LARGE-SCALE MULTIVARIATE SPARSE REGRESSION WITH APPLICATIONS TO UK BIOBANK

BY JUNYANG QIAN, YOSUKE TANIGAWA, RUILIN LI, ROBERT TIBSHIRANI, MANUEL A RIVAS AND TREVOR HASTIE

1Department of Statistics, Stanford University
2Department of Biomedical Data Science, Stanford University
3Institute for Computational and Mathematical Engineering, Stanford University

In high-dimensional regression problems, often a relatively small subset of the features are relevant for predicting the outcome, and methods that impose sparsity on the solution are popular. When multiple correlated outcomes are available (multitask), reduced rank regression is an effective way to borrow strength and capture latent structures that underlie the data. Our proposal is motivated by the UK Biobank population-based cohort study, where we are faced with large-scale, ultrahigh-dimensional features, and have access to a large number of outcomes (phenotypes): lifestyle measures, biomarkers, and disease outcomes. We are hence led to fit sparse reduced-rank regression models, using computational strategies that allow us to scale to problems of this size. We use a scheme that alternates between solving the sparse regression problem and solving the reduced rank decomposition. For the sparse regression component, we propose a scalable iterative algorithm based on adaptive screening that leverages the sparsity assumption and enables us to focus on solving much smaller sub-problems. The full solution is reconstructed and tested via an optimality condition to make sure it is a valid solution for the original problem. We further extend the method to cope with practical issues such as the inclusion of confounding variables and imputation of missing values among the phenotypes. Experiments on both synthetic data and the UK Biobank data demonstrate the effectiveness of the method and the algorithm. We present multiSnpnet package, available at http://github.com/junyangq/multiSnpnet that works on top of PLINK2 files, which we anticipate to be a valuable tool for generating polygenic risk scores from human genetic studies.

1. Introduction. The past two decades have witnessed rapid growth in the amount of data available to us. Many areas such as genomics, neuroscience, economics and Internet services have been producing increasingly larger datasets that have high dimension, large sample size, or both. A variety of statistical methods and computational tools have been developed to accommodate this change so that we are able to extract valuable information and insight from these massive datasets (Hastie, Tibshirani and Friedman, 2009; Efron and Hastie, 2016; Dean and Ghemawat, 2008; Zaharia et al., 2010; Abadi et al., 2016).

Keywords and phrases: Large-scale algorithm, ultrahigh-dimensional problem, sparse regression, reduced-rank regression, polygenic risk score, UK Biobank
One major motivating application for this work is the study of data from population-scale cohorts like UK Biobank with genetic data from over one million genetic variants and phenotype data from thousands of phenotypes in over 500,000 individuals (Bycroft et al., 2018). These data present unprecedented opportunities to explore very comprehensive genetic relationships with phenotypes of interest. In particular, the subset of tasks we are interested in is the prediction of a person’s phenotype value, such as disease affection status, based on his or her genetic variants.

Genome-wide association studies (GWAS) is a very powerful and widely used framework for identifying genetic variants that are associated with a given phenotype. See, for example, Visscher et al. (2017) and the references therein. It is based on the results of univariate marginal regression over all candidate variants and tries to find a subset of significant ones. While being computationally efficient and easy to interpret, GWAS has fairly limited prediction performance because at most one predictor can present in the model. If prediction performance is our main concern, it is natural to consider the class of multivariate methods, i.e. that considers multiple variants simultaneously. In the past, wide data were prevalent where only a limited number, like thousands, of samples were available. In this regime, some sophisticated multivariate methods could be applicable, though they have to more or less deal with dimension reduction or variable selection. In this setting, we handle hundreds of thousands samples and even more variables. In such cases, statistical methods and computational algorithms become equally important because only efficient algorithmic design will allow for the application of sophisticated statistical modeling. Recently, we introduced some algorithms addressing these challenges. In particular, Qian et al. (2020) proposed an iterative screening framework that is able to fit the exact lasso/elastic-net solution path in large-scale and ultrahigh-dimensional settings, and demonstrate competitive computational efficiency and superior prediction performance over previous methods.

In this paper, we consider the scenarios where multivariate responses are available in addition to the multiple predictors, and propose a suite of statistical methods and efficient algorithms that allow us to further improve the statistical performance in this large \( n \) and large \( p \) regime. Some characteristics we want to leverage and challenges we want to solve include:

1. **Statistics.** There are thousands of phenotypes available in the UK Biobank. Many of them are highly correlated with each other and can have a lot of overlap in their driving factors. By treating them separately, we lose this information that could have been used to stabilize our model estimation. The benefit of building a joint model can be seen from the following simplified model. Suppose all the outcomes \( y^k, k = 1, \ldots, q \) are independent noisy observations of a shared factor \( u = X \beta \) such that \( y^k = u + e^k \). It is easy to see that by taking an average across all the outcomes, we obtain a less noisy response \( \bar{y} \), and this will give us more accurate parameter estimation and better prediction than the model built on any of the single outcome. The assumption of such latent structure is an important approach to capturing the correlation structure among the outcomes and can bring in a significant reduction in variance if the data indeed behave in a similar way. We will formalize this belief and build a model on top of it. In addition, in the presence of high-dimensional features, we will follow the “bet on sparsity” principle (Hastie, Tibshirani...
and Friedman, 2009), and assume that only a subset of the predictors are relevant to the
prediction.

Therefore, the statistical model we will build features two major assumptions: low-
rank in the signal and sparse effect. Furthermore, we will introduce integrated steps to
systematically deal with confounders and missing values.

2. Computation. On a large-scale dataset, building a multivariate model can pose great com-
putational challenges. For example, loading the entire UK Biobank dataset into memory
with double precision will take more than one terabyte of space, while typically most
existing statistical computing tools assume that the data are already sitting in memory.
Even if large memory is available, one can always encounter data or construct features
so that it becomes insufficient. Hence, instead of expecting sufficient memory space, we
would like to find a scalable solution that is less restricted by the size of physical memory.

There is a dynamic data access mechanism provided by the operating system called
memory mapping (Bovet and Cesati, 2005) that allows for easy access to larger-than-
memory data on the disk. In essence, it carries a chunk of data from disk to memory when
needed and swap some old chunks of data out of memory when it is full. In principle,
we could add a layer of memory mapping on top of all the procedures and then access
the data as if they were in memory. However, there is one important practical component
that should never be ignored: disk I/O. This is known to be expensive in the operating
system and can greatly delay the computation if frequent disk I/Os are involved. For this
reason, we do not pursue first-order gradient-based methods such as stochastic gradient
descent (Bottou, 2010) or dual averaging (Xiao, 2010; Duchi, Agarwal and Wainwright,
2011) because it can take a large number of passes over the data for the objective function
to converge to the optimum.

To address this, we design the algorithm so that it needs as few full passes over the
data as possible while solving the exact objective. In particular, by leveraging the sparsity
assumption, we propose an adaptive screening approach that allows us to strategically
select a small subset of variables into memory, do intensive computation on the subset,
and then verify the validity of all the left-out variables. The last step is important because
we want to guarantee that the solution obtained from the algorithm is a valid solution to
the original full problem.

1.1. Reduced-Rank Regression for Multiple Responses. In the standard multivariate linear
regression model, given a model matrix \( X = (x_1, \ldots, x_p) \in \mathbb{R}^{n \times p} \) and a multivariate response
matrix \( Y = (y_1, \ldots, y_q) \in \mathbb{R}^{n \times q} \), we assume that

\[
Y = XB + E,
\]

where each row of \( E = (e_1, \ldots, e_q) \) is assumed to be an independent sample from some
multivariate Gaussian distribution \( E^{(i)} \overset{\text{iid}}{\sim} \mathcal{N}(0, \Sigma_E) \). When \( n \geq q \), it is easy to see that an
maximum likelihood estimator (MLE) can be found by solving a least squares problem with
multiple outcomes, i.e.

\[
\hat{B} \in \text{argmin}_{B \in \mathbb{R}^{p \times q}} \frac{1}{2} \| Y - XB \|_F^2,
\]

\[(1.1)\]
where \( \|A\|_F^2 = \sum_{i=1}^n \sum_{j=1}^m A_{ij}^2 \) is the squared Frobenius norm of a matrix \( A \in \mathbb{R}^{n \times m} \). When \( n \geq p \) and \( X \) has full rank, (1.1) has the closed-form solution \( \hat{B} = (X^T X)^{-1} X^T Y \). Notice that this is equivalent to solving \( q \) single-response regression problems separately.

However, in many scenarios, there can be some correlation structure in the signals that we can capture to improve the statistical efficiency of the estimator. One approach to modeling the correlation is to assume that there is a set of latent factors that act as the drivers for all the outcomes. When we assume that the dependencies of the latent factors on the raw features and the outcomes on the latent factors are both linear, it is equivalent to making a low-rank assumption on the coefficient matrix. Reduced-rank regression (Anderson, 1951, hereafter RRR) assumes that the coefficient matrix \( B \) has a fixed rank \( r \leq \min(p, q) \), or

\[
B = UV^T,
\]

where \( U = (u_1, \ldots, u_r) \in \mathbb{R}^{p \times r} \), \( V = (v_1, \ldots, v_q)^\top \in \mathbb{R}^{q \times r} \). With the decomposed coefficient matrices, an alternative way to express the multivariate model is to assume that there exists a set of latent factors \( \{z_\ell \in \mathbb{R}^n : 1 \leq \ell \leq r\} \) such that for each \( \ell \),

\[
\begin{align*}
z_\ell &= X u_\ell, \\
y_k &= Z v_k + e_k.
\end{align*}
\]

Figure 1 gives a visualization of the dependency structure described above. It can also be seen as a a multilayer perceptron (MLP) with linear activation and one hidden layer, or multitask learning with bottleneck. We notice that under the decomposition, the parameters are not identifiable. In fact, if we apply any nonsingular linear transformation \( M \in \mathbb{R}^{r \times r} \) such that \( V' = VM^\top \) and \( U' = UM^{-1} \), it yields the same model but different parameters. As a result, we also have an infinite number of MLEs.

Under the rank constraint, an explicit global solution can be obtained. Let \( MDN^\top \) be the singular value decomposition (SVD) of \( (X^T X)^{-\frac{1}{2}} X^T Y \), a set of solution is given by \( \hat{U} = (X^T X)^{-\frac{1}{2}} X^T Y N \), \( \hat{V} = N \). Velu and Reinsel (2013) has a comprehensive discussion on the model under classical large \( n \) settings.

1.2. Sparse Models in High-Dimensional Problems. In the setting of high-dimensional problems where \( p > n \), the original low-rank coefficient matrix \( B \) can be unidentifiable. Often sparsity is assumed in the coefficients to model the belief that only a subset of the features are relevant to the outcomes. To find such a sparse estimate of the coefficients, a widely used approach is to add an appropriate non-smooth penalty to the original objective function to encourage the desired sparsity structure. Common choices include the lasso penalty (Tibshirani, 1996), the elastic-net penalty (Zou and Hastie, 2005) or the group lasso penalty (Yuan and Lin, 2006). There has been a great amount of work studying the consistency of estimation and model selection under such settings. See Greenshtein and Ritov (2004); Meinshausen and Bühlmann (2006); Zhao and Yu (2006); Bach (2008); Wainwright

\[\text{We use } v_k^\top \text{ to represent the } k\text{th row of } V \text{ for convenience.}\]
Fig 1: Diagram of the reduced rank regression. The nodes in grey are latent variables. The arrows represent the dependency structure. Known as *multitask* learning in the machine learning community.

(2009); Bickel, Ritov and Tsybakov (2009); Obozinski et al. (2011); Bühlmann and Van De Geer (2011) and references therein. In particular, the group lasso, as the name suggests, encourages group-level sparsity induced by the following penalty term:

\[ P_g(\beta) = \sum_{j=1}^{J} \| \beta_j \|_2, \]

where \( \beta_j \in \mathbb{R}^{p_j} \) is the subvector corresponding the \( j \)th group of variables and \( \| \beta_j \|_2 = \sqrt{\sum_{\ell=1}^{p_j} \beta_{j,\ell}^2} \) is the vector \( \ell_2 \)-norm. The \( \ell_2 \)-norm enforces that if the fitted model has \( \| \hat{\beta}_j \|_2 = 0 \), all the elements in \( \hat{\beta}_j \) will be 0, and otherwise with probability one all the elements will be nonzero. This yields a desired group-level selection in many applications. Throughout the paper, we will adopt the group lasso penalty, defining each predictor’s coefficients across all outcomes as a distinct group, in order to achieve homogeneous sparsity across multiple outcomes. In addition to variable selection for better prediction and interpretation, we will also see the computational advantages we leverage to develop an efficient algorithm.

2. Sparse Reduced-Rank Regression. Given a rank \( r \), we are going to solve the following penalized rank-constrained optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| Y - XB \|_F^2 + \lambda \sum_{j=1}^{P} \| B_j \|_2, \\
\text{s.t.} & \quad \text{rank}(B) \leq r.
\end{align*}
\]

Alternatively, we can decompose the matrix explicitly as \( B = UV^\top \) where \( U \in \mathbb{R}^{P \times r}, V \in \mathbb{R}^{q \times r} \). It can be shown that the problem above is equivalent to the Sparse Reduced Rank
Regression (SRRR) proposed by Chen and Huang (2012):

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| Y - XUV^T \|_F^2 + \lambda \sum_{j=1}^p \| U_j \|_2, \\
\text{s.t.} & \quad V^TV = I.
\end{align*}
\]

Alternating minimization was proposed by Chen and Huang (2012) to solve this non-convex optimization problem, where two algorithms were considered: subgradient descent and a variational method. The subgradient method was shown to be faster when \( p \gg n \) and the variational method faster when \( n \gg p \). However, in each iteration, the computational complexity of either method is at least quadratic in the number of variables \( p \). It makes the problem almost intractable in ultrahigh-dimensional problems, which is common, for example, in modern genetic studies. Moreover, to obtain a model with good prediction performance, we are interested in solving the problem over multiple \( \lambda \)'s rather than a single one. For such purposes, we design a path algorithm with adaptive variable screening that will be both memory and computationally efficient.

We note that (2.2) is a non-convex optimization problem. Although the algorithm we propose next is to solve (2.2) exactly, we do not claim the solution we obtain to be any of the local minima or global minima — a property nice to have but cannot be easily guaranteed in general. The subgradient descent or variational method does not offer this guarantee. We will show later that under some regularity conditions, the solution will be optimal, but in general it is a limiting point of the some alternating minimization scheme. For convenience, in the rest of the paper, when we say a valid or exact solution to the original problem, we mean it is a limiting point of the proposed scheme. That being said, we will show empirically that the solution found by the algorithm is usually reasonably good on the landscape.

3. Fast Algorithms for Large-Scale and Ultrahigh-Dimensional Problems.

First, we present a naive version of the path solution, which will be the basis of our subsequent development. The path is defined on a decreasing sequence of \( \lambda \) values \( \lambda_{\max} = \lambda_1 > \lambda_2 > \cdots > \lambda_L \geq 0 \), where \( \lambda_{\max} \) is often defined by one that leads to the trivial (e.g. all zero) solution and the rest are often determined by an equally spaced array on the log scale. In particular, for Problem (2.1), we are able to figure out the exact lower bound of \( \lambda_{\max} \) for which the solution is trivial.

**Lemma 3.1.** In problem (2.1), if \( r > 0 \), the maximum \( \lambda \) that results in a nontrivial solution \( \hat{B}(\lambda) \) is

\[
\lambda_{\max} = \max_{1 \leq j \leq p} \| x_j^T Y \|_2.
\]

The proof is straightforward, which is a result of the Karush–Kuhn–Tucker (KKT) condition (See Boyd, Boyd and Vandenberghe (2004) for more details). We present the full argument in Appendix B.1. The naive path algorithm tries to solve the problem independently across different \( \lambda \) values.
3.1. Alternating Minimization. The algorithm is described in Algorithm 1. For each value of $\lambda$ in a pre-defined sequence $\lambda_1 > \cdots > \lambda_L \geq 0$, it applies alternating minimization to Problem (2.2) till convergence.

Algorithm 1 Alternating Minimization

1: Let $k = 0$, and initialize $U^{(0)} \in \mathbb{R}^{n \times r}$, $V^{(0)} \in \mathbb{R}^{q \times r}$.
2: while $k = 0$ or $\|U^{(k)}V^{(k)\top} - U^{(k-1)}V^{(k-1)\top}\| > \epsilon$ do
3:   V-step: Fix $U^{(k)}$, solve $V$: the orthogonal Procrustes problem
   
   \[
   \minimize_{V \in \mathbb{R}^{q \times r}: V^\top V = I} \|Y - Xu^{(k)}V^\top\|_F^2.
   \]
   Let $Y^\top Xu^{(k)} = MDN^\top$ (skinny SVD) and solve $V^{(k+1)} = MN^\top$.
4:   U-step: Fix $V^{(k+1)}$, solve $U$: the group lasso problem
   
   \[
   \minimize_{U \in \mathbb{R}^{p \times r}} \frac{1}{2} \|YV^{(k+1)} - Xu\|_F^2 + \lambda \sum_{j=1}^{p} \|U_j\|_2.
   \]
5:   $k = k + 1$
6: end while

In the V-step (3.1), we will be solving the orthogonal Procrustes problem given a fixed $U^{(k)}$. An explicit solution can be constructed from the singular value decomposition, as detailed in the following Lemma.

