Structured Sufficient Dimension Reduction via Decomposition and Sparsity

Zhanyan Yang Yangdi Shang

January 25, 2026

Abstract

We study high-dimensional sufficient dimension reduction with sparse signals, aiming to recover the central subspace while selecting relevant variables. Using sliced inverse regression statistics $(\widehat{\Sigma}, \widehat{U})$, we propose StructuredSDR, which factorizes the direction matrix as $B = WV^\top$ and imposes a row-wise sparsity penalty on W. The resulting problem is solved by alternating optimization: proximal-gradient updates for W and a closed-form orthogonal Procrustes update for V. We further use warm starts and a post-refit debiasing step to improve numerical stability, and enforce a matched sparsity budget via a top-k truncation rule for fair variable-selection comparison. Monte Carlo experiments with at least 50 replicates on four benchmark simulation models show that StructuredSDR achieves smaller subspace error and stronger test-set dependence preservation in single-index settings, while exhibiting complementary strengths relative to SEAS in multi-index and inverse-regression settings. Overall, StructuredSDR offers a practical and controllable alternative for high-dimensional SDR.

Keywords: sufficient dimension reduction; sliced inverse regression; high-dimensional sparsity; alternating optimization; variable selection.

1 Introduction

1.1 Background

In regression analysis with high-dimensional predictors, researchers are often interested in understanding how a response variable depends on a large number of covariates. When the predictor dimension is large relative to the sample size, directly modeling the relationship in the original space leads to both statistical and computational challenges. High-dimensional estimators tend to be unstable, sensitive to noise, and difficult to interpret. In many applications, only a small subset of variables or a low-dimensional structure is truly relevant to the response, making dimension reduction a fundamental tool for both inference and prediction.

Sufficient dimension reduction (SDR) provides a model-free framework for addressing this problem. Rather than assuming a specific parametric form for the regression function, SDR

seeks a low-dimensional linear subspace of the predictor space such that projecting the predictors onto this subspace preserves all information about the conditional distribution of the response given the predictors. If such a subspace exists, the original high-dimensional regression problem can be replaced by an equivalent low-dimensional one, greatly reducing complexity while retaining essential information. This perspective is particularly attractive in situations where the true relationship between predictors and response may be nonlinear or unknown.

Among classical SDR methods, sliced inverse regression (SIR) is one of the most widely used approaches. SIR studies the inverse relationship by examining how the conditional mean of the predictors varies across different slices of the response variable. By discretizing the response and analyzing between-slice variations of predictor means, SIR is able to recover directions in the predictor space that are informative for the response. The method relies primarily on second-order moments and is therefore easy to implement and scalable in moderate dimensions. Under suitable conditions, the subspace identified by SIR coincides with the central subspace that characterizes the full conditional distribution of the response.

However, classical SIR encounters serious difficulties in high-dimensional settings. When the predictor dimension is large, the sample covariance matrix may be ill-conditioned or singular, leading to unstable estimates. More importantly, classical SIR does not incorporate structural information such as sparsity or low intrinsic dimension, which are often essential for interpretability and accurate estimation in modern applications. As a result, naive application of SIR in high dimensions can produce noisy subspace estimates involving many irrelevant variables.

To overcome these limitations, recent developments in high-dimensional SDR introduce structural constraints into the estimation procedure. Two types of structure are particularly important. First, the dimension of the central subspace is typically small compared to the ambient dimension, suggesting a low-rank representation. Second, only a small subset of predictors is often relevant, implying sparsity at the variable level. Incorporating these structures allows SDR methods to simultaneously perform dimension reduction and variable selection, leading to more stable and interpretable results.

From a computational standpoint, structured SDR methods rely heavily on modern optimization techniques. Penalties such as group sparsity and low-rank regularization give rise to non-smooth objective functions, which are commonly handled using proximal algorithms and related iterative methods. Techniques such as proximal gradient methods and alternating optimization enable efficient numerical solutions even when the problem size is large. These tools play a central role in making structured SDR practically feasible and are closely related to optimization methods discussed in this course.

This project focuses on structured sufficient dimension reduction based on SIR-type statistical objects. The goal is to investigate how different structural modeling choices and optimization strategies affect subspace recovery, variable selection, and computational efficiency in high-dimensional settings. By developing and analyzing a structured SDR approach within a unified experimental framework, this study aims to provide a clear and reproducible com-

parison of methods and to illustrate how modern optimization techniques can be effectively integrated into high-dimensional statistical learning problems.

