g'_j - \sum_{j\in J} \lambda'_j g'_j\|_\infty \\
&\leq \max_{j\in J} \|g_j - g'_j\|_\infty \sum_{j\in J} |\lambda_j| + \max_{j\in J} \|g'_j\|_\infty \sum_{j\in J} |\lambda_j - \lambda'_j| \\
&\leq \max_{j\in J} \|g_j - g'_j\|_\infty + \sum_{j\in J} |\lambda_j - \lambda'_j|.
\end{aligned}$$

Consequently, if we let $\mathcal{G} = \{g : \ g \in \mathcal{C},\ \text{Pen}(g) \leq 1,\ \|g\|_\infty \leq 1\}$, let $\|\cdot\|_1$ denote the ℓ_1-norm and let B_1^d denote a unit ℓ_1-ball in $\mathbb{R}^d$, then

$$H(\delta, \tilde{\mathcal{H}}_J, \|\cdot\|_\infty) \leq |J| H(\delta/2, \mathcal{G}, \|\cdot\|_\infty) + H(\delta/2, B_1^{|J|}, \|\cdot\|_1).$$

By the results in [9], $H(\delta/2, \mathcal{G}_j, \|\cdot\|_\infty) \lesssim (1/\delta)^{1/m}$. By the standard bounds on the covering numbers of a norm ball, $H(\delta/2, B_1^{|J|}, \|\cdot\|_1) \lesssim |J| + |J| \log(1/\delta)$. Thus, $H(\delta, \tilde{\mathcal{H}}_J, \|\cdot\|_\infty) \lesssim |J|(1/\delta)^{1/m}$ for $\delta \in (0, 1)$. □

C Additional Experimental Results

C.1 Performance on the Birthweight Dataset

We study the Birthweight dataset, taken from the `R` package `grpreg`. Here, we predict birth weight using 7 grouped covariates. The dataset has 189 observations, which we randomly split into 75% for training and 25% for testing. On this dataset, we fit regularization paths for Group ℓ_0 , Lasso, and SCAD. For Group ℓ_0, we use an additional ℓ_2 regularization and consider $\lambda_2 \in \{1, 2, 4\}$. In Figure C.1, we plot the test MSE versus the sparsity level for the different methods. The results show that the Group ℓ_0-based methods outperform Group Lasso and SCAD when the group size is 2 or more.

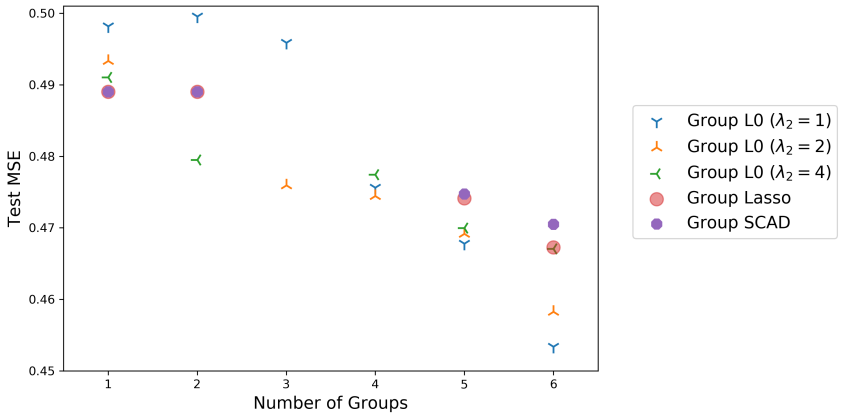

Figure C.1: Test MSE on the Birthweight dataset. For Group ℓ_0, we consider additional ridge regularization and vary the corresponding regularization parameter $\lambda_2 \in \{1, 2, 4\}$. Group sizes 3 and 4 could not be attained using Group Lasso and SCAD.

C.2 Additional Timing Comparisons

Here we consider the same setup as in the experiment of Section 6.2, and we report the running times for additional values of $\mathcal{M}_{\text{U}}$ to demonstrate the sensitivity of the runtime to $\mathcal{M}_{\text{U}}$. Let M^* be the value of $\mathcal{M}_{\text{U}}$ used in Section 6.2—note that this is the smallest value of $\mathcal{M}_{\text{U}}$. We express our choices of $\mathcal{M}_{\text{U}}$ in terms of M^*. We report the results for cases (i) and (ii) in Tables C.1 and C.2, respectively.

Table C.1: Running time in seconds for solving case (i), i.e., the MIP in (26) with $\lambda_2 = \lambda_2^*$, to optimality. A dash (-) indicates that Gurobi cannot solve the problem in 24 hours and has an optimality gap of 100% upon termination.

p	$\mathcal{M}_{\text{U}} = M^*$		$\mathcal{M}_{\text{U}} = 1.5M^*$		$\mathcal{M}_{\text{U}} = \infty$	
	Ours	Gurobi	Ours	Gurobi	Ours	Gurobi
10^3	96	24223	186	12320	192	2399
10^4	199	-	245	-	333	-
10^5	231	-	404	-	421	-
10^6	386	-	1014	-	1250	-
5×10^6	1922	-	3686	-	4036	-

Table C.2: Running time in seconds for solving case (ii), i.e., the MIP in (26) with $\lambda_2 = 0$, to optimality. A star or dash (-) indicates that the solver cannot solve the problem in 24 hours. For star, the optimality gap (in percent) is shown in parenthesis, whereas the gap is 100% for dash.

p	$\mathcal{M}_{\text{U}} = M^*$		$\mathcal{M}_{\text{U}} = 1.5M^*$		$\mathcal{M}_{\text{U}} = 2M^*$	
	Ours	Gurobi	Ours	Gurobi	Ours	Gurobi
10^3	373	8737	913	10675	1010	13901
10^4	466	-	2813	-	*(3.9)	-
10^5	1136	-	*(4.7)	-	*(20.7)	-
10^6	1628	-	*(5.1)	-	*(21.6)	-

D Additional Details on the Datasets

D.1 Description of the Amazon Reviews Dataset

This dataset is a subset of the Amazon Grocery and Gourmet Food dataset [27]. To obtain $\mathbf{X}$ and $\mathbf{y}$, we follow the same steps described in [24], and we restrict $\mathbf{X}$ to the top 5500 words in the corpus. Here $\mathbf{X}$ is a TF/IDF representation of the text reviews and $\mathbf{y}$ is a continuous variable which measures review helpfulness. To obtain the groups, we run Latent Dirichlet Allocation (LDA) [11] on the corpus using `scikit-learn` [48], where we set the number of groups to 100. We then use the LDA solution to construct a collection of probability vectors $\{\boldsymbol{\pi}^{(i)}\}_{i=1}^{100}$, each corresponding to a topic. Here $\pi_j^{(i)}$ refers to the probability of encountering word j in topic i. We assign word j to the group with index $\arg\max_i \{\pi_j^{(i)}\}_{i=1}^{100}$ (i.e., to the group that allocates j the highest probability). For example, the top 5 words in group 1 are "coffee roast cup keurig cups" so the topic is on coffee. Group 2 has "bpa worse cans dented claim", which refers to

problems with the packaging of the product. To obtain the training set, we sub-sample uniformly at random from the corpus and remove any covariates with zero variance (after sub-sampling), which reduces the number of covariates from 5500 to 3482. Note that the 100 groups have different sizes, ranging between 9 and 85.