**Lemma 3.2.** Suppose $p \geq r$ and $Z \in \mathbb{R}^{p \times r}$. Let $Z = MDN^\top$ be its singular value decomposition, where $M \in \mathbb{R}^{p \times r}$, $D \in \mathbb{R}^{r \times r}$ and $N \in \mathbb{R}^{r \times r}$. An optimal solution to
\[
\maximize_{V \in \mathbb{R}^{p \times r}: V^\top V = I} \text{Tr}(Z^\top V)
\]
is given by $\hat{V} = MN^\top$, and the objective function has optimal value $\|Z\|_*$, the nuclear norm of $Z$. When $Z$ has full rank, the solution is unique.

**Proof.** See in Appendix B.2. 

To analyze the computational complexity of the algorithm, we see a one-time computation of $Y^\top X$ that costs $O(npq)$. In each iteration, there is $O(pqr)$ complexity for the matrix multiplication $Y^\top Xu^{(k)}$ and $O(qr^2)$ for computing the SVD and the final solution. Therefore, the per-iteration computational complexity for the V-step is $O(pqr + qr^2)$, or $O(pqr)$ when $p \gg q$.

In the U-step, we are solving a group lasso problem. Computing $YV^{(k+1)}$ takes $O(nqr)$ time. The group-lasso problem can be solved by glmnet (Friedman, Hastie and Tibshirani, 2010) with the mgaussian family. With coordinate descent, its complexity is $O(\tilde{k}pqn)$, where $\tilde{k}$ is the number of iterations until convergence and is expected to be small with a reasonable
initialization, for example, provided by warm start. Thus, the per-iteration complexity for
the U-step is \( O(nqr + knpq) \), which is \( O(kpqn) \) when \( p \gg r \).

Therefore, the overall computational complexity scales at least linearly with the number of
features, and will have a large multiplier if the sample size is large as well. While subsampling
can effectively reduce the computational cost, in high-dimensional settings, it is critical to
have sufficient samples for the quality of estimation. Instead, we seek for computational
techniques that can lower the actual number of features involved in expensive iterative
computation without giving up any statistical efficiency. Thanks to the induced sparsity by
the objective function, we are able to achieve it by variable screening.

3.2. Variable Screening for Ultrahigh-Dimensional Problems. In this section, we discuss
strategic ways to find a good subset of variables to focus on in the computation that would
allow us to reconstruct the full solution easily. In particular, we would like to iterate through
the following steps for each \( \lambda \):

1. **Screen** a strong set \( S \) and treat all the left-out variables \( S^c \) as null variables that poten-
tially have zero coefficients;
2. **Solve** a significantly smaller problem on the subset of variables \( S \);
3. **Check** an optimality condition to guarantee the constructed full solution \( \hat{B} = (\hat{B}_S, \hat{B}_{S^c}) \)
   with \( \hat{B}_{S^c} = 0 \) is indeed a valid solution to the original problem. If the condition is not
   satisfied, go back to the first step with an expanded set \( S \).

3.2.1. Screening Strategies. We have seen Lemma 3.1 that determines the entry point of
any nonzero coefficient on the solution path. Furthermore, there is evidence that the variables
entering the model (as one decreases the \( \lambda \) value) tend to have large values by this criterion.
Tibshirani et al. (2012) developed on this idea and proposed the strong rules as a sequential
variable screening mechanism. The strong rules state that in a standard lasso problem with
the model matrix \( X = (x_1, \ldots, x_p) \in \mathbb{R}^{n \times p} \) and a single response \( y \in \mathbb{R}^n \), assume \( \hat{\beta}(\lambda_{k-1}) \) is
the lasso solution at \( \lambda_{k-1} \), then the \( j \)th predictor is discarded at \( \lambda_k \) if

\[
|x_j^T (y - X \hat{\beta}(\lambda_{k-1}))| < \lambda_k - (\lambda_{k-1} - \lambda_k).
\]

The key idea is that the inner product above is almost “non-expansive” in terms of \( \lambda \). As
a result, the KKT condition suggests that the variables to be discarded by (3.3) would
have coefficient 0 at \( \lambda_k \). However it is not a guarantee. The strong rules can fail, though
failures occur rarely when \( p > n \). In any case, the KKT condition is checked to ensure the
exact solution is found. Although Tibshirani et al. (2012) focused mostly on the lasso-type
problem, they also suggested extension to general objective functions and penalties. For
general objective function \( f(\beta) \) with \( p_j \)-norm penalty \( \|\beta_j\|_{p_j} \) for the \( j \)th group, the screening
criterion will be based on the dual norm of its gradient \( \|\nabla_j f(\beta)\|_{q_j} \), where \( 1/p_j + 1/q_j = 1 \).

Inspired by the general strong rules, we propose three sequential screening strategies for
the sparse reduced rank objective (2.2), named after their respective characteristics: Multi-
Gaussian, Rank-Less and Fix-V. They are based either on the solution of a relaxed convex
problem at the same \( \lambda_k \) or on the exact solution at the previous \( \lambda_{k-1} \).
1. (Multi-Gaussian) Solve the full-rank convex problem at \( \lambda_k \) and use its active set as the candidates for the low-rank settings. The main advantage is that the screening is always stable due to the convexity. However this approach often overselects and brings extra burden to the computation. By assuming a higher rank than necessary, the effective number of responses would become more than that of a low-rank model. As a result, more variables would potentially be needed to serve for an enlarged set of responses.

2. (Rank-Less) Find variables that have large \( c_j = \|X^T_j(Y - Xu(\lambda_{k-1})V(\lambda_{k-1})^T)\|_2 \). This is analogous to the strong rules applied to the vanilla multi-response lasso ignoring the rank constraint.

3. (Fix-V) Find variables that have large \( c'_j = \|X^T_j(YV(\lambda_{k-1}) - Xu(\lambda_{k-1}))\|_2 \). This is similar to the strong rules applied in the U-step with \( V \) assumed fixed. To see the rationale better, we take another perspective. The squared error in SRRR (2.2) can also be written as

\[
\|Y - XUV^T\|^2_F = \text{Tr}(Y^TY) - 2\text{Tr}(Y^TXUV^T) + \text{Tr}(XUV^TVU^TX^T)
\]

Since \( V^TV = I \), the optimization problem becomes

\[
\min_{U,V,V^TV=I} \frac{1}{2}\|XU\|^2_F - \text{Tr}(Y^TXUV^T) + \lambda \sum_{j=1}^p \|U_j\|_2
\]

For any given \( U \), we can solve \( V = MN^T \), where \( Y^TXU = MDN^T \) is its singular value decomposition. Let \( f(U) = \frac{1}{2}\|XU\|^2_F - \|Y^TXU\|_\star \). The problem is reduced to

\[
\min_U f(U) + \lambda \sum_{j=1}^p \|U_j\|_2
\]

The general strong rule suggests that we screen based on the gradient (or subgradient if \( Y^TXU \) is not full rank); that is

\[
\nabla_U f(U) = X^TXU - X^TYMN^T = X^T(XU - YV).
\]

Therefore, the general strong rules endorse the use of this screening rule.

We do some experiments to compare the effectiveness of the rules. We simulate the model matrix under an independent design and an equi-correlated design with correlation \( \rho = 0.5 \). The exact solution path is computed using Algorithm 1 with several random initializations and the convex relaxation-based initialization (as in the Multi-Gaussian rule). Let \( S(\lambda) \) be the true active set at \( \lambda \). For each method \( \ell \) above, we can find, based on either the exact solution at \( \lambda_{k-1} \) or the full-rank solution at \( \lambda_k \), the threshold it needs so that by the screening criterion, the selected subset \( \hat{S}(\lambda_k)^{(\ell)} \) contains the true subset at \( \lambda_k \), i.e. \( \hat{S}(\lambda_k|\lambda_{k-1})^{(\ell)} \supset S(\lambda_k) \). This demonstrates how deep each method has to search down the variable list to include all necessary variables, and thus how accurate the screening mechanism is — the smaller the subset, the better the method. Fig. 2 shows an example scenario with problem size \( n = 200, p = 500, q = 20 \) and \( k = 20 \) nonzero rows of coefficients. The true rank is set to be 3 and signal-to-noise ratio (SNR) is 1. We compute the solution over 50 \( \lambda \) values equally spaced on the log-scale. Additional experiments are showed in Appendix D.1.
Fig 2: Size of screened set under different strategies. Left: independent design. Right: equi-correlated design with $\rho = 0.5$. Size: $n = 200, p = 500, q = 20$ and $k = 20$ nonzero rows of coefficients. Signal-to-noise ratio (SNR) = 1, and we use the true rank = 3.

We see from both plots that the curve of the Fix-V method is able to track that of the exact subset fairly well, while the Rank-Less and Multi-Gaussian methods both choose a much larger subset in order to cover the full active set in the exact solution. In the rest of the paper, we will adopt the Fix-V method to do variable screening.

3.2.2. Optimality Condition. Although the Fix-V method turns out to be most effective in choosing the subset of variables, in practice we have no access to the true subset and have to take an estimate. Instead of trying to find a sophisticated threshold, we will do batch screening at a fixed size (this size can change adaptively though). Given a size $M$, we will take the $M$ variables that rank the top under this criterion. Clearly we can make mistakes by having left out some important variables in the screening stage. In order to make sure that our solution is exact rather than approximate in terms of the original problem, we need to check the optimality condition and take in more variables when necessary.

Suppose we find a solution $\hat{U}_S, \hat{V}_S$ on a subset of variables $X_S$ by alternating minimization. We will verify the assembled solution $\hat{U} = (\hat{U}_S, 0), \hat{V} = \hat{V}_S$ is a limit point of the original optimization problem. The argument is supported by the following lemma.

**Lemma 3.3.** In the U-step (3.2), given $V$ and $\lambda$, if we have an exact solution $\hat{U}_S$ for the sub-problem with $X_S$, then $\hat{U} = (\hat{U}_S, 0)$ is a solution to the full problem if and only if for all $j \in S^c$,

$$\|x_j^\top (YV - X_S \hat{U}_S)\|_2 \leq \lambda.$$  

**Proof.** Since this is a convex problem, $\hat{U}$ is a solution if and only if $0 \in \partial f(\hat{U})$ where $f$ is the objective function in (3.2) and $\partial f$ is its subdifferential. For the vector $\ell_2$-norm, we
know that the subdifferential of $\|x\|_2$ is $\{s \in \mathbb{R}^p : \|s\|_2 \leq 1\}$ if $x = 0$ and $\{x/\|x\|_2\}$ if $x \neq 0$.

Notice that $X_S \hat{U}_S = X \hat{U}$ by the definition of $\hat{U}$. Since we have an exact solution on $S$, we know $0 \in \partial f(\hat{U})_j$ for all $j \in S$. On the other hand, for $j \in S^c$, $0 \in \partial f(\hat{U})$ if and only if $0 \in \{x_j^\top (X \hat{U} - YV) + \lambda s_j : \|s_j\|_2 \leq 1\}$, which is further equivalent to $\|x_j^\top (YV - X_S \hat{U}_S)\|_2 = \|x_j^\top (YV - X \hat{U})\|_2 \leq \lambda$.

Therefore, once we obtain a solution $\hat{U}_S, \hat{V}_S$ for the sub-problem and get condition (3.4) verified, we know in the V-step, by the lemma above, $\hat{U} = (\hat{U}_S, 0)$ is the solution given $\hat{V} = \hat{V}_S$. In the U-step, since $X \hat{U} = X_S \hat{U}_S$, $\hat{U}$ is the solution to the full problem. We see that $(\hat{U}, \hat{V})$ is a limiting point of the alternating minimization algorithm for the original problem. However if the condition fails, we expand the screened set or bring in the violated variables, and do the fit again. As mentioned earlier, we do not guarantee the final solution to be a local minimum or global minimum, unless under some regularity conditions. It is instead a limiting point of the vanilla alternating minimization algorithm, i.e. Algorithm 1. In other words, if we start from the constructed solution (with zero coefficients for the leftout variables), the algorithm should converge in one iteration and return the same solution.

We have seen the main ingredients of the iterative algorithm: screening, solving and checking. Next we discuss some useful practical considerations and extensions.

### 3.3. Computational Considerations.

#### 3.3.1. Initialization and Warm Start.
Recall that in the training stage our goal is to fit an SRRR solution path across different $\lambda$ values. It is easy to see that with a careful choice of parameterization, the path is continuous in $\lambda$. To leverage this property, we adopt a warm start strategy. Specifically, we initialize the coefficients of the existing variables at $\lambda_{k+1}$ using the solution at $\lambda_k$ and zero-initialize the newly added variables. With warm start, much less iterations will be needed to converge to the new minimum.

However, this by no means guarantees that we are all on a good path. It’s likely that we are trapped into a neighborhood of local optimum and end up with much higher function value than the global minimum. One way to alleviate this, if affordable, is to solve the corresponding full-rank problem first, and initialize the coefficients with low-rank approximation of the full-rank solution. We can compare the limiting function values with the warm-start initialization and see which converges to a better point. Although we didn’t use in the actual implementation and experiments, one could also do random exploration — randomly initialize some of the coefficients, run the algorithm multiple times and find one that achieves the lowest function value. That said, we lose the advantage of warm start though. The good news is, in the experiments we have done, we didn’t observe very clear suboptimal behavior by the warm start and full-rank strategies.

#### 3.3.2. Size of Expanded Variable Set.
As described above, when the KKT check fails, we expand the current screened set by adding more variables from the rest of the variables. In our framework, the number of variables to add is a hyperparameter subject to the user’s
choice. There is some computational tradeoff in determining the expansion size — the more
variables one adds each time, the less likely that the KKT check will fail so as to call for
another round of computation, but with a larger variable set, the per-round computation
also scales up.

3.3.3. Early Stopping. Although we pre-specify a sequence of $\lambda$ values $\lambda_1 > \lambda_2 > \cdots > \lambda_L$
where we want to fit the SRRR models, we do not have to fit them all given our goal is to
find the best predictive model. Once the model starts to overfit as we move down the $\lambda$ list,
we can stop our process since the later models will have no practical use and are expensive
to train. Therefore, in the actual computation, we monitor the validation error along the
solution path and call it a stop if it shows a clear upward trend. One other point we would
like to make in this regard is that the validation metric can be defined either as an average
MSE over all phenotypes or a subset of phenotypes we are most interested in. This is because
practically the best $\lambda$ value can be different for different phenotypes in the joint model.

3.4. Extensions. On top of the core algorithm, we introduce several extensions that are
lightweight but can be important for real-world applications, such as dealing with non-
uniform scaling of the responses, existence of confounders and missing data. In Appendix
A of the supplement, we also introduce a weighting mechanism that allows one to encode
priors on different responses.

3.4.1. Standardization. We often want to standardize the predictors if they are not on
the same scale because the penalty term is not invariant to change of units of the variables.
However we emphasize that some thought has to be put into this before standardizing the
predictors. If the predictors are already on the same scale, standardizing them could bring
unintended advantages to variables with smaller variance themselves. It is more reasonable
not to standardize in such cases.

In terms of the outcomes, since they can be at different scales, it is important to stan-
dardize them in the training stage so that no one dominates in the objective function. At
prediction (both training and test time), we scale back to the original levels using their
respective variances from the training set. In fact, the real impact an outcome has to the
overall objective is determined by the proportion of unexplained variance. It would be good
to weight the responses properly based on this if such information is available or can be
estimated, e.g. via heritability estimation for phenotypes in genetic studies.

3.4.2. Adjustment Covariates. In some applications such as genome-wide association
studies (GWAS), there may be confounding variables $Z \in \mathbb{R}^{n \times m}$ that we want to adjust
for in the model. For example, population stratification, defined as the existence of a sys-
tematic ancestry difference in the sample data, is one of the common factors in GWAS that
can lead to spurious discoveries. This can be controlled for by including some leading prin-
cipal components of the SNP matrix as variables in the regression (Price et al., 2006). In the
presence of such variables, we solve the following problem instead. With a slight abuse of
notation, in this section, we use $W$ to denote the coefficient matrix for the covariates instead of a weight matrix:

$$\minimize \frac{1}{2} \|Y - ZW - XUV^\top\|_F^2 + \lambda \sum_{j=1}^p \|U_j\|_2,$$

(3.5)

s.t. $V^\top V = I$.

The main components don’t change except two adjustments. When determining the starting $\lambda$ value, we use Lemma 3.4.

**Lemma 3.4.** In problem (3.5), if $r > 0$, the maximum $\lambda$ that results in a nontrivial solution $\hat{B}(\lambda)$ is

$$\lambda_{\text{max}} = \max_{1 \leq j \leq p} \|x_j^\top \hat{R}\|_2,$$

where $\hat{R} = Y - Z\hat{W}$ and $\hat{W}$ is the multiple outcome regression coefficient matrix.