1.2 Research Content

This project focuses on high-dimensional structured sufficient dimension reduction based on sliced inverse regression (SIR). The objective is to propose and implement a new structured SDR method, referred to as StructuredSDR, on a unified statistical representation, and to conduct a reproducible comparison with the SEAS method. The study aims to address the following question: in high-dimensional settings, is it possible to stably estimate the central subspace, automatically perform variable selection and structural dimension control, and at the same time retain an effective low-dimensional representation under potentially nonlinear associations between the predictors and the response.

1.2.1 Unified formulation: from SIR statistics to a quadratic objective

For each simulation replicate, two core statistics are constructed from the training data using SIR:

- $M \in \mathbb{R}^{p\times p}$, the centered sample covariance matrix, serving as an estimator of the population covariance Σ;
- $U \in \mathbb{R}^{p\times H}$, the matrix formed by stacking the centered conditional mean vectors of the predictors across slices, serving as an estimator of the population quantity U.

At the population level, the central subspace in SIR can be expressed as $\mathrm{span}(\Sigma^{-1}U)$. Equivalently, this subspace can be characterized as the solution to an unpenalized quadratic optimization problem,

$$B^\star = \arg\min_{B\in\mathbb{R}^{p\times H}} \ \mathrm{tr}(B^\top \Sigma B) - 2\,\mathrm{tr}(B^\top U).$$

At the sample level, replacing (Σ, U) by (M, U) yields an isomorphic quadratic objective. Both methods considered in this project, SEAS and StructuredSDR, can be viewed as adding low-rank and row-sparse structure on top of this quadratic objective in order to accommodate high-dimensional settings and to enable interpretable variable selection.

1.2.2 SEAS: convex structured estimation (baseline method)

SEAS treats $B \in \mathbb{R}^{p\times H}$ as the optimization variable and augments the quadratic fitting term with two types of structural penalties:

- row-wise sparsity via a group-lasso penalty $\lambda_1 \sum_{j=1}^{p} \|B_{j\cdot}\|_2$, which encourages only a small number of variables to have nonzero rows;

- low-rank structure via a nuclear norm penalty $\lambda_2\|B\|$, which induces automatic rank reduction and hence structural dimension selection.

The resulting optimization problem can be written as

$$\min_{B\in\mathbb{R}^{p\times H}} \ \operatorname{tr}(B^\top MB) - 2\operatorname{tr}(B^\top U) + \lambda_1 \sum_{j=1}^{p} \|B_{j\cdot}\|_2 + \lambda_2\|B\|_*.$$

Since this objective is convex, SEAS can be solved using standard variable splitting and proximal operators. The estimated subspace basis and structural dimension are extracted from the singular structure of the solution matrix, and the selected variables are determined by whether the corresponding rows are zero. In this project, SEAS serves as the baseline method for evaluating the estimation quality and structural recovery performance of StructuredSDR under the same (M, U) input.

1.2.3 StructuredSDR: decomposition-based structured estimation (proposed method)

The StructuredSDR method proposed in this project adopts an explicit low-rank decomposition strategy. Let d denote the structural dimension and parameterize

$$B = WV^\top, \qquad W \in \mathbb{R}^{p\times d}, \quad V \in \mathbb{R}^{H\times d}, \quad V^\top V = I_d.$$

The rows of W correspond directly to the predictor variables, so imposing row-wise sparsity on W leads naturally to variable selection. To obtain a computationally tractable formulation, the row-sparsity constraint is relaxed using a group-lasso ($\ell_{2,1}$) penalty, resulting in the objective function adopted in this study,

$$\min_{W,\ V^\top V=I_d} \ f(W,V) := \operatorname{tr}(W^\top MW) - 2\operatorname{tr}(W^\top UV) + \lambda\|W\|_{2,1}, \quad \|W\|_{2,1} = \sum_{j=1}^{p} \|W_{j\cdot}\|_2.$$

This objective is convex in W when V is fixed, and reduces to a matrix optimization problem under orthogonality constraints when W is fixed. This structure naturally leads to an alternating optimization scheme:

- with V fixed, the W-step solves a convex subproblem consisting of a quadratic term and a row-wise group-lasso penalty, which can be handled using proximal gradient methods and their accelerated variants;
- with W fixed, the V-step reduces to an orthogonal Procrustes problem, which admits a closed-form solution.

The final subspace estimate is taken as $\hat{\beta} = \operatorname{orth}(W)$, i.e., an orthonormal basis of $\operatorname{span}(W)$, and the selected variable set is defined as

$$\hat{S}_{\text{struct}} = \{j : \|W_{j\cdot}\|_2 > 0 \text{ (or exceeds a numerical threshold)}\}.$$

In this way, StructuredSDR realizes low-rank structure through the choice of d and row sparsity through the $\ell_{2,1}$ penalty on W, forming a decomposition-based structured SDR approach that runs in parallel to the convex matrix regularization strategy used in SEAS.