The proof is almost the same as before. The other nuance we should be careful about is when fitting the model, we should leave those covariates unpenalized because they serve for the adjustment purpose and should not be experiencing the selection stage. In particular, in the U-step (group lasso) given $V$, direct computation would reduce to solving the problem

$$\minimize_{U, W} \frac{1}{2} \|YV - ZWV - XU\|_F^2 + \lambda \sum_{j=1}^p \|U_j\|_2,$$

which is not as convenient as standard group lasso problem. Instead, we find that $W$ can always be solved explicitly in terms of other variables. In fact, the minimizer of $W$ is a least squares solution that can be expressed as $\hat{W} = (Z^\top Z)^{-1}Z^\top(Y - XUV^\top)$. Plug in and we find that the problem to be solved can be written as

$$\minimize_U \frac{1}{2} \|(I - H_Z)YV - (I - H_Z)XU\|_F^2 + \lambda \sum_{j=1}^p \|U_j\|_2,$$

where $H_Z = Z(Z^\top Z)^{-1}Z^\top$ is the projection matrix on the column space of $Z$. This becomes a standard group lasso problem and can be solved by using, for example, the **glmnet** package with the **mgaussian** family.

### 3.4.3. Missing Values.

In practice, there can be missing values in either the predictor matrix or the outcome matrix. If we only discard samples that have any missing value, we could lose a lot of information. For the predictor matrix, we could do imputation as simple as mean imputation or something sophisticated by leveraging the correlation structure. For missingness in the outcome, there is a natural way to integrate an imputation step seamlessly with the current procedure, analogous to the softImpute idea in Mazumder, Hastie and
Tibshirani (2010). We first define a projection operator for a subset of two dimensional indices $\Omega \subseteq \{1, \ldots, n\} \times \{1, \ldots, p\}$. Let $\mathcal{P}_\Omega : \mathbb{R}^{n \times p} \to \mathbb{R}^{n \times p}$ be such that

$$
\mathcal{P}_\Omega(Y)_{i,j} = \begin{cases}
Y_{i,j}, & (i,j) \in \Omega, \\
0, & (i,j) \notin \Omega.
\end{cases}
$$

Let $\Omega$ be the set of indices where the response values are observed; in other words, $\Omega^c$ is the set of missing locations. Instead of (2.2), now we solve the following problem.

$$
\begin{aligned}
&\text{minimize} & & \frac{1}{2} \|\mathcal{P}_\Omega(Y) - \mathcal{P}_\Omega(XUV^T)\|_F^2 + \lambda \sum_{j=1}^p \|U_j\|_2, \\
&\text{s.t.} & & V^TV = I,
\end{aligned}
$$

(3.6)

We can easily see that an equivalent formulation of the problem is

$$
\begin{aligned}
&\text{minimize} & & \frac{1}{2} \|Y' - XUV^T\|_F^2 + \lambda \sum_{j=1}^p \|U_j\|_2, \\
&\text{s.t.} & & V^TV = I, \mathcal{P}_\Omega(Y') = \mathcal{P}_\Omega(Y).
\end{aligned}
$$

This inspires a natural projection step to deal with the additional constraint. It can be well integrated with the current alternating minimization scheme. In fact, after each alternation between the U-step and the V-step, we can impute the missing values from the current predictions $XUV^T$, and then continue into the next U-V alternation with the completed matrix. This only adds computation in each iteration at a low cost of $O(n|\mathcal{A}(\lambda)|rq)$, especially when the size of the screened set $|\mathcal{A}(\lambda)| \ll p$, which is common in practical high-dimensional applications.

3.4.4. Lazy Reduced Rank Regression. There is an alternative way to find a low-rank coefficient profile for the multivariate regression. Instead of pursing to solve the non-convex problem (2.2) directly, we can follow a two-stage procedure:

1. Solve a full-rank multi-gaussian sparse regression, i.e.,

$$
\text{minimize}_B \quad \frac{1}{2} \|Y - XB\|_F^2 + \lambda \sum_{j=1}^p \|B_j\|_2.
$$

2. Conduct SVD of the resulting coefficient matrix $\hat{B}$ and use its rank $r$ approximation as our final estimator.

The advantage of this approach is that it is stable. The first stage is a convex problem and can be handled efficiently by, for example, glmnet. A variety of adaptive screening rules are also applicable in this situation to assist dimension reduction. The second stage is fairly standard and efficient as long as there are not too many active variables. However, the disadvantage
is clear too. The low-rank approximation is conducted in an unsupervised manner, so could lead to some degrade in the prediction performance.

That said, as before, we should still evaluate the out-of-sample performance as the penalty parameter $\lambda$ varies and pick the best on the solution path as our final estimated model. In many cases, we compute the full-rank model under the exact mode anyways, so the set of lazy models can be thought of as an efficient byproduct for our choice.

3.5. Full Algorithm. We incorporate the options above and present the full algorithm in Algorithm 2.

3.6. Memory and Time Costs. Unlike most traditional procedures that need to operate on the full data, our SRRR algorithm takes advantage of sequential variable screening so that it restricts intensive computation on a small subset of the variables. This makes it possible for memory-efficient implementations and also allows one to handle larger-than-RAM data.

In our Algorithm 2, variable screening and KKT Check will each incur one pass over the full data (that can in fact be merged into a single pass). We don’t have to bring all into memory but can instead use techniques such as memory mapping to deal with it efficiently.

Since we will iterate over the candidate active set from screening multiple times, we often choose to explicitly load them into memory and this will be the most memory expensive part. We will need at most $O(n(|A| + M))$ space to save the set, where $A$ is the current active set and $M$ is the size of the additional variables to be brought in from screening.

That being said, in sparse ultrahigh-dimensional problems where we often expect $|A| \ll p$, our screening-based approach can greatly reduce the memory demand.

For each iteration, the computational complexity of variable screening is $O(npq + nr(|A(\lambda)| + q))$, or $O(npq)$ if $|A(\lambda)|, q \ll p$. However, the majority of this computation is independent for each variable and highly parallelizable. The actual computation time can be greatly reduced if we have access to multicore machines. The cost of alternating minimization depends on the actual number of iterations, which in turn depends on the initialization.

This is why warm start can really help in computing the solution path. Down to each alternation, it takes $O(nr(|A| + q) + qr^2 + \kappa n |A|r)$ to compute, assuming that we use coordinate descent and it takes $\kappa$ steps to solve the group lasso problem (as glmnet does). We note that $\kappa$ is often very small (e.g. 2-5) when we use warm start for the group lasso as well.

After taking into account the number of alternating iterations $K$, the overall computational cost for solving one cycle of variable screening, alternating minimization and KKT Check is $O(npq)$ (parallelizable) + $O(n|A|rq) + O(K(nr(|A| + q) + qr^2 + \kappa n |A|r))$. If we further assume $r, q \ll |A|, n$, this cost becomes $O(npq)$ (parallelizable) + $O((K \cdot \kappa + q)nr|A|)$.

Therefore, the total computational cost of the algorithm scales linearly in the (average) per-cycle cost, or $O(Rnpq)$ (parallelizable) + $O(R(K \cdot \kappa + q)nr|A|)$, depending on the potential number $R$ of such cycles that is further determined by the number of repetitions per $\lambda$ (due to KKT failure) and early stopping.
Algorithm 2 Large-scale and Ultrahigh-dimensional Sparse Reduced Rank Regression

1: Standardize or weight the responses. Define a sequence of \( \lambda \) values \( \lambda_{\max} = \lambda_0 > \lambda_1 > \cdots > \lambda_L \geq 0 \).
2: for \( \ell = 1 \) to \( L \) do
3: Initialize \( \ell = 0 \), \( U(\lambda_\ell) = U(\lambda_{\ell-1}), V(\lambda_\ell) = V(\lambda_{\ell-1}), W(\lambda_\ell) = W(\lambda_{\ell-1}) \), and \( A(\lambda_\ell) \) be the active set at \( \lambda_{\ell-1} \).
4: while \( \ell = 0 \) or KKT Check at \( \ell - 1 \) failed do
5: [Variable Screening] Find \( M \) variables \( S_M \subseteq \Omega \setminus A(\lambda_\ell) \) with largest values in \( \|x_j^T(Y - ZW(\lambda_\ell) - X_{A(\lambda_\ell)}U_{A(\lambda_\ell)}(\lambda_\ell)V(\lambda_\ell))^T\| \), and let
\[ A(\lambda_\ell) = A(\lambda_\ell) \cup S_M. \]
6: [Alternating Minimization] Let \( k = 0 \) and \( U^{(0)} = U_{A(\lambda_\ell)}(\lambda_\ell), V^{(0)} = V(\lambda_\ell), W^{(0)} = W(\lambda_\ell) \) and \( Y^{(0)} = Y \).
7: while \( k = 0 \) or \( \|U^{(k)}V^{(k)T} - U^{(k-1)}V^{(k-1)T}\| > \epsilon \) do
8: V-step: Fix \( U^{(k)} \), solve \( V \): the orthogonal Procrustes problem
\[ \text{minimize} \|Y^{(k)} - ZW^{(k)} - X_{A(\lambda_\ell)}U^{(k)}V^{(k)}\|^2_F. \]
Let \( (Y^{(k)} = ZW^{(k)})X_{A(\lambda_\ell)}U^{(k)} = MN^T \) ( skinny SVD) and solve \( V^{(k+1)} = MN^T \).
9: U-step: Fix \( V^{(k+1)} \), solve \( U \) and \( W \): the group lasso problem
(3.7) \[ U^{(k+1)} = \arg\min_U \frac{1}{2}\|I - H_Z\|Y^{(k)}V^{(k+1)} - (I - H_Z)X_{A(\lambda_\ell)}U\|^2_F + \lambda_\ell \sum_{j=1}^P \|U_j\|_2, \]
and \( W^{(k+1)} = (Z^TZ)^{-1}Z^T(Y^{(k)} - X_{A(\lambda_\ell)}U^{(k+1)}V^{(k+1)}) \).
10: Y-step: Impute the missing values
\[ Y^{(k+1)}_\Omega = Y^{(k)}_\Omega, \quad Y^{(k+1)}_{\Omega^c} = (ZW^{(k+1)} + X_{A(\lambda_\ell)}U^{(k+1)}V^{(k+1)})_{\Omega^c}. \]
11: \( k = k + 1 \)
12: end while
13: Let \( U_{A(\lambda_\ell)}(\lambda_\ell) = U^{(k)}, U_{A(\lambda_\ell)}(\lambda_\ell) = 0, V(\lambda_\ell) = V^{(k)}, W(\lambda_\ell) = W^{(k)} \) and \( Y = Y^{(k)} \).
14: [KKT Check] Check the criterion for all \( j \in \Omega \setminus A(\lambda_\ell), \)
\[ \|x_j^T(Y - ZW(\lambda_\ell) - X_{A(\lambda_\ell)}U_{A(\lambda_\ell)}(\lambda_\ell)V(\lambda_\ell))^T\| \leq \lambda_\ell. \]
15: \( t = t + 1 \)
16: end while
17: end for

4. Convergence Analysis. In this section, we present some convergence properties of the alternating minimization algorithm (Algorithm 1) on sparse reduced rank regression. Let
\[ g(U, V) = \frac{1}{2}\|Y - XUV^T\|^2_F + \lambda \sum_{j=1}^P \|U_j\|_2. \]
**THEOREM 4.1.** For any $k \geq 1$, the function values are monotonically decreasing:

$$g(U^k, V^k) \geq g(U^{k+1}, V^k) \geq g(U^{k+1}, V^{k+1}).$$

Furthermore, we have the following finite convergence rate:

$$\min_{1 \leq k \leq K^*} g(U^k, V^k) - g(U^{k+1}, V^{k+1}) \leq \frac{1}{K} (g(U^1, V^1) - g^\infty),$$

where $g^\infty = \lim_{k \to \infty} g(U^k, V^k)$. It implies that the iteration will terminate in $O(1/\epsilon)$ iterations.

The proof is straightforward and we won’t detail here. It presents the fact that alternating minimization is a descent algorithm. In fact, this property holds for all alternating minimization or more general blockwise coordinate descent algorithms. However it does not say how good the limiting point is. In the next result, we show a local convergence result that under some regularity conditions, if the initialization is closer enough to a global minimum, it will converge to a global minimum at linear rate. It is based on similar results on proximal gradient descent by Dubois, Delmas and Obozinski (2019). To define a local neighborhood, it would be easier if we eliminate $V$ by always setting it to a minimizer given $U$. That is, by the second part of Lemma 3.2, at the optimal of $V$, the cross-term (in a trace form) coming from the expansion of the squared difference can be reduced to a nuclear norm, and thus the objective function becomes $F(\lambda, U) = \frac{1}{2} \| XU \|^2 - \| Y^T XU \|_* + \lambda \sum_{j=1}^p \| U_j \|^2$. We define a sublevel set $S_c(\lambda) = \{U \in \mathbb{R}^{p \times r} : F(\lambda, U) \leq c\}$.

**THEOREM 4.2.** Assume $X^\top X$ is invertible and $\sigma^2_{\max} \geq \sigma^2_{\min} > 0$ be its smallest and largest eigenvalues. Let $s_j$ be the $j$th singular value of $(X^\top X)^{\top} \frac{1}{2} X^\top Y$. There exists $\bar{\lambda} > 0$ such that for all $0 \leq \lambda < \bar{\lambda}$ and $0 \leq \mu < \sigma^2_{\min}(1 - s^2_{r+1}/s^2_r)$, there is a sublevel set $S(\lambda, \mu)$ where the level depends on $\lambda$ and $\mu$ such that if $U^k \in S(\lambda, \mu)$, we have

$$\Delta(U^{k+1}, V^{k+1}) \leq \left(1 - \min \left(\frac{1}{2}, \frac{\mu}{\sigma^2_{\max}}\right)\right) \Delta(U^k, V^k),$$

where $\Delta(U, V) = g(U, V) - g(U^*, V^*)$ and $(U^*, V^*)$ is a global minimum.

From a high level, the proof is based on the fact that under the conditions, the function is strongly convex near the global minima. If we starting from this region, we achieve good convergence rate with alternating minimization algorithm. The full proof is given in Appendix B.3.

It is easy to see that the theorem above implicitly assumes the classical setting where $n \geq p$ since otherwise $X^\top X$ would not be invertible. However, it is still applicable to our algorithm. The algorithm does not attempt to solve alternating minimization at the full scale, but only does it after variable screening. With screening, it is very likely that we will again be working under the classical setting. Moreover, with warm start, there is higher chance that the initialization lies in the local region as defined above. Therefore, this theorem can provide useful guidance on the practical computational performance of the algorithm.
5. Simulation Studies. In this section, we study the predictive performance of SRRR and the computational performance of the proposed algorithm through simulations on some synthetic data.

5.1. Predictive Performance. We conduct some experiments to gain more insight into the method and compare with the single-response lasso method. Due to space limit, we demonstrate the results in one experiment setting and include results for other settings such as correlated features, deviation from the true low-rank structure etc., in Appendix D. We experiment with three different sizes and three different signal-to-noise ratio (SNR): \((n, p, k) = (200, 100, 20), (200, 500, 20), (200, 500, 50)\), where \(k\) is the number of variables with true nonzero coefficients, and the target SNR = 0.5, 1, or 3. The number of responses \(q = 20\) and the true rank \(r = 3\). We generate the \(X \in \mathbb{R}^{n \times p}\) with independent samples from some multivariate Gaussian \(\mathcal{N}(0, \Sigma_X)\) where \(\Sigma_X = I_p\) in this section. More results under correlated designs are presented in the appendix. The response is generated from the true model \(Y = XUV^\top + E\), where each entry in the support of \(U \in \mathbb{R}^{p \times r}\) (sparsity \(k\)) is independently drawn from a standard Gaussian distribution, and \(V \in \mathbb{R}^{q \times r}\) takes the left singular matrix of a Gaussian ensemble. Hence \(B = UV^\top\) is the true coefficient matrix. The noise matrix is generated from \(\mathcal{N}(0, \sigma_e^2 I_q)\), where \(\sigma_e^2\) is chosen such that the signal-to-noise ratio

\[
\text{SNR} = \frac{\text{Tr}(B^\top \Sigma_X B)}{\sigma_e^2 \cdot \text{Tr}(\Sigma_E)}
\]

is set to a given level. The performance is evaluated by the test \(R^2\), defined as follows:

\[
R^2 = 1 - \frac{\|Y - X\hat{B}\|_F^2}{\|Y - \bar{Y}\|_F^2},
\]

where \(\hat{B}\) is the fitted coefficient matrix and \(\bar{Y}\) is a mean matrix with the mean response vector of \(Y\) stacked across the rows. The main insight we obtain from the experiments is that the method is more robust to overestimating than underestimating the rank. A significant degrade in performance can be identified even if we are only off the rank by 1 from below. In contrast, the additional variance brought along by overestimating the rank doesn’t seem to be a big concern. This, in essence, can be ascribed to bias and variance decomposition. In our settings, the bias incurred in underestimating the rank and thus 1/3 loss of parameters contributes a lot more to the MSE compared with the increased variance due to 1/3 redundancy in the parameters.

5.2. Computational Performance. In the algorithm, screening can significantly save the computational cost by focusing intensive operations on a small subset of the variables and reduce the memory cost with the need of only loading in a subset of the variables by some designated implementation.

We implement the algorithm in our package \texttt{multiSnpnet} that uses memory mapping for partial loading of the data and parallel computation for variable screening and KKT check. Although the package targets at SNP data in a widely used format provided by PLINK 2.0
Fig 3: $R^2$ each run is evaluated on a test set of size 5000. “oracle” is the result where we know the true active variables and solve on this subset of variables. “glmnet” fits the responses separately. “SRRR-r” indicates the SRRR results with assumed rank $r$.

(Chang et al., 2015), the only difference from general application in terms of computation is its compact file format (due to finite, discrete SNP levels) that aims to reduce the memory cost.

The simulation is conducted with this package on randomly generated SNP data under different settings. The performance should be largely generalizable to normal numeric data under a similar implementation.

We fix the number of samples $n = 50K$ (50,000), the number of responses $q = 20$ but vary the number of variables $p$ from low dimensions to high dimensions: 10K, 20K, 50K, 100K, 200K. We further fix the sparsity ratio to 2%; that is, 2% of the rows of the coefficient matrix are nonzero. The overall noise level for all responses is chosen so that the signal-to-noise ratio (SNR) is 1. We assume different true ranks in the data generating process: 2, 5, 10.
and experiment with different SRRR ranks: 2, 5, 10 in the algorithm. The experiments are run with 16 cores and 32GB of memory.

In Fig 4, we measure the total computational time (top row) and as a key component of the algorithm, the average number of inner U-V iterations (bottom row) spent in solving a 50-λ solution path (with λ_{min}/λ_{max} = 0.1) versus the dimension p. Each column corresponds to a different true rank in data generation. Each line corresponds to a different assumed rank in the algorithm. We repeat each experiment 5 times and measure the average. For the top row, we see that overall the computational cost grows superlinearly in the dimension of the ambient space, which is consistent with our analysis in Section 3.6. For the bottom row, the iteration continues until the change in the objective value is less than 10^{-7} times the initial value or the number of iterations reaches 50.

Fig 4: Top: Computational time needed under different data dimensions. Bottom: Average number of inner iterations needed under different data dimensions.

We further compare the computational performance with other methods of solving the SRRR. In particular, we compare with the subgradient method proposed in Chen and Huang (2012), which is implemented (in C++) in the rrpack package (Chen, 2019). Since the implementation does not focus on large-scale problems, it requires full loading of the data into memory and only adopts sequential (non-parallel) computation. This leads to longer computational time, and for this reason, we run experiments on relatively smaller-scale problems. The data generating process is the same as that for Fig 4. Moreover, for fair comparison, we constrain to use a single core in multiSnpnet to do sequential computation (but also include multi-core performance for reference).
For $n = 1000, p = 5000$, 4GB of memory is needed by rrpack, while less than 1GB is needed by multiSnpnet. For the larger case $n = 10000, p = 20000$, we use 32GB of memory for both methods.

### Table 1
Comparison of computational time by different SRRR methods (implementations). Time is all in minutes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time $(n = 1000, p = 5000)$</th>
<th>Time $(n = 10000, p = 20000)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subgradient method (Chen and Huang, 2012)</td>
<td>3.476</td>
<td>$&gt; 24 \times 60$</td>
</tr>
<tr>
<td>MultiSnpnet (single core)</td>
<td>1.419</td>
<td>21.247</td>
</tr>
<tr>
<td>MultiSnpnet (16 cores)</td>
<td>1.306</td>
<td>13.381</td>
</tr>
</tbody>
</table>

We see that the time needed by the subgradient method increases drastically as the dimension increases. It has to go through every variable in each iteration regardless of the sparsity and by Chen and Huang (2012), the per-iteration complexity grows quadratically in the dimension $p$. Moreover, the first-order method tends to have slower convergence rate and the practical implication can be prominent in large-scale and high-dimensional settings.

### 6. Real Data Application: UK Biobank
The UK Biobank (Bycroft et al., 2018) is a large, prospective population-based cohort study with individuals collected from multiple sites across the United Kingdom. It contains extensive genetic and phenotypic detail such as genome-wide genotyping, questionnaires and physical measures for a wide range of health-related outcomes for over 500,000 participants, who were aged 40-69 years when recruited in 2006-2010. In our application, we consider the relationship between an individual’s genotype and his/her phenotypic outcomes. Building on a recent line of work (Qian et al., 2020; Sinnott-Armstrong et al., 2021; Lello et al., 2018) that fits a lasso solution on the large dataset and correlation structures in phenotypes, some of which are driven by a shared genetic basis, we hypothesize joint modeling for multiple outcomes would improve the prediction and stabilize the variable selection.

We focus on 337,199 White British unrelated individuals in the UK Biobank (Bycroft et al., 2018) that satisfy the same set of population stratification criteria as in DeBoever et al. (2018). Each individual has up to 805,426 measured variants, and each variant is encoded by one of the four levels where 0 corresponds to homozygous major alleles, 1 to heterozygous alleles, 2 to homozygous minor alleles and NA to a missing genotype. We consider age, sex and the top 10 pre-computed principal components (PCs) of the SNP matrix to adjust for population stratification (Price et al., 2006). Given the large sample size, we randomly partition the full data so that 70% is used for training, 10% for validation and 20% for test. The solution path is fit on the training set, the desired regularization is selected on the validation set, and the final model is evaluated on the test set.

In the experiment, we compare the performance of the multivariate-response SRRR model with the single-response lasso model, which we rely on fast implementation of the snpnet package (Qian et al., 2020) and also refer to as snpnet in the results section. For continuous responses, we evaluate the prediction by R-squared ($R^2$), defined in (5.2) but we compute it
for each single phenotype separately. There are binary responses in the data such as many
disease outcomes. Although in principle we can solve for a mixture of Gaussian and binomial
likelihood using Newton’s method, for ease of computation in this large-scale setting, it is a
reasonable approximation to treat them as continuous responses and fit the standard SRRR
model. However, after the model is fit, we will refit a logistic regression on the predicted
score to obtain a probability estimation. Notice that the refit is still trained on the training
set at each $\lambda$ value. For binary responses, we evaluate the area under the receiver operating
characteristic curve (AUC-ROC). When comparing different methods, we also evaluate the
absolute change and relative change over the baseline method (in particular the already
competitive lasso in our case), where the relative change for a given metric is defined as
$$\frac{\text{metric}_{\text{new}} - \text{metric}_{\text{lasso}}}{|\text{metric}_{\text{lasso}}|}.$$  

Computationally, in the UK Biobank experiments, the SNP data are stored in a com-
pressed PLINK format with two-bit encodings. PLINK 2.0 (Chang et al., 2015) provides an
extensive set of efficient data operations. In particular, its very fast, multithreaded matrix
multiplication module has been heavily used in the screening and KKT check steps in this
work and other lasso-based results (Li et al., 2020; Qian et al., 2020) on the UK Biobank.

Here we present the experimental results on 35 biomarkers in the main text. In Appendix
E of the supplement, we have additional results for asthma and 7 blood and respiratory
biomarkers as well as detailed description of the biplot visualization and the scores used in
biological interpretation of the results.

![Fig 5: Alanine aminotransferase and albumin prediction performance plots. Different colors and shapes correspond to lower rank predictive performance across (x-axis) training data set and (y-axis) validation data set for (left) alanine aminotransferase and (right) albumin. For lower rank representation we applied lazy rank evaluation.](image-url)
Fig 6: Change in prediction accuracy for multiresponse model compared to single response model. The bars (y-axis 1) indicate $R^2$ relative change (%) $(R_{mr}^2 - R_{single}^2)/R_{single}^2$ for each phenotype (x-axis) and the black dots (y-axis 2) indicate $R^2$ absolute change $R_{mr}^2 - R_{single}^2$. Top: $R^2$ change for each biomarker across different biomarker categories (color). Bottom: $R^2$ change in predictive accuracy for multiresponse model compared to single response model for urinary biomarkers. A color version of the figure is presented in the online version of the paper.
We apply SRRR to 35 biomarkers from the UK Biobank biomarker panel in Sinnott-Armstrong et al. (2021). Noticeably, for the liver biomarkers including alanine aminotransferase and albumin, and the urinary biomarkers including Microalbumin in urine and Sodium in urine, we see an improvement in prediction performance for the SRRR application beyond the single-response snpnet models (see Figures 5 and 6).

To gain biological insights on the identified genotype-phenotype relationship, we consider the biplot representation of the SVD of coefficient matrix (Gower, Lubbe and Le Roux, 2011; Gabriel, 1971; Tanigawa et al., 2019) with a specific focus on AST to ALT ratio, an important biomarker for liver disease. In our comparative experiment against single-response model implemented in snpnet, the relative increment of predictive performance was modest for AST to ALT ratio (Figure 6). Nonetheless, we find the multi-response model offers better interpretation compared to the single-response model.

To focus on the phenotype of our interest, we re-ranked the latent components on the basis of the relative importance of each component for AST to ALT ratio using the phenotype squared cosine score as described in Tanigawa et al. (2019). We identified components 9, 18, 20, 8, and 3 as the top five components of importance and investigated the phenotypes driving each component (Figure 7).

We find the genetics of AST to ALT ratio can be decomposed into components, where component 9 is driven by the genetics of total protein and non-albumin protein, component 18 is driven by the genetics of vitamin D, and component 20 is driven by the genetics of AST to ALT ratio, vitamin D, and aspartate aminotransferase. The biplot representation of those five components (Figure 8) provides a way to interpret the genetic variants contributing to the prediction of AST to ALT ratio in terms of components. For example, protein-altering variants in TNFRSF13B and MAP1A (rs34562254 and rs55707100, respectively) influence on increasing total protein, non-albumin protein, and AST to ALT ratio, whereas a protein-altering variant rs4588 in GC gene, which encodes GC vitamin D binding protein, has vitamin D-lowering effect and AST to ALT ratio-increasing effect. These results highlight the benefit of joint modeling of multiple responses with sparse predictive models in the interpretation of features.

7. Related Work. There are many other methods that were proposed for multivariate regression in high-dimensional settings. As mentioned earlier, Chen and Huang (2012) extended the Reduced Rank Regression (Anderson, 1951) to sparse and low-rank scenarios and proposed the Sparse Reduced Rank Regression (SRRR). They compared the SRRR with rank-free methods including $L_2$SVS (Similä and Tikka, 2007), $L_\infty$SVS (Turlach, Venables and Wright, 2005) that replaces the $\ell_2$-norm with $\ell_\infty$-norm of each row, and RemMap (Peng et al., 2010) that imposes an additional elementwise sparsity of the coefficient matrix. They also compared SRRR with Sparse Partial Least Squares (SPLS) (Chun and Keleş, 2010) — an important extension of the classical multivariate regression method Partial Least Squares (Wold, 1966) to high-dimensional settings — and pointed out that the latter does not target directly on prediction of the responses and thus often impairs the performance. Canonical Correlation Analysis (CCA) (Hotelling, 1936) that tries to constructed uncorrelated components in both the feature space and the response space to maximize their correlation coefficients also falls short in the aspect, even though some connection can be established with
Fig 7: Latent components of SRRR regression coefficients offers interpretation of the genetics of AST to ALT ratio. The top five key components for the genetics of AST to ALT ratio are identified with the phenotype squared cosine score (Tanigawa et al., 2019) and shown with the relative importance of phenotypes for each component. A color version of the figure is presented in the online version of the paper.

The reduced rank regression as seen in Appendix C. In addition, an important related multivariate method Independent Component Analysis (ICA) (Comon, 1994; Jutten and Herault, 1991) tries to identify latent components based on the independence and non-Gaussianity assumptions. It is fairly useful in blind source separation by maximizing the independence among different sources (for example, see the survey article by Hyvärinen and Oja (2000)). We do not assume independence among different predictors and also aim for the predictive performance, and therefore ICA most likely won’t serve well for our purpose.

More recently, there is a line of new advances in sparse and low-rank regression problems. For example, Ma and Sun (2014) proposed a subspace assisted regression with row sparsity and studied its near-optimal estimation properties. Ma, Ma and Sun (2020) furthered this work to a two-way sparsity setting, where nonzero entries are present only on a few rows and columns. Li, Liu and Chen (2019) proposed an integrative multi-view reduced-rank regression that encourages group-wise low-rank coefficient matrices with a composite nuclear norm. Dubois, Delmas and Obozinski (2019) developed a fast first-order proximal gradient...
Fig 8: The latent structures of the top five key SRRR components for AST to ALT ratio. Using trait squared cosine score described in Tanigawa et al. (2019), the top five key SRRR components for AST to ALT ratio (components 9, 18, 20, 8, and 3) are identified from a full-rank SVD of coefficient matrix $\hat{\beta}$ from SRRR ($\hat{\beta} = UDV^T$) and shown as a series of biplots. In each panel, principal components of genetic variants (rows of $UD$) are shown in blue as scatter plot using the main axis and singular vectors of traits (rows of $V$) are shown in red dots with lines using the secondary axis, for the identified key components. The five traits and variants with the largest distance from the center of origin are annotated with their name. A color version of the figure is presented in the online version of the paper.

algorithm on the SRRR objective reparameterized by a single matrix and proves linear local convergence. Luo et al. (2018) proposed a mixed-outcome reduced-rank regression method.
that deals with different types of responses and also missing data, though it does not aim for high-dimensional settings with variable selection.

There is also a collection of successful applications based on SRRR and its variants. Shen and Thompson (2020) summarized some of the recent SRRR applications in brain imaging genomics. In particular, Vounou, Nichols and Montana (2010) and Vounou et al. (2012) studied rank-one SRRR models with \( \ell_1 \) penalty terms for feature selection, where the latter exploited a penalized linear discriminant analysis (LDA) step at the beginning to filter the responses to predict for and introduces a stability model selection approach by data re-sampling. Silver et al. (2012a) proposes a pathways SRRR model that can encode pathway information with the group lasso penalty, an idea developed earlier in Silver et al. (2012b) for a single trait. In Zhu et al. (2016), the authors proposed a structured SRRR (S-SRRR) that penalizes both components of the decomposed coefficient matrix in a row-wise group manner that selects both the features and the responses. Zhu et al. (2017) extended S-SRRR to graph regularized S-SRRR (GRS-SRRR) by integrating a self-supervised model that tried to capture a sparse internal correlation structure between the SNPs. This is further followed up with a robust modification by Zhu, Zhang and Fan (2018). However none of them solves problems of the same size as we target at — both large-scale and ultrahigh-dimensional.

In these applications, the dimension can be as high as hundreds of thousands, but the number of samples is fairly limited, at most 1000s, which is also common in traditional genetic studies. To facilitate computation, some of them propose algorithms to solve instead an approximation of the target problem, such as using the identity matrix to replace the nontrivial covariance structure \( X^T X \) (Vounou, Nichols and Montana, 2010; Vounou et al., 2012). This allows for analytic expression and thus fast computation when solving the \( \ell_1 \)-penalized objective in the iteration.

Among them, Silver et al. (2012a) proposed an alternating scheme and adopts an active-set strategy suggested in Silver et al. (2012b) for the group lasso step, similar to our proposal in this paper, though they only focus on the rank-one model, a special case of our formulation. However it uses the active-set screening purely for the group-lasso step, and in its proposed iterative scheme (and also for ours), multiple iterations are often needed until convergence for any single \( \lambda \) value, e.g. see Figure 4. Therefore it needs at least the same number of screening rounds, each of which can be very expensive for large-scale and ultrahigh-dimensional problems as it would involve a full pass over the entire data. In contrast, the screening strategy we propose works at the \( \lambda \) value so that we don’t have to do that repeatedly in the inner iterations that involves the orthogonal Procrustes and the group-lasso steps, which can effectively reduce the number of full-data operations.

In genetics, DeGAs (Tanigawa et al., 2019) and MetaPhat (Lin et al., 2020) were proposed to decompose genetic associations from summary level data using LD-pruning along with \( p \)-value thresholding for variable selection. DeGAs was extended for genetic risk prediction and to "paint" an individual’s risk to a disease based on genetic component loadings in an approach referred to as DeGAs-risk (Aguirre et al., 2021).