1.2.4 Answering the research question through comparable estimation targets

The comparison in this project is built on consistency at two levels.

First, the statistical input is identical: both methods take the same (M, U) constructed from the training data as input and perform structured estimation on the same SIR-based quadratic objective.

Second, the output objects are aligned: both methods ultimately produce a $p \times d$ subspace basis $\hat{\beta}$, representing an estimate of the central subspace, and a variable set $\hat{S} \subseteq \{1, \ldots, p\}$, representing the selected predictors.

This alignment allows for a direct and fair evaluation of central subspace recovery and variable selection performance under a unified framework. In addition, since the goal of SDR is not only structural recovery but also the preservation of meaningful dependence between X and Y, the dependence between the reduced representation $Z = X\hat{\beta}$ and the response Y is also evaluated. Together, subspace recovery, support recovery, and dependence preservation constitute a notion of functionality: a method is considered effective if it not only converges numerically, but also produces stable, interpretable, and informative structured dimension reduction results.

1.2.5 Connection to course content

The modeling and algorithmic components of this project are closely related to key topics covered in the course.

- High-dimensional regularization: row-wise sparsity via group-lasso is used for variable selection, while low-rank structure is used to control dimension and structural complexity.
- Convex versus nonconvex optimization: SEAS is formulated as a convex matrix estimation problem, whereas StructuredSDR is a decomposition-based nonconvex problem with a block-separable structure.
- Proximal and splitting algorithms: SEAS relies on variable splitting and proximal operators for nuclear norm and group sparsity penalties; StructuredSDR employs proximal gradient updates in the W-step and closed-form updates under orthogonality constraints in the V-step.
- Model selection and validation: regularization parameters and structural dimensions are selected in a data-driven manner using validation sets, reflecting standard principles of model selection.

- Monte Carlo evaluation and reproducibility: repeated simulations, summary statistics, and uncertainty quantification via standard errors are used to assess stability across different data generating mechanisms.

1.2.6 Novelty and contributions beyond coursework

The focus of this project is not a one-off implementation of a single algorithm, but the construction of a complete and reproducible research pipeline for structured SDR. The main contributions are as follows.

First, a decomposition-based structured SDR framework is proposed and implemented for the SIR quadratic objective. By parameterizing B as $WV^\top$, low-rank structure is made explicit and variable selection is directly encoded through row sparsity of W, forming a parallel approach to the convex matrix regularization strategy used in SEAS.

Second, a fully aligned comparison protocol is established. By using identical (M, U) inputs and matching output objects (subspace bases and variable supports), the project enables reproducible and interpretable comparisons of structural recovery and dependence preservation, rather than isolated performance metrics from single runs.

Third, the StructuredSDR method is developed in a modular manner. A base version consists only of the core alternating optimization steps, and additional modules are introduced to improve practical usability and performance. The detailed mechanisms and parameter settings of these modules are presented in the experimental design section, ensuring that the study goes beyond a direct repetition of classroom exercises and instead provides a substantive extension focused on reproducibility, stability, and systematic evaluation in high-dimensional SDR.

These components together form the core of the present study: under a unified SIR-based quadratic formulation, structured SDR estimation is carried out using both convex matrix regularization (SEAS) and decomposition-based row-sparse modeling (StructuredSDR), and systematic simulation experiments are used to address the question of whether high-dimensional SDR can be performed stably and with interpretable structure.

2 Monte Carlo Simulation

2.1 Experimental Design

This section presents the overall design and workflow of the simulation experiments, which aim to compare the performance of SEAS-SIR and StructuredSDR under the same data generating mechanisms and the same class of SIR-based statistical inputs. The experimental protocol follows a closed-loop structure consisting of data generation, train/validation/test splitting, construction of SIR statistics, hyperparameter tuning, refitting, test evaluation, and Monte Carlo aggregation. Key algorithmic details, including initialization, step-size selection, convergence thresholds, and post-processing, are discussed to ensure fairness and reproducibility of

the comparison.

2.1.1 Data generating models and basic settings (M1–M4)

In each replicate, a dataset (X, y) is generated from a specified model, and the true central subspace directions (or direction matrices) together with the true support are recorded as benchmarks. The four models follow the standard simulation settings used in SEAS and cover linear, nonlinear, multiple-index, and inverse regression scenarios.