8. Summary and Discussion. In this paper, we propose a framework to solve large-scale and ultrahigh-dimensional sparse reduced rank regression (SRRR) problems that
encourage both sparsity and low-rank structures when multiple correlated outcomes are present. An alternating minimization algorithm with specially designed screening scheme is developed for large-scale and ultrahigh-dimensional applications such as the UK Biobank population cohort. We demonstrate the effectiveness of the method on both synthetic and real datasets focusing on the 35 biomarker panel as well as asthma and 7 related blood and pulmonary biomarkers made available by UK Biobank (Sinnott-Armstrong et al., 2021). We show that the joint predictive modeling of multiple response improves the predictive performance for some phenotypes and that the low-rank structure and the sparsity of the coefficient matrix offers biological interpretation of the results. We anticipate that the approach presented here will generalize to thousands of phenotypes being measured in UK Biobank, e.g. metabolomics and imaging data that are currently being generated in over 100,000 individuals. For genetics studies, we develop an R package multiSnpnet available at https://github.com/junyangq/multiSnpnet that contains efficient implementation of the proposed framework for data in the PLINK2 format (Chang et al., 2015).

Although the 35 biomarkers in our main experiment are all continuous, there are many phenotypes in the UK Biobank or other biological studies that are binary, such as asthma in Appendix E of the supplement. For the sake of efficient computation, we use continuous approximation to these outcomes. This is reasonable in large-scale studies but ideally one would like to solve the exact problem based on their respective likelihood. In principle, there is no theoretical challenge in the algorithmic design. We can use Newton’s method and enclose the procedure with an outer loop that conducts quadratic approximation of the objective function. However, the quadratic problem involving both penalty and low-rank constraint can be very messy. We might need some heuristics to find a more convenient approximation. We see this as future work along with extending the SRRR algorithm to other families including time-to-event multiple responses that can be used for survival analysis. Furthermore, for an individual we can project a variant and phenotype loading across the reduced rank to their risk to arrive at a similar analysis of outlier individuals with unusual painting of genetic risk and to quantify the overall contribution of a component which may aid in disease risk interpretation.

In Section 3.4.4, we propose lazy SRRR as a convenient alternative to capturing some low-rank structure in the coefficient profiles, and that in our empirical studies (for example, Figure 5), it leads to almost equivalent predictive performances to the ones obtained with the exact alternated scheme. However, it is not a guarantee for all cases. As mentioned briefly in Section 3.4.4, the components are found in a “unsupervised” way given the full-rank coefficient matrix and it is not hard to imagine cases where this can miss some of the signals that would have been captured by the standard SRRR. In addition, the lazy SRRR requires one to solve the full-rank problem in the first place, which could be extravagant when the true rank is much smaller than the number of responses. That being said, the lazy scheme is a nice alternative due to the stability in the training process. There is no issue of local minimum as in the standard SRRR. It can be a byproduct of the full-rank model if we can afford to compute that.

Overall, we see the method and algorithms presented here as an important toolkit to large-scale and ultrahigh-dimensional multivariate regression problems. The design principles of
the framework, such as specialized screening scheme and the integration of missing value
imputation, do not limit to SRRR itself and can in fact generalize to variants of SRRR
in the literature as described above as well as more general classes of problems to inspire
solutions for large-scale data problems.

**Acknowledgement.** This research has been conducted using the UK Biobank Resource
under Application Number 24983, “Generating effective therapeutic hypotheses from ge-
nomic and hospital linkage data” (http://www.ukbiobank.ac.uk/wp-content/uploads/
2017/06/24983-Dr-Manuel-Rivas.pdf). Based on the information provided in Protocol
44532 the Stanford IRB has determined that the research does not involve human subjects
as defined in 45 CFR 46.102(f) or 21 CFR 50.3(g). All participants of UK Biobank provided
written informed consent (more information is available at https://www.ukbiobank.ac.
uk/2018/02/gdpr/). We thank all the participants in the UK Biobank. M.A.R. is supported
by Stanford University, a National Institutes of Health (NIH) Center for Multi-
ethnic Mapping of Mendelian and Complex Diseases grant (5U01 HG009080) and grant
1R01MH121455-01. M.A.R. also wants to thank the Sanofi IDEA award. Y.T. is supported
by a Funai Overseas Scholarship from the Funai Foundation for Information Technology and
the Stanford University School of Medicine. Research reported in this publication was sup-
ported by the National Human Genome Research Institute of the NIH under Award Number
R01HG010140 (M.A.R.). The content is solely the responsibility of the authors and does not
necessarily represent the official views of the NIH. R.T. was partially supported by NIH
grant 5R01 EB001988-16 and NSF grant 19 DMS1208164. T.H. was partially supported by
grant DMS-1407548 from the National Science Foundation, and grant 5R01 EB 001988-21
from the National Institutes of Health. We want to thank the editor, an associate editor,
and two anonymous reviewers from the journal for their constructive comments that helped
us improve the manuscript significantly.

**REFERENCES**

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S.,
Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G.,
**TensorFlow**: A System for Large-scale Machine Learning, In *Proceedings of the 12th USENIX Confer-
ence on Operating Systems Design and Implementation, OSDI’16* 265–283. USENIX Association, Berke-
ley, CA, USA.

Aguirre, M., Tanigawa, Y., Venkataraman, G. R., Tibshirani, R., Hastie, T. and Rivas, M. A.
Genetics.*

Anderson, T. W. (1951). Estimating Linear Restrictions on Regression Coefficients for Multivariate Nor-

Learning Research* 9 1179–1225.

Bickel, P. J., Ritov, Y. and Tsybakov, A. B. (2009). Simultaneous analysis of Lasso and Dantzig

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMP-
STAT’2010* 177–186. Springer.


LARGE-SCALE MULTIVARIATE SPARSE REGRESSION


SUPPLEMENT TO “LARGE-SCALE MULTIVARIATE SPARSE REGRESSION WITH APPLICATIONS TO UK BIOBANK”

Junyang Qian, Yosuke Tanigawa, Ruilin Li,
Robert Tibshirani, Manuel A. Rivas, Trevor Hastie

APPENDIX A: EXTENSION OF PREDICTOR AND RESPONSE WEIGHTS

Sometimes we have strong reasons or evidence to prioritize some of the predictors than the others. We can easily extend the standard objective (2.2) and reflect this belief in a weighted penalty $\lambda \sum_{j=1}^{P} w_j \| U_j \|_2$ where the weight $w_j$ controls inversely the relative importance of the $j$th variable. For example, $w_j = 0$ implies $j$th variable will always be included in the model, while a large $w_j$ will almost exclude the variable from the model.

In the response space, we can also impose a weighting mechanism to prioritize the training of certain responses. For a given set of nonnegative weights $w_k, 1 \leq k \leq q$, the SRRR objective (2.2) can be modified to $(1/2) \sum_{k=1}^{q} w_k \| Y_k - XUV_k^\top \|_2^2 + \lambda \sum_{j=1}^{P} \| U_j \|_2$ with the same constraint, or equivalently,

$$\minimize \frac{1}{2} \| YW^{1/2} - XUV^\top \|_F^2 + \lambda \sum_{j=1}^{P} \| U_j \|_2,$$

$$\text{s.t.} \quad V^\top W^{-1} V = I,$$

where the weight matrix $W = \text{diag}(w_1, \ldots, w_q)$. To solve the problem with our alternating minimization scheme, we can see that in the V-step, instead of solving the standard orthogonal Procrustes problem with an elegant analytic solution derived from the SVD, we have to deal with a so-called weighted orthogonal Procrustes problem (WOPP). Finding the solution of the WOPP is far more complicated. See, for instance, Mooijaart and Commandeur (1990), Chu and Trendafilov (1998) and Viklands (2006). An iterative procedure is often needed to compute the solution. For better computational efficiency, we instead solve the problem with the original orthonormal constraint:

$$\minimize \frac{1}{2} \| YW^{1/2} - XUV^\top \|_F^2 + \lambda \sum_{j=1}^{P} \| U_j \|_2,$$

$$\text{s.t.} \quad V^\top V = I.$$

That is, we amplify the magnitude of some responses so that the objective value is more sensitive to the loss incurred on these responses. When making prediction, we will need to scale them back to the original units.
APPENDIX B: ADDITIONAL PROOFS

B.1. Proof of Lemma 3.1. This is intuitively the same as one without the rank constraint because when the coefficients just start to become nonzero, the coefficient matrix is low-rank in its nature; that is, the group-lasso penalty will yield a special solution path that turns on the coefficient matrix from all zero in a row-wise manner as \( \lambda \) decreases through \( \lambda_{\text{max}} \) and as a result, the rank of the coefficient matrix will in general change in a continuous way (on the set \( \mathbb{N} \) of natural numbers). Therefore, for the purpose of finding the maximum meaningful \( \lambda \), we can ignore the rank constraint unless \( r = 0 \). Without the constraint, it follows from the KKT condition that having all coefficients to be zero is equivalent to

\[
\lambda \geq \lambda_{\text{max}} = \max_{1 \leq j \leq p} \| x_j^T Y \|_2.
\]

Therefore, the maximum \( \lambda \) that accommodates a nontrivial solution is \( \lambda_{\text{max}} = \max_{1 \leq j \leq p} \| x_j^T Y \|_2 \).

B.2. Proof of Lemma 3.2. We plug in the SVD of \( Z \) and have

\[
\text{Tr}(Z^T V) = \text{Tr}(NDM^T V) = \text{Tr}(DM^T VN) = \sum_{k=1}^r D_{kk} S_{kk},
\]

where \( S = M^T VN \) and the last equality is due to the fact that \( D \) is a diagonal matrix. Notice that by the SVD, \( M^T M = I \) and \( (VN)^T (VN) = I \). Hence,

\[
|S_{kk}| = |\langle M_j, (VN)_{j} \rangle| \leq \|M_j\|_2 \|(VN)_{j}\|_2 = 1.
\]

Since \( D_{kk} \) are all non-negative, to maximize \( \sum_{k=1}^r D_{kk} S_{kk} \), we should let \( S_{kk} = \) 1 for all \( k \) where \( D_{kk} > 0 \). Therefore, one solution is given by \( M = VN \) and so \( V = MN^T \). The maximum value of the objective is thus \( \sum_{k=1}^r D_{kk} = \|Z\|_*, \) the nuclear norm of \( Z \).

When \( Z \) has full rank, we know \( D_{kk} > 0 \) for all \( 1 \leq k \leq r \). The maximizer must satisfy \( M = VN \) and so the solution is unique.

B.3. Proof of Theorem 4.2. We notice that in Problem (2.2) we can solve explicitly for \( V \) and plug back into the objective function. It yields the objective function (after dropping the constant term \( (1/2) \|Y\|_F^2 \)):

\[
F_\lambda(U) = \frac{1}{2} \|XU\|_2^2 - \|Y^T XU\|_* + \lambda \sum_{j=1}^p \|U_j\|_2,
\]

We let \( f_\lambda(U) = (1/2) \|XU\|_2^2 - \|Y^T XU\|_* \) without the penalty term so that \( F_\lambda(U) = f_\lambda(U) + \lambda \sum_{j=1}^p \|U_j\|_2 \). Define a local smooth approximation of \( F_\lambda \) as

\[
\bar{F}_\lambda(U'; U) = f_\lambda(U) + \langle \nabla f_\lambda(U), U' - U \rangle + (1/2t) \|U' - U\|_F^2 + \lambda \sum_{j=1}^p \|U_j\|_2,
\]
and $\mathbf{U}^+ = \text{argmin}_{\mathbf{U}} [F^t_{\lambda}(\mathbf{U}'; \mathbf{U}) - F_{\lambda}(\mathbf{U})]$. Dubois, Delmas and Obozinski (2019) showed that if $t$ is small enough such that $F^t_{\lambda}(\mathbf{U}^+; \mathbf{U}) \geq F_{\lambda}(\mathbf{U}^+)$, we have

\begin{equation}
F_{\lambda}(\mathbf{U}^+) - F^*_\lambda \leq \left(1 - \min \left(\frac{1}{2}, \mu t\right)\right) (F_{\lambda}(\mathbf{U}) - F^*_\lambda).
\end{equation}

Consider the iterates $(\mathbf{U}^k, \mathbf{V}^k)_{k \geq 1}$ in the alternating minimization algorithm. Notice that $
abla f_{\lambda}(\mathbf{U}^k) = \mathbf{X}^T \mathbf{X} \mathbf{U}^k - \mathbf{X}^T \mathbf{Y} \mathbf{V}^k$. We have

\begin{align*}
F_{\lambda}(\mathbf{U}^{k+1}) &= g(\mathbf{U}^{k+1}, \mathbf{V}^{k+1}) - \frac{1}{2} \| \mathbf{Y} \|^2_F \quad (g \text{ is the SRRR objective function}) \\
&\leq g(\mathbf{U}^{k+1}, \mathbf{V}^k) - \frac{1}{2} \| \mathbf{Y} \|^2_F \\
&= \min_{\mathbf{U}} g(\mathbf{U}, \mathbf{V}^k) - \frac{1}{2} \| \mathbf{Y} \|^2_F \\
&= \min_{\mathbf{U}} \left( \frac{1}{2} \| \mathbf{Y} - \mathbf{X} \mathbf{U}^k (\mathbf{V}^k)^T \|^2_F + \langle \mathbf{X}^T (\mathbf{X} \mathbf{U}^k - \mathbf{Y} \mathbf{V}^k), \mathbf{U} - \mathbf{U}^k \rangle + \frac{1}{2} \text{Tr}((\mathbf{U} - \mathbf{U}^k)^T \mathbf{X}^T \mathbf{X}(\mathbf{U} - \mathbf{U}^k)) \right) \\
&\quad + \lambda \sum_{j=1}^p \| \mathbf{U}_j \|_2 - \frac{1}{2} \| \mathbf{Y} \|^2_F \\
&\leq \min_{\mathbf{U}} \left( f_{\lambda}(\mathbf{U}^k) + \langle \nabla f_{\lambda}(\mathbf{U}), \mathbf{U}' - \mathbf{U} \rangle + \frac{1}{2} \sigma^2_{\max} \| \mathbf{U} - \mathbf{U}^k \|^2_F \right) + \lambda \sum_{j=1}^p \| \mathbf{U}_j \|_2 \\
&= \min_{\mathbf{U}} F_{\lambda}^{1/\sigma^2_{\max}}(\mathbf{U}; \mathbf{U}^k),
\end{align*}

where the fourth line is the quadratic expansion of $g(\mathbf{U}, \mathbf{V}^k)$ at $\mathbf{U}^k$, the second to last is by the fact that $\text{Tr}((\mathbf{U} - \mathbf{U}^k)^T \mathbf{X}^T \mathbf{X}(\mathbf{U} - \mathbf{U}^k)) \leq \sigma^2_{\max} \| \mathbf{U} - \mathbf{U}^k \|^2_F$, and the last equality is by the definition of $F_{\lambda}^t$ function. Therefore, if we let $\mathbf{U}^{k,+} = \text{argmin}_{\mathbf{U}} [F_{\lambda}^{1/\sigma^2_{\max}}(\mathbf{U}; \mathbf{U}^k) - F_{\lambda}(\mathbf{U}^k)]$, we have

\begin{equation}
F_{\lambda}(\mathbf{U}^{k+1}) - F^*_\lambda \leq F_{\lambda}(\mathbf{U}^{k,+}) - F^*_\lambda.
\end{equation}

We need to show that $\mathbf{U}^{k,+}$ satisfies the condition $F_{\lambda}^{1/\sigma^2_{\max}}(\mathbf{U}^{k,+}; \mathbf{U}^k) \geq F_{\lambda}(\mathbf{U}^{k,+})$. To see this, notice that in fact for any $\mathbf{U}$,

\begin{align*}
\frac{1}{2} \| \mathbf{X} \mathbf{U} \|^2_F &= \frac{1}{2} \| \mathbf{X} \mathbf{U}^k \|^2_F + \langle \mathbf{X}^T \mathbf{X} \mathbf{U}^k, \mathbf{U} - \mathbf{U}^k \rangle + \frac{1}{2} \| \mathbf{X}(\mathbf{U} - \mathbf{U}^k) \|^2_F \\
&\leq \frac{1}{2} \| \mathbf{X} \mathbf{U}^k \|^2_F + \langle \mathbf{X}^T \mathbf{X} \mathbf{U}^k, \mathbf{U} - \mathbf{U}^k \rangle + \frac{1}{2} \sigma^2_{\max} \| \mathbf{U} - \mathbf{U}^k \|^2_F.
\end{align*}

Since $\mathbf{X}^T \mathbf{Y} \mathbf{V}^k$ is a subgradient of $\| \mathbf{Y}^T \mathbf{X} \mathbf{U} \|$ at $\mathbf{U}^k$, we have

\begin{equation*}
-\| \mathbf{Y}^T \mathbf{X} \mathbf{U} \|_\ast \leq -\| \mathbf{Y}^T \mathbf{X} \mathbf{U}^k \|_\ast - \langle \mathbf{X}^T \mathbf{Y} \mathbf{V}^k, \mathbf{U} - \mathbf{U}^k \rangle.
\end{equation*}
Adding the two inequalities up, and we have $F_\lambda(U) \leq \tilde{F}_\lambda^{1/\sigma^2_{\max}}(U; U^k)$ for all $U$. In particular, it holds for $U^{k+}$. Therefore, by (B.2) and (B.3), we have

$$F_\lambda(U^{k+1}) - F_\lambda(U^k) \leq \left( 1 - \min \left( \frac{1}{2}, \frac{\mu}{\sigma^2_{\max}} \right) \right) (F_\lambda(U^k) - F_\lambda^*)$$

and the convergence is linear.

APPENDIX C: CONNECTION WITH CCA

Canonical Correlation Analysis (CCA) has an internal connection with Reduced-Rank Regression (RRR). In particular, it can be shown that the low-rank components constructed on the $X$ space turn out to be the same by a relaxed CCA and a generalized RRR. CCA finds linear combinations $XU \in \mathbb{R}^{n \times r}$ of variables in $X \in \mathbb{R}^{n \times p}$ and linear combinations $YV \in \mathbb{R}^{n \times r}$ of variables in $Y \in \mathbb{R}^{n \times q}$ that attain the maximum correlation. We assume both $X$ and $Y$ have been centered. CCA solves the following optimization problem:

$$\max_{U,V} \quad \text{Tr}(U^T X^T YV),$$

s.t. $U^T X^T XU = V^T Y^T YV = I_r$. (C.1)

In particular, in the one dimensional case, this reduces to the problem of maximizing our familiar correlation coefficient. An equivalent representation to (C.1) can be written as

$$\min_{U,V} \quad \|YV - XU\|_F^2,$$

s.t. $U^T X^T XU = V^T Y^T YV = I_r$. (C.2)

The solution to the problem is $\hat{U} = S^{-1/2}_{xx} Q^{(r)}$, $\hat{V} = S^{-1/2}_{yy} P^{(r)}$, where $P^{(r)}$ and $Q^{(r)}$ are the $r$ leading left and right singular vectors of matrix $R = S^{-1/2}_{yy} S_{yy} S^{-1/2}_{xx}$. $P^{(r)}$ is also the $r$ leading eigenvectors of $S^{-1/2}_{yy} S_{xx} S^{-1/2}_{yy} S_{xy} S^{-1/2}_{yy}$. A relaxed form of CCA problem ignoring the $U$-constraint solves

$$\min_{U,V} \quad \|YV - XU\|_F^2,$$

s.t. $V^T Y^T YV = I_r$. (C.3)

The solution is $\hat{U} = S^{-1/2}_{xx} S_{xy} S^{-1/2}_{yy} P^{(r)}$, $\hat{V} = S^{-1/2}_{yy} P^{(r)}$, where $P^{(r)}$ is the $r$ leading eigenvectors of $S^{-1/2}_{yy} S_{xx} S^{-1/2}_{xy} S_{yy} S^{-1/2}_{yy}$. Therefore, the solution for $V$ remains unchanged, though $U$ is different due to the constraint.

On the other hand, in the (generalized) reduced rank regression, given a given positive-definite matrix $\Gamma$, the problem becomes

$$\min_{U,V} \quad \text{Tr}(\Gamma^{1/2} (Y - XUV^T)^T (Y - XUV^T) \Gamma^{1/2}).$$
This can be derived, for example, as an maximum likelihood estimator under the Gaussian assumption with known covariance $\Gamma^{-1}$. One solution (Velu and Reinsel, 2013) is given by

$$
\hat{U} = S_{xx}^{-1} S_{xy} \Gamma^{1/2} P^{(r)}, \\
\hat{V} = \Gamma^{-1/2} P^{(r)},
$$

where $P^{(r)}$ is the leading eigenvectors of $R = \Gamma^{1/2} S_{yx} S_{xx}^{-1} S_{xy} \Gamma^{1/2}$. We see that the solution when $\Gamma = S_{yy}^{-1}$ is closely related to the relaxed CCA solution. $U$ is the same while $V$ is the so-called reflexive inverse of $V$ there.

APPENDIX D: ADDITIONAL NUMERICAL EXPERIMENTS

D.1. Screening Rules. We show additional experiments comparing different screening strategies as in Fig 2. We fix the number of samples $n = 200$ and the number of responses $q = 20$.

Suppose we base on the solution at $\lambda_k$ and have an estimate $\hat{S}(\lambda_k, m)$ of the true active set $S(\lambda_k+1)$ at $\lambda_k+1$ by including $m$ candidate variables, we compute the minimal fraction of excess selection

$$
f = \min \left\{ \frac{|\hat{S}(\lambda_k, m)| - |S(\lambda_k+1)|}{|\hat{S}(\lambda_k+1)|} : m \geq |S(\lambda_k+1)|, S(\lambda_k+1) \subseteq \hat{S}(\lambda_k, m) \right\}.
$$

For different screening rules, we make the estimate by searching down the variable set ordered by the screening rule based on the current solution until each variable of the true active set at the next $\lambda$ has been included. This evaluates the number of redundant variables to be selected in order to identify the true active set and can help assess the quality of screening by different rules. However, we note that since the SRRR problem is nonconvex, it is hard to verify if a solution is a global minimizer. To approximate that, we use the multi-gaussian solution (which is convex) at the corresponding $\lambda$ as initialization and run the alternating minimization procedure until convergence. As we see in the experiments in Appendix D.2, this initialization is effective in finding a reasonable solution.

In Table 2-5, we evaluate the excess selection criterion above over 100 $\lambda$’s equally spaced on the log-scale on $[0.01 \lambda_{\text{max}}, \lambda_{\text{max}}]$. Each table corresponds to a different combination of correlation structure and SNR. Each row corresponds to a different combination of problem size, support size and the true rank. We repeat the experiment under each setting 30 times, and report the average and the associated standard error in the parentheses under the average for three specific $\lambda$ values on the solution path ($\lambda_{25}, \lambda_{50}, \lambda_{75}$). We also show the size of the true active set (we expect to include) in the “Exact” column. Here, R-L stands for Rank-Less, F-V for Fix-V, M-G for Multi-Gaussian as seen in Fig. 2.
<table>
<thead>
<tr>
<th>$p$</th>
<th>$k$</th>
<th>$r$</th>
<th>$\lambda_{15}$</th>
<th>$\lambda_{50}$</th>
<th>$\lambda_{75}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10</td>
<td>3</td>
<td>7.9 (0.05) 0 11.57 (0.2)</td>
<td>22.9 (1) 3.4 (0.18) 0 (0)</td>
<td>85.3 (0.9) 0.18 (0.01) 0 (0.01)</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>5</td>
<td>8.9 (0.09) 0 10.27 (0.03) 0 (0.32)</td>
<td>41.4 (2.2) 1.61 (0.16) 0 (0.17)</td>
<td>97.4 (0.4) 0.03 (0) 0 (0)</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>10</td>
<td>9.8 (0.09) 0 9.08 (0.09) 0 (0.09)</td>
<td>85.7 (1.4) 0.17 (0.02) 0.01 (0.01)</td>
<td>100 (0) 0 0 0</td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>3</td>
<td>18.7 (0.43) 0 4.54 (0.7) 0 (0)</td>
<td>53.6 (0.3) 0.85 (0) 0 (0.3)</td>
<td>93.5 (0.9) 0.07 (0) 0.07 (0.01)</td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>5</td>
<td>25.5 (0.53) 0.02 3.19 (0.04) 0 (0.02)</td>
<td>76.2 (1.4) 0.32 (0.05) 0 (0.05)</td>
<td>99.4 (0) 0.01 0 0.01 (0.01)</td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>10</td>
<td>30.7 (0.52) 0 2.31 (0.8) 0 (0.08)</td>
<td>97.5 (0.3) 0.03 (0) 0 (0)</td>
<td>100 (0) 0 0 0</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>3</td>
<td>8.4 (0.04) 0.02 21.75 (0.05) 0 (0.84)</td>
<td>34.2 (1.8) 5.06 (0.33) 0 (0.37)</td>
<td>149.7 (1.6) 0.34 (0) 0.34 (0)</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>5</td>
<td>8.9 (0.2) 0 20.54 (0.09) 0 (0.48)</td>
<td>75.7 (3.7) 1.83 (0.19) 0.15 (0.23)</td>
<td>185 (1) 0.08 0 0.08 (0.01)</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>10</td>
<td>9.9 (0.12) 0 18.59 (0.05) 0 (0.13)</td>
<td>155.3 (2.7) 0.29 (0.02) 0.02 (0.03)</td>
<td>199.5 (0.1) 0 0 0</td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>3</td>
<td>21 (0.64) 0 8.92 (0.9) 0 (0.45)</td>
<td>80.6 (2.7) 1.52 (0.09) 0.02 (0.09)</td>
<td>170 (1.2) 0.18 0 0.18 (0.01)</td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>5</td>
<td>24.9 (0.85) 0 7.15 (0.9) 0 (0.25)</td>
<td>124.5 (2.6) 0.61 (0.04) 0.01 (0.04)</td>
<td>193.2 (0.4) 0.04 0 0.04 (0)</td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>10</td>
<td>36.5 (0.97) 0 4.78 (0.8) 0 (0.26)</td>
<td>183.5 (1.6) 0.09 (0) 0 (0.01)</td>
<td>200 (0) 0 0 0</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>3</td>
<td>21.5 (1.33) 0.01 1.59 (1) 0 (0.19)</td>
<td>127.4 (4.4) 2.96 (0.16) 0 (0.11)</td>
<td>268 (2.9) 0.87 (0) 0.81 (0)</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>5</td>
<td>28.9 (2.27) 0 1.96 (1.2) 0 (0.15)</td>
<td>211 (4.6) 1.37 (0.06) 0.06 (0.04)</td>
<td>352.7 (1.7) 0.42 (0) 0.38 (0)</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>10</td>
<td>50.2 (1.97) 0.04 1.31 (2.7) 0 (0.01)</td>
<td>352.9 (3.3) 0.41 (0.01) 0 (0.01)</td>
<td>444.4 (1.2) 0.12 0 0.1 (0)</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>3</td>
<td>35.5 (2.19) 0.01 2.29 (1.6) 0 (0.12)</td>
<td>166.1 (3.2) 2 (0.06) 0 (0)</td>
<td>285.9 (2.1) 0.75 0.01 0.71 (0)</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>5</td>
<td>55 (2.76) 0.11 2.11 (2.3) 0 (0.09)</td>
<td>254.5 (3.8) 0.97 (0.03) 0 (0.02)</td>
<td>365.3 (1.8) 0.37 0.01 0.34 (0)</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>10</td>
<td>104.7 (1.72) 0.11 0.97 (3.7) 0 (0.06)</td>
<td>376.3 (2.5) 0.33 (0) 0 (0.01)</td>
<td>447.8 (1.1) 0.12 0 0.09 (0)</td>
</tr>
</tbody>
</table>
Table 3
Fraction of excess selection under different screening rules. $\text{Cov}(X) = I$ and $\text{SNR} = 2$.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$k$</th>
<th>$r$</th>
<th>$\lambda_{25}$</th>
<th>$\lambda_{50}$</th>
<th>$\lambda_{75}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Exact</td>
<td>R-L</td>
<td>F-V</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>3</td>
<td>8 (0)</td>
<td>0 (0.25)</td>
<td>2.63 (0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9.1 (0.01)</td>
<td>0 (0.33)</td>
<td>2.89 (0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9.7 (0.1)</td>
<td>0 (0.33)</td>
<td>4.02 (0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>19.5 (0.01)</td>
<td>0 (0)</td>
<td>3.06 (0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>24.2 (0.01)</td>
<td>0 (0.08)</td>
<td>2.6 (0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>27.9 (0.01)</td>
<td>0 (0.04)</td>
<td>2.29 (0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8.5 (0)</td>
<td>0 (0.47)</td>
<td>4.4 (0)</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>5</td>
<td>9.1 (0.01)</td>
<td>0 (0.6)</td>
<td>6.62 (0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9.8 (0)</td>
<td>0 (0.5)</td>
<td>7.53 (0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20.7 (0.01)</td>
<td>0 (0.22)</td>
<td>5.51 (0)</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>10</td>
<td>24.3 (0.02)</td>
<td>0 (0.12)</td>
<td>5.18 (0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>28.4 (0.01)</td>
<td>0 (0.04)</td>
<td>4.82 (0)</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>3</td>
<td>20.5 (0.01)</td>
<td>0 (0.08)</td>
<td>0.03 (0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>24.9 (0.03)</td>
<td>0 (0.01)</td>
<td>0.03 (0)</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>3</td>
<td>32.4 (0.1)</td>
<td>0.01 (0.12)</td>
<td>0.12 (0)</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>5</td>
<td>41.6 (0.14)</td>
<td>0.01 (0.2)</td>
<td>0.13 (0)</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>10</td>
<td>49.7 (0.13)</td>
<td>0 (0.01)</td>
<td>0.11 (0)</td>
</tr>
</tbody>
</table>
Table 4
Fraction of excess selection under different screening rules. Cov(X) equi-correlated and SNR = 0.5.