- *(M1) Linear single-index model*: $X \sim N(0, \mathrm{AR}(\rho))$ with $\rho = 0.5$, $\varepsilon \sim N(0, 1)$, and $y = X^\top \beta^\star + \varepsilon$, with $d_{\text{true}} = 1$.
- *(M2) Nonlinear single-index model*: the same (X, ε) structure as in (M1), and $y = \sinh(X^\top \beta^\star) + \varepsilon$, with $d_{\text{true}} = 1$.
- *(M3a) Multiple-index model (elliptical distribution)*: $X \sim N(0, \mathrm{AR}(0.5))$, $\varepsilon \sim N(0, 1)$, and $y = (X^\top \beta_1^\star) \exp(X^\top \beta_2^\star + 0.5\varepsilon)$, with $d_{\text{true}} = 2$.
- *(M3b) Multiple-index model (non-elliptical mixture distribution)*: X follows a three-component Gaussian mixture with different means and correlation structures, and the same response-generating mechanism as in (M3a), with $d_{\text{true}} = 2$.
- *(M4) PFC inverse model*: $y \sim \mathrm{Unif}(-1, 1)$, $f(y) = (y, |y|)^\top$, and $X = \beta_1^\star y + \beta_2^\star |y| + \varepsilon$, where $\varepsilon \sim N(0, \mathrm{AR}(0.5))$, with $d_{\text{true}} = 2$.

The true coefficients follow the sparsity patterns used in SEAS: in single-index models, the first 10 coordinates of $\beta^\star$ are nonzero; in multiple-index and PFC models, only the first 6 coordinates of $(\beta_1^\star, \beta_2^\star)$ are nonzero. The true support $S^\star$ is therefore defined by whether a row of the coefficient matrix is identically zero.

Unless otherwise stated, the default configuration is $n_{\text{total}} = 450$, $p = 250$, $n_{\text{reps}} = 20$, and $H = 6$ SIR slices. Across models, only the data generation mechanism differs, while all subsequent steps remain identical.

2.1.2 Parallel comparison protocol: Train/Validation/Test splitting and unified inputs

For each replicate, the data are randomly permuted and split into training (60%), validation (20%), and test (20%) sets. The training set is used to construct SIR statistics and fit the models, the validation set is used for hyperparameter tuning, and the test set is reserved for final evaluation.

Both methods operate on the same SIR-based statistical inputs constructed from the training data. Specifically, M denotes the centered covariance estimator and U denotes the matrix of centered slice-wise conditional means. By restricting both methods to the same (M, U) interface, the comparison avoids biases arising from unequal access to raw covariate information.

Algorithm 1: Parallel simulation pipeline for StructuredSDR and SEAS

Input : model type (M1–M4); (n_{total}, p); split ratios; number of slices H; candidate grids; candidate dimension set; number of replicates n_{reps}.

Output: aggregated performance summaries (mean and standard deviation) over n_{reps} replicates.

for $r = 1$ **to** n_{reps} **do**

- Generate (X, y) and record true directions and true support $S^\star$
- Split the data into Train, Validation, and Test sets
- Compute $(M, U) \leftarrow \text{SIR}(X_{\text{tr}}, y_{\text{tr}}; H)$
- Perform hyperparameter tuning for both methods using validation distance correlation
- Refit both methods on Train+Validation to obtain final $\hat{\beta}$ and $\hat{S}$
- Evaluate on the Test set: subspace distance, TPR/FPR, test distance correlation, and timing

Aggregate results across replicates using mean and standard deviation

Two settings are considered. In the known-d setting, the structural dimension is fixed at the true value d_{true}. In the unknown-d setting, the dimension is selected from the candidate set $d \in \{1, 2, 3, 4\}$ using the validation data.

2.1.3 Hyperparameter tuning and evaluation criteria

Hyperparameter selection is based on the dependence between the reduced representation and the response. For each candidate configuration, an estimated subspace $\hat{\beta}$ is obtained and the validation score is computed as

$$\text{Score} = \text{dCor}(X_{\text{val}}\hat{\beta},\ y_{\text{val}}),$$

where distance correlation is used to capture potentially nonlinear dependence. The configuration with the largest validation score is selected for each replicate.

Final evaluation is conducted on the test set. For each replicate, the following metrics are computed and then aggregated across n_{reps} replicates using mean and standard deviation: subspace recovery error measured by the Frobenius norm of projection matrix differences, variable selection accuracy in terms of TPR and FPR, dependence preservation measured by test-set distance correlation, and computational cost including tuning time and refitting