<table>
<thead>
<tr>
<th></th>
<th>k</th>
<th>r</th>
<th>(\lambda_{25}) Exact</th>
<th>R-L</th>
<th>F-V</th>
<th>M-G</th>
<th>(\lambda_{50}) Exact</th>
<th>R-L</th>
<th>F-V</th>
<th>M-G</th>
<th>(\lambda_{75}) Exact</th>
<th>R-L</th>
<th>F-V</th>
<th>M-G</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10</td>
<td>3</td>
<td>5.1 (0.04) 0.2 (0.73) 8.19 (0.03) 0.03 6.24 (0.03) 0.2 (0.45) 8.73 (2.1) (0.07) (0) (0.07) 55.6 (0.86) 0.01 0.87</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>5</td>
<td>7.1 (0.03) 0.3 (0.54) 6.09 (0.03) 0.03 5.19 (0.02) 0.2 (0.03) 6.73 (1.4) (0.02) (0) (0.02) 81.5 (0.24) 0 0.24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>10</td>
<td>8.7 (0.02) 0.03 5.05 (0.18) 0.02 5.23 (0.09) 0.1 (0.23) 8.19 (0.3) (0) (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>3</td>
<td>10.7 (0.04) 0.01 5.97 (0.08) 0.01 5.19 (0.01) 0.2 (0.01) 8.19 (0.3) (0) (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>3</td>
<td>6.4 (0.07) 0 (1.1) 5.59 (0.01) 0.01 5.24 (0.01) 0.9 (0.01) 5.24 (3.5) (0.07) (0) (0.07) 96.8 (1.14) 0.02 1.14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>5</td>
<td>7.2 (0.05) 0.03 8.92 (0.36) 0.01 12.13 (0.01) 0.78 (0.4) (0.04) (0) (0.4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>10</td>
<td>8.5 (0.01) 0.14 10.01 (0.11) 0.09 4.35 (1.2) (0.01) (0) (0.01)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>3</td>
<td>10.7 (0.06) 0.01 9.88 (0.22) 0.01 5.91 (3.4) (0.05) (0) (0.05)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>5</td>
<td>12.7 (0.06) 0 (0.6) 12.13 (0.11) 0.02 4.64 (1.7) (0.01) (0) (0.01)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>10</td>
<td>14 (0.04) 0.01 7.35 (0.11) 0.09 2.57 (3.4) (0.05) (0) (0.05)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>3</td>
<td>12 (0.19) 0.02 0.43 (3.8) (0.34) 0.01 4.08 (5.7) (0.07) (0) (0.06)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>5</td>
<td>15.7 (0.16) 0.03 0.48 (6.9) (0.22) 0.02 2.69 (263.8) 0.91 0.04 0.78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>10</td>
<td>17.5 (0.05) 0.3 0.31 (5.2) (0.07) 0.06 3.27 (3.3) (0.01) (0) (0.01)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>3</td>
<td>17.9 (0.07) 0.02 0.26 (4) (0.22) 0.01 1.15 (196.5) 1.55 0.04 1.38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>5</td>
<td>18.8 (0.07) 0.01 0.42 (3.4) (0.16) 0.03 2.48 (268.6) 0.87 0.05 0.74</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>10</td>
<td>19.8 (0.05) 0.04 0.41 (10.4) (0.08) 0.02 0.21 (3.5) (0.01) (0) (0.01)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p$</td>
<td>$k$</td>
<td>$r$</td>
<td>$\lambda_{\text{L}}$</td>
<td>$\lambda_{\text{F-V}}$</td>
<td>$\lambda_{\text{M-G}}$</td>
<td>$\lambda_{\text{L}}$</td>
<td>$\lambda_{\text{F-V}}$</td>
<td>$\lambda_{\text{M-G}}$</td>
<td>$\lambda_{\text{L}}$</td>
<td>$\lambda_{\text{F-V}}$</td>
<td>$\lambda_{\text{M-G}}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>3</td>
<td>5.4 (0.3)</td>
<td>0.02 (0.02)</td>
<td>0.92 (0.01)</td>
<td>9.1 (0.2)</td>
<td>0.16 (0.05)</td>
<td>10.07 (0.23)</td>
<td>20.5 (1.3)</td>
<td>3.96 (0.28)</td>
<td>1.44 (0.32)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>5</td>
<td>6.9 (0.3)</td>
<td>0.04 (0.01)</td>
<td>0.62 (0.01)</td>
<td>9.9 (0.2)</td>
<td>0.15 (0.04)</td>
<td>9.15 (0.14)</td>
<td>34.9 (2.3)</td>
<td>2.13 (0.19)</td>
<td>2.22 (0.2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>10</td>
<td>8.2 (0.3)</td>
<td>0.01 (0.01)</td>
<td>0.35 (0.01)</td>
<td>10.2 (0.1)</td>
<td>0.11 (0.04)</td>
<td>8.81 (0.11)</td>
<td>70 (2.3)</td>
<td>0.44 (0.04)</td>
<td>0.47 (0.05)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>3</td>
<td>11.2 (0.5)</td>
<td>0.04 (0.02)</td>
<td>1.57 (0.01)</td>
<td>23.7 (0.8)</td>
<td>0.3 (0.02)</td>
<td>3.37 (0.15)</td>
<td>54.9 (1.7)</td>
<td>1.12 (0.06)</td>
<td>1.25 (0.07)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>5</td>
<td>12.5 (0.5)</td>
<td>0.1 (0.02)</td>
<td>1.31 (0.01)</td>
<td>25.6 (0.6)</td>
<td>0.16 (0.02)</td>
<td>2.96 (0.09)</td>
<td>54.8 (0.2)</td>
<td>0.8 (0.06)</td>
<td>0.9 (0.08)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>10</td>
<td>13.7 (0.6)</td>
<td>0.06 (0.01)</td>
<td>1.3 (0.1)</td>
<td>28.6 (0.4)</td>
<td>0.1 (0.02)</td>
<td>2.51 (0.05)</td>
<td>81.4 (1.9)</td>
<td>0.23 (0.03)</td>
<td>0.25 (0.04)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>3</td>
<td>6.4 (0.4)</td>
<td>0.03 (0.01)</td>
<td>0.96 (0.15)</td>
<td>9.8 (0.2)</td>
<td>0.5 (0.02)</td>
<td>19.84 (0.61)</td>
<td>32.5 (2.9)</td>
<td>5.7 (0.45)</td>
<td>6.32 (0.5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>5</td>
<td>6.8 (0.4)</td>
<td>0.04 (0.02)</td>
<td>0.74 (0.1)</td>
<td>9.9 (0.2)</td>
<td>0.3 (0.01)</td>
<td>19.43 (0.41)</td>
<td>48.7 (3.8)</td>
<td>0.48 (0.13)</td>
<td>3.74 (0.31)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>10</td>
<td>7.8 (0.6)</td>
<td>0.02 (0.02)</td>
<td>0.46 (0.01)</td>
<td>10.3 (0.3)</td>
<td>0.18 (0.05)</td>
<td>18.62 (0.27)</td>
<td>123.2 (3.4)</td>
<td>0.66 (0.07)</td>
<td>0.71 (0.08)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>3</td>
<td>11.2 (0.6)</td>
<td>0.05 (0.01)</td>
<td>2 (0.24)</td>
<td>24.1 (0.8)</td>
<td>0.57 (0.07)</td>
<td>7.58 (0.27)</td>
<td>61.9 (3.4)</td>
<td>2.25 (0.13)</td>
<td>2.46 (0.15)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>5</td>
<td>13.4 (0.6)</td>
<td>0.09 (0.02)</td>
<td>1.34 (0.1)</td>
<td>26.6 (0.8)</td>
<td>0.25 (0.07)</td>
<td>6.61 (0.17)</td>
<td>81.6 (2.1)</td>
<td>1.4 (0.13)</td>
<td>1.5 (0.07)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>10</td>
<td>14.4 (0.7)</td>
<td>0.05 (0.01)</td>
<td>1.31 (0.1)</td>
<td>29.4 (0.4)</td>
<td>0.18 (0.02)</td>
<td>6.03 (0.26)</td>
<td>137.9 (4.5)</td>
<td>0.45 (0.04)</td>
<td>0.51 (0.07)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>3</td>
<td>11.8 (0.8)</td>
<td>0.09 (0.02)</td>
<td>0.03 (0.02)</td>
<td>26.8 (1.3)</td>
<td>0.92 (0.27)</td>
<td>0.02 (0.15)</td>
<td>95.6 (6.2)</td>
<td>4.38 (0.25)</td>
<td>3.16 (0.16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>5</td>
<td>15.3 (0.8)</td>
<td>0.13 (0.02)</td>
<td>0.05 (0.01)</td>
<td>31.7 (1.5)</td>
<td>0.95 (0.28)</td>
<td>0.02 (0.13)</td>
<td>138.1 (6.5)</td>
<td>2.67 (0.25)</td>
<td>1.86 (0.09)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>10</td>
<td>17.4 (1.4)</td>
<td>0.12 (0.02)</td>
<td>0.07 (0.02)</td>
<td>30.7 (1.8)</td>
<td>0.35 (0.03)</td>
<td>0.02 (0.02)</td>
<td>234.5 (6.5)</td>
<td>1.06 (0.14)</td>
<td>0.56 (0.07)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>3</td>
<td>18.1 (1.4)</td>
<td>0.16 (0.02)</td>
<td>0.05 (0.01)</td>
<td>39.8 (1.5)</td>
<td>0.76 (0.13)</td>
<td>0.02 (0.02)</td>
<td>122.3 (5.5)</td>
<td>2.97 (0.13)</td>
<td>2.14 (0.07)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>5</td>
<td>18.2 (1.4)</td>
<td>0.15 (0.02)</td>
<td>0.03 (0.01)</td>
<td>42 (1.5)</td>
<td>0.54 (0.15)</td>
<td>0.03 (0.03)</td>
<td>156.5 (4.4)</td>
<td>2.11 (0.08)</td>
<td>1.44 (0.04)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>10</td>
<td>21.7 (1.3)</td>
<td>0.15 (0.02)</td>
<td>0.09 (0.01)</td>
<td>51.7 (1.5)</td>
<td>0.54 (0.06)</td>
<td>0.02 (0.03)</td>
<td>260.6 (4.4)</td>
<td>0.88 (0.08)</td>
<td>0.48 (0.04)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
D.2. Initialization Schemes. We know for nonconvex problems, initialization is important for the convergence to a good solution. This clearly applies to SRRR, unless the coefficient matrix is assumed to be full rank. In this section, we do further investigation into the convergence qualities and compare different initialization schemes. Similar to Appendix D.1, we have different tables corresponding to different combinations of correlation structure and SNR, and each row corresponds to a different combination of problem size, support size and the true rank. The schemes under comparison include:

- Full-rank (FR): For each $\lambda$ value, we initialize the iteration by the solution to the full-rank multi-gaussian solution at the same $\lambda$ value.
- Warm-start (WM): We initialize from the SRRR solution (could be local minimum) at the previous $\lambda$ value.
- Random (R): For each $\lambda$ value, we initialize the entire $U$ matrix with i.i.d. standard gaussian entries and solve the SRRR. This is repeated 5 times and the solution with minimum objective value is used.
- Random Subset (RS): For each $\lambda$ value, we only initialize the $U$ matrix with i.i.d. standard gaussian entries on the active set derived from the solution at the previous $\lambda$ value and then solve the SRRR at the current $\lambda$ value. This is repeated 5 times and the solution with minimum objective value is used.

To compare the quality of the solutions, we use the following metric:

$$f = \frac{F_\lambda(\hat{B}) - F_\lambda(\hat{B}_{\min})}{F_\lambda(\hat{B}_{\min})},$$

where $F_\lambda$ is the SRRR objective (2.1) and $\hat{B}_{\min}$ is the solution that achieves the minimum among these four approaches. We repeat the experiment under each setting 30 times, and report the average and the associated standard error in the parentheses under the average for three specific $\lambda$ values on the solution path ($\lambda_{25}, \lambda_{50}, \lambda_{75}$). We also show the (average) minimum objective value $F_\lambda(\hat{B}_{\min})$ in the “Min” column.
Table 6
Fraction of suboptimality under different initialization schemes. \( \text{Cov}(X) = I \) and \( \text{SNR} = 0.5 \).

<table>
<thead>
<tr>
<th>( p )</th>
<th>( k )</th>
<th>( r )</th>
<th>( \lambda_{25} )</th>
<th>( \lambda_{50} )</th>
<th>( \lambda_{75} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Min FR WM R RS</td>
<td>Min FR WM R RS</td>
<td>Min FR WM R RS</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>3</td>
<td>796.6 0 0.02 0 0</td>
<td>669.1 0 0 0 0</td>
<td>599.7 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(59.8) (0) 0.01 (0) (0)</td>
<td>(50) (0) (0) (0) (0)</td>
<td>(44.9) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>1285.5 0 0 0 0</td>
<td>1066.9 0 0 0 0</td>
<td>928 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(53.6) (0) 0 (0) (0)</td>
<td>(44.6) (0) (0) (0) (0)</td>
<td>(38.7) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>2617.8 0 0 0 0</td>
<td>2098.2 0 0 0 0</td>
<td>1677.9 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(123.4) (0) 0 (0) (0)</td>
<td>(100) (0) (0) (0) (0)</td>
<td>(79.1) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>3</td>
<td>2580.7 0 0.02 0 0</td>
<td>2132.7 0 0 0 0</td>
<td>1869.5 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(131.7) (0) 0.01 (0) (0)</td>
<td>(108.4) (0) (0) (0) (0)</td>
<td>(95.2) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>5</td>
<td>4291 0 0 0 0</td>
<td>3488.7 0 0 0 0</td>
<td>2996.7 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(185.7) (0) 0 (0) (0)</td>
<td>(150.2) (0) (0) (0) (0)</td>
<td>(128.8) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>10</td>
<td>8100.4 0 0 0 0</td>
<td>6225 0 0 0 0</td>
<td>4987.4 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(278.9) (0) (0) (0) (0)</td>
<td>(214.1) (0) (0) (0) (0)</td>
<td>(169.1) (0) (0) (0) (0)</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>3</td>
<td>866.1 0 0.01 0 0</td>
<td>726.4 0 0 0 0</td>
<td>641.4 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(44) (0) 0.01 (0) (0)</td>
<td>(37.1) (0) (0) (0) (0)</td>
<td>(32.8) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>5</td>
<td>1342.9 0 0.01 0 0</td>
<td>1112.4 0 0 0 0</td>
<td>936.7 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(55.1) (0) 0.01 (0) (0)</td>
<td>(45.9) (0) (0) (0) (0)</td>
<td>(39) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>2700.3 0 0 0 0</td>
<td>2123.2 0 0 0 0</td>
<td>1536.3 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(95) (0) (0) (0) (0)</td>
<td>(76.6) (0) (0) (0) (0)</td>
<td>(56.3) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>3</td>
<td>2488.8 0 0.01 0 0</td>
<td>2040.8 0 0 0 0</td>
<td>1763.1 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(97.2) (0) 0.01 (0) (0)</td>
<td>(80.9) (0) (0) (0) (0)</td>
<td>(69.3) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>5</td>
<td>4336.3 0 0 0 0</td>
<td>3498.8 0 0 0 0</td>
<td>2889.6 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(136.8) (0) 0 (0) (0)</td>
<td>(111) (0) (0) (0) (0)</td>
<td>(90.2) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>10</td>
<td>8195.5 0 0 0 0</td>
<td>6052.7 0 0 0 0</td>
<td>4322.8 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(218.1) (0) (0) (0) (0)</td>
<td>(165.2) (0) (0) (0) (0)</td>
<td>(118.3) (0) (0) (0) (0)</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>3</td>
<td>2433.6 0 0.01 0 0</td>
<td>1993.1 0 0 0 0</td>
<td>1693.5 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(98) (0) 0.01 (0) (0)</td>
<td>(80.8) (0) (0) (0) (0)</td>
<td>(68.4) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>5</td>
<td>4161 0 0 0 0</td>
<td>3308.2 0 0 0 0</td>
<td>2649.2 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(146.7) (0) (0) (0) (0)</td>
<td>(119.4) (0) (0) (0) (0)</td>
<td>(93.7) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>10</td>
<td>8048.8 0 0 0 0</td>
<td>5647.1 0 0 0 0</td>
<td>3784 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(236.6) (0) (0) (0) (0)</td>
<td>(172.3) (0) (0) (0) (0)</td>
<td>(115.3) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3</td>
<td>4268.4 0 0.01 0 0</td>
<td>3466.1 0 0 0 0</td>
<td>2921.5 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(194.4) (0) 0.01 (0) (0)</td>
<td>(156.5) (0) (0) (0) (0)</td>
<td>(131) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>5</td>
<td>7054.1 0 0 0 0</td>
<td>5513.9 0 0 0 0</td>
<td>4413.7 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(230.8) (0) (0) (0) (0)</td>
<td>(184.2) (0) (0) (0) (0)</td>
<td>(145.1) (0) (0) (0) (0)</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>10</td>
<td>13110.8 0 0 0 0</td>
<td>8925.7 0 0 0 0</td>
<td>5998.7 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(316.5) (0) (0) (0) (0)</td>
<td>(221.4) (0) (0) (0) (0)</td>
<td>(146.6) (0) (0) (0) (0)</td>
</tr>
<tr>
<td>( p )</td>
<td>( k )</td>
<td>( r )</td>
<td>( \lambda_{25} )</td>
<td>( \lambda_{50} )</td>
<td>( \lambda_{75} )</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>3</td>
<td>328.1</td>
<td>212</td>
<td>158.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(21.9)</td>
<td>(14.3)</td>
<td>(10.6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>5</td>
<td>568.5</td>
<td>364.6</td>
<td>272.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(31.9)</td>
<td>(20.3)</td>
<td>(15.1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>10</td>
<td>1152.7</td>
<td>729.8</td>
<td>514.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(58.3)</td>
<td>(36.9)</td>
<td>(25.9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>3</td>
<td>1185</td>
<td>768.2</td>
<td>556</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(51.8)</td>
<td>(33)</td>
<td>(23.6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>5</td>
<td>2006.6</td>
<td>1282.6</td>
<td>919.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(81.9)</td>
<td>(51.6)</td>
<td>(36.4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>10</td>
<td>3905.2</td>
<td>2422.8</td>
<td>1633</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(117.7)</td>
<td>(72.1)</td>
<td>(46.8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>3</td>
<td>392.6</td>
<td>255</td>
<td>190.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(21.5)</td>
<td>(14.1)</td>
<td>(10.6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>5</td>
<td>596.1</td>
<td>383.5</td>
<td>282.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(20)</td>
<td>(16.7)</td>
<td>(12.3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>10</td>
<td>1206.4</td>
<td>763.5</td>
<td>516</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(34.9)</td>
<td>(22)</td>
<td>(14.9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>3</td>
<td>1145.2</td>
<td>745.4</td>
<td>532.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(40.3)</td>
<td>(27.1)</td>
<td>(19.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>5</td>
<td>2003.6</td>
<td>1289.8</td>
<td>903.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(84.7)</td>
<td>(56.1)</td>
<td>(38.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>10</td>
<td>3726</td>
<td>2309.2</td>
<td>1459.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(117.7)</td>
<td>(65.5)</td>
<td>(40.7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>3</td>
<td>1054.4</td>
<td>685.1</td>
<td>488.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(54.4)</td>
<td>(35.5)</td>
<td>(25.1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>5</td>
<td>1857.1</td>
<td>1187.8</td>
<td>808.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(69.9)</td>
<td>(44.7)</td>
<td>(29.2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>10</td>
<td>3776.4</td>
<td>2312</td>
<td>1375.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(114.3)</td>
<td>(70)</td>
<td>(41.7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>3</td>
<td>2040.6</td>
<td>1327.5</td>
<td>918.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(75.1)</td>
<td>(50.8)</td>
<td>(34.7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>5</td>
<td>3212.8</td>
<td>2037.7</td>
<td>1366.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(124.6)</td>
<td>(80.1)</td>
<td>(52.3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>10</td>
<td>6123.4</td>
<td>3663.4</td>
<td>2131.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(137.6)</td>
<td>(84.1)</td>
<td>(47.8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
</tbody>
</table>
### Table 8

Fraction of suboptimality under different initialization schemes. $\text{Cov}(X)$ equi-correlated and SNR = 0.5.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$k$</th>
<th>$r$</th>
<th>$\lambda_25$</th>
<th>$\lambda_50$</th>
<th>$\lambda_75$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>FR</td>
<td>WM</td>
<td>R</td>
<td>RS</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
<td>827.3</td>
<td>0</td>
<td>0.05</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>10</td>
<td>1339.4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>10</td>
<td>2454.6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>10</td>
<td>2820.4</td>
<td>0</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>30</td>
<td>843.7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>30</td>
<td>1339.4</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>30</td>
<td>2454.6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>30</td>
<td>2820.4</td>
<td>0</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>30</td>
<td>843.7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>3</td>
<td>850.3</td>
<td>0</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>10</td>
<td>1297</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>10</td>
<td>2432.9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>30</td>
<td>2593.6</td>
<td>0</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>30</td>
<td>126.8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>30</td>
<td>2593.6</td>
<td>0</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>30</td>
<td>126.8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>3</td>
<td>2176.2</td>
<td>0</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>30</td>
<td>4053.7</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>30</td>
<td>8843.4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>30</td>
<td>126.8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>50</td>
<td>4085.4</td>
<td>0</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>50</td>
<td>8843.4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>50</td>
<td>126.8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1000</td>
<td>30</td>
<td>3</td>
<td>2176.2</td>
<td>0</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>30</td>
<td>4053.7</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>30</td>
<td>8843.4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>30</td>
<td>126.8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>p</td>
<td>k</td>
<td>r</td>
<td>$\lambda_{25}$</td>
<td>$\lambda_{50}$</td>
<td>$\lambda_{75}$</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>---------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Min</td>
<td>FR</td>
<td>WM</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>3</td>
<td>326.6</td>
<td>0</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(35.5)</td>
<td>(0)</td>
<td>(0.02)</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>5</td>
<td>633.4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(45.8)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>10</td>
<td>1221.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(61.5)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>3</td>
<td>1094.5</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(72.5)</td>
<td>(0)</td>
<td>(0.02)</td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>5</td>
<td>1842.1</td>
<td>0</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(128.8)</td>
<td>(0)</td>
<td>(0.01)</td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>10</td>
<td>3566.4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(201.1)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>3</td>
<td>394.7</td>
<td>0</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(31.6)</td>
<td>(0)</td>
<td>(0.02)</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>5</td>
<td>552.9</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(42.6)</td>
<td>(0)</td>
<td>(0.01)</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>10</td>
<td>1224.2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(63.3)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>3</td>
<td>1137.9</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(104)</td>
<td>(0)</td>
<td>(0.02)</td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>5</td>
<td>1732.6</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(100.7)</td>
<td>(0)</td>
<td>(0.01)</td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>10</td>
<td>3620.8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(160.2)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>3</td>
<td>1143.3</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(119.7)</td>
<td>(0)</td>
<td>(0.02)</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>5</td>
<td>1867.1</td>
<td>0</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(123.5)</td>
<td>(0)</td>
<td>(0.01)</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>10</td>
<td>3714.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(199)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>3</td>
<td>1820.5</td>
<td>0</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(218.6)</td>
<td>(0)</td>
<td>(0.02)</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>5</td>
<td>3122.7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(242.9)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>10</td>
<td>6031.6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(267.7)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
</tbody>
</table>
D.3. Prediction Performance of SRRR. We conduct some experiments to gain more insight into the method and compare with other methods. We generate the $X \in \mathbb{R}^{n \times p}$ with independent samples from some multivariate Gaussian $\mathcal{N}(0, \Sigma_X)$. For the first several cases, we generate the response from the true, most favorable model $Y = XUV^\top + E$, where each entry in the support of $U \in \mathbb{R}^{p \times r}$ (sparsity $k$) is independently drawn from a standard Gaussian distribution, and $V \in \mathbb{R}^{q \times r}$ takes the left singular matrix of a Gaussian ensemble. Hence $B = UV^\top$ is the true coefficient matrix. The noise matrix is generated from $\mathcal{N}(0, \sigma_e^2 \Sigma_E)$, where $\sigma_e^2$ is chosen such that the signal-to-noise ratio

\begin{equation}
\text{SNR} = \frac{\text{Tr}(B^\top \Sigma_X B)}{\sigma_e^2 \cdot \text{Tr}(\Sigma_E)}
\end{equation}

is set to a given level. The performance is evaluated by the test $R^2$, defined as follows:

\[ R^2 = 1 - \frac{\|Y - X\hat{B}\|_F^2}{\|Y - \bar{Y}\|_F^2}. \]

We consider several sets of experiments.

1. **Scenario 1-9** Small experiments: $(n, p, k) = (200, 100, 20), (200, 500, 20), (200, 500, 50), q = 20, r = 3$. The $X$ has independent design, and the noise across different responses are all independent, i.e. $\Sigma_X = I_p, \Sigma_E = I_q$. Target SNR = 0.5, 1, 3. The results are evaluated on test sets of size 5000.

2. **Scenario 10-18** Same as Scenario 1-9. The true coefficient matrix is no longer exact low rank. It is perturbed by Gaussian noise with mean 0 and standard deviation 0.5.

3. **Scenario 19-27** Same as Scenario 1-9, except that the predictors are correlated. In particular,

\[ \text{Cov}(x_j, x_{j'}) = \begin{cases} 1, & j = j' \\ \rho, & j \neq j' \end{cases}. \]

We let $\rho = 0.5$ in this set of simulation.

4. **Scenario 28-36** Same as Scenario 10-18, except that the predictors are correlated as in Scenario 19-27.

From the simulations, we find that underestimating the rank can degrade the performance instantly. Overestimating the rank will give one a variance penalty, but it seems to be rather robust compared with the other direction.
D.3.0.1. Scenarios 1-9. Small experiments: \((n, p, k) = (200, 100, 20), (200, 500, 20), (200, 500, 50), q = 20, r = 3\). The \(X\) has independent design, and the noise across different responses are all independent, i.e. \(\Sigma_X = I_p, \Sigma_E = I_q\). Target SNR = 0.5, 1, 3. The results are evaluated on test sets of size 5000.

Fig D.1: Scenario 1-9. \(R^2\) each run is evaluated on a test set of size 5000.
D.3.0.2. Scenario 10-18. Same as Scenario 1-9. The true coefficient matrix is no longer exact low rank. It is perturbed by Gaussian noise with mean 0 and standard deviation 0.5.

Fig D.2: Scenario 10-18. $R^2$ each run is evaluated on a test set of size 5000. The oracle here does not take into account the noise in true coefficient matrix, and do reduced rank regression on the true support and the true rank.
**D.3.0.3. Scenario 19-27.** Same as Scenario 1-9, except that the predictors are correlated. In particular,

\[
\text{Cov}(x_j, x_{j'}) = \begin{cases} 
1, & j = j', \\
\rho, & j \neq j'.
\end{cases}
\]

We let \(\rho = 0.5\) in this set of simulation.

Fig D.3: Scenario 19-27. \(R^2\) each run is evaluated on a test set of size 5000.
D.3.0.4. Scenario 28-36. Same as Scenario 10-18, except that the predictors are correlated as in Scenario 19-27.

Fig D.4: Scenario 28-36. $R^2$ each run is evaluated on a test set of size 5000.
APPENDIX E: ADDITIONAL UK BIOBANK RESULTS ON ASTHMA AND 7 BLOOD AND RESPIRATORY BIOMARKERS

In addition to the 35 biomarkers in the main text, we have also studied asthma and 7 blood and respiratory biomarkers. Here, we defined asthma based on a combination of self-reported questionnaire data and hospital in-patient record data described in DeBoever et al. (2018); Tanigawa et al. (2019). Additionally, we included 4 blood count measurements (monocyte count, neutrophil count, eosinophil count, basophil count) from Category 100081 and 3 spirometry measurements (forced vital capacity (FVC), peak expiratory flow (PEF), and forced expiratory volume in 1 second (FEV1)) from Category 100020 in UK Biobank containing results of haematological assays that were performed on whole blood and pulmonary function tests that were performed at the assessment center, respectively.

We apply the SRRR to the set of phenotypes and expect some performance improvement by leveraging the correlation structure. Overall, we see small rank representation can maintain predictive power for specific phenotypes (see Figure E.1) and that overall the multiresponse model improves the prediction over the single-response lasso model (see Figure E.2). Among the 8 phenotypes included in the analysis, basophil count, which has known pleiotropic roles including inflammatory reactions in immune response and prevention of blood clotting, showed the largest relative improvements, suggesting the utilities of joint predictive modeling of multiple responses with shared group lasso penalty that captures pleiotropic genetic effects across multiple phenotypes.

![Figure E.1: Asthma and Basophil count prediction performance plots.](image)

Fig E.1: Asthma and Basophil count prediction performance plots. Different colors correspond to lower rank predictive performance across (x-axis) training data set and (y-axis) validation data set for (left) asthma and (right) basophil count.
To aid the interpretation of the estimated relationship between genetic variants and phenotypes, we turn our attention to the coefficient matrix $\hat{\beta}$ for the best SRRR model selected on the basis of the predictive performance on the validation set. We apply SVD on the coefficient matrix and primarily focus on the first two components for the interpretation analysis presented here, given that our best performing model was the full-rank model and that we find the first two components explain 35% and 17% of the variance, respectively (Figure E.3). In a special case where we were to choose the rank 2 model, it is possible to directly visualize the coefficient matrix without approximation. When we investigate the relative importance of phenotypes to each component using phenotype contribution score, which quantifies the squared loadings in right singular vectors as described in Tanigawa et al. (2019), we find that the top component is mainly driven by monocyte count, neutrophil count, and eosinophil count; whereas the second component is mainly driven by pulmonary function measurements, such as FVC, FEV1, and PEF, as well as monocyte count (Figure E.3).

Taking advantage of the sparsity of the coefficient matrix from SRRR, one can investigate the genetic variants driving each components and their relationship to the phenotypes. To that end, we applied biplot visualization (Gower, Lubbe and Le Roux, 2011; Gabriel, 1971; Tanigawa et al., 2019) of the decomposed coefficient matrix, $\hat{\beta} = UDV^T$, where the latent component projection of the genetic variants ($UD$) is shown with the primary axes and the singular vectors of the phenotypes ($V$) are shown with the secondary axes (Figure E.4).
three spirometry measures, FEV1, FVC, and PEF, are shown in the similar direction towards the positive values on the second component, indicating that SRRR identified similar set of genetic effects for those three traits. Asthma is shown on the opposite side of the plot, indicating that our model suggest the genetic effects that increase the predicted spirometry measures are protective against asthma. Among the four blood measures, neutrophill, eosinophil, and basophil counts all show large negative values primarily on the first component whereas monocyte count has the non-zero values on both the first and the second components.

Furthermore, it is possible to identify genetic variants driving each components in biplot, where the inner product of a genetic variant (an element in $UD$, shown as a blue point) and a phenotype (an element in $V$, shown as a red point) approximates the regression coefficients ($\hat{\beta}$) of the corresponding variant-phenotype pair. When a variant-phenotype pair is projected on the same line on the same (opposite) direction, the projection of the genetic associations on the shown latent components is positive (negative). When the variant and phenotype vectors are orthogonal or one of the vectors are of zero length, the projection of the genetic associations of the variant-phenotype pair on the displayed latent components is zero. A genetic variant rs445 in $CDK6$ gene, for example, has the largest positive projected value on the component 1 and located on the opposite direction from neutrophill count (Figure E.4), indicating that the genetic variant has the neutrophill-lowering effects, which is consistent with the previous report from univariate genome-wide association analysis (McInnes et al., 2019).

To gain insights on the underlying biology for each component, one can also perform downstream analysis. With genomic regions enrichment of annotations tool (GREAT), for example, one can assess whether the set of genetic variants characterized for each latent component are enriched for known biological processes or functions (McLean et al., 2010; Tanigawa, Dyer and Bejerano, 2019). To that end, we took the top 5,000 genetic variants driving each component and characterized their enrichment to Mammalian phenotype ontology characterized from mouse single-gene knockout experiments (Smith and Eppig, 2015) as described in Tanigawa et al. (2019). We identified the immunity-related mouse phenotypes, including “abnormal humoral immune response” (GREAT binomial $p$-value = $3.8 \times 10^{-56}$) and “increased B cell number” ($p = 6.8 \times 10^{-24}$) for the first and the second component, respectively (Table 10). Together, those results indicate the utilities of feature sparsity and low-rank structure in the coefficient matrix in SRRR for biological interpretation of the identified genotype-phenotype relationships.
Fig E.3: Relative variance explained by each component and the relative importance of phenotypes for each component. The latent components (x-axis) are characterized by applying SVD on the coefficient matrix, $\hat{\beta} = UDV^\top$. The relative importance of phenotypes to each components are quantified using the right singular vectors $V$ using the phenotype contribution score in Tanigawa et al. (2019) and shown in different colors such that the sum is proportional to the relative variance explained by each component.
Fig E.4: The biplot visualization of the top two latent components. Applying SVD on the coefficient matrix $\hat{\beta}$ from SRRR ($\hat{\beta} = UDV^T$), the genetic variants projected on the first two components (elements in $UD$) are shown in blue using the primary axes whereas the singular values of phenotypes (elements in $V$) are shown in red using the secondary axes.

### Table 10

<table>
<thead>
<tr>
<th>Component</th>
<th>Rank</th>
<th>Term ID</th>
<th>Term description</th>
<th>Fold</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>MP:0001800</td>
<td>abnormal humoral immune response</td>
<td>2.22</td>
<td>3.74E-56</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>MP:0000691</td>
<td>enlarged spleen</td>
<td>2.37</td>
<td>3.30E-54</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>MP:0002401</td>
<td>abnormal lymphopoiesis</td>
<td>2.12</td>
<td>3.67E-54</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>MP:0002490</td>
<td>abnormal immunoglobulin level</td>
<td>2.22</td>
<td>3.51E-51</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>MP:0020201</td>
<td>abnormal granulocyte number</td>
<td>2.18</td>
<td>4.29E-49</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>MP:0005014</td>
<td>increased B cell number</td>
<td>2.04</td>
<td>6.76E-24</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>MP:0000693</td>
<td>spleen hyperplasia</td>
<td>2.89</td>
<td>1.04E-20</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>MP:0012441</td>
<td>abnormal monocyte cell number</td>
<td>2.11</td>
<td>1.78E-19</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>MP:0002620</td>
<td>abnormal monocyte morphology</td>
<td>2.09</td>
<td>2.70E-19</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>MP:0001601</td>
<td>abnormal myelopoiesis</td>
<td>2.34</td>
<td>1.12E-18</td>
</tr>
</tbody>
</table>
F.1. Compliance with ethical regulations and informed consent. This research has been conducted using the UK Biobank Resource under Application Number 24983, “Generating effective therapeutic hypotheses from genomic and hospital linkage data” (http://www.ukbiobank.ac.uk/wp-content/uploads/2017/06/24983-Dr-Manuel-Rivas.pdf). Based on the information provided in Protocol 44532 the Stanford IRB has determined that the research does not involve human subjects as defined in 45 CFR 46.102(f) or 21 CFR 50.3(g). All participants of UK Biobank provided written informed consent (more information is available at https://www.ukbiobank.ac.uk/2018/02/gdpr/).

F.2. Population stratification in UK Biobank. We used genotype data from the UK Biobank dataset release version 2 and the hg19 human genome reference for all analyses in the study. To minimize the variabilities due to population structure in our dataset, we restricted our analyses to include 337,151 White British individuals (Figure F.1) based on the following five criteria (DeBoever et al., 2018; Tanigawa et al., 2019) reported by the UK Biobank in the file “ukb_sqc_v2.txt”:
1. self-reported white British ancestry (“in_white_British_ancestry_subset” column)
2. used to compute principal components (“used_in_pca_calculation” column)
3. not marked as outliers for heterozygosity and missing rates (“het_missing_outliers” column)
4. do not show putative sex chromosome aneuploidy (“putative_sex_chromosome_aneuploidy” column)
5. have at most 10 putative third-degree relatives (“excess_relatives” column).

F.3. Variant annotation and quality control. We prepared a genotype dataset by combining the directly-genotype variants, copy number variants (CNVs) and HLA allelotype datasets.
We annotated the directly-genotyped variants using the VEP LOFTEE plugin (https://github.com/konradjk/loftee) and variant quality control by comparing allele frequencies in the UK Biobank and gnomAD (gnomad.exomes.r2.0.1.sites.vcf.gz) as previously described. We focused on variants outside of the major histocompatibility complex (MHC) region (chr6:25477797-36448354) as previously described. We focused on the variants according to the following criteria:
• Missiness of the variant is less than 1%, considering that two genotyping arrays (the UK BiLEVE array and the UK Biobank array) which covers a slightly different set of variants.
• Minor-allele frequency is greater than 0.01%, given the recent reports casting questions on the reliability of ultra low-frequency variants.
• The variant is in the LD-pruned set
• Hardy-Weinberg disequilibrium test p-value is less than $1.0 \times 10^{-7}$
• Manual cluster plot inspection. We investigated the cluster plots for subset of variants and removed 11 variants that have unreliable genotype calls.
• Passed the comparison of minor allele frequency with gnomAD dataset as described before
CNVs were called by applying PennCNV v1.0.4 on raw signal intensity data from each array within each genotyping batch as previously described. We applied a filter on minor-allele frequency (MAF > 0.01%), which resulted in 8,274 non-rare (MAF > 0.01%) CNVs.

The HLA data from the UK Biobank contains all HLA loci (one line per person) in a specific order (A, B, C, DRB5, DRB4, DRB3, DRB1, DQB1, DQA1, DPB1, DPA1). We downloaded these values, which were imputed via the HLA:IMP*2 program (Resource 182); the UK Biobank reports one value per imputed allele, and only the best-guess alleles are reported. Out of the 362 alleles reported in UKB, we used 175 alleles that were present in >0.1% of the population surveyed.

Fig F.1: The identification of unrelated White British individuals in UK Biobank. The first two genotype principal components (PCs) are shown on the x- and y-axis and the identified unrelated White British individuals (Methods) are shown in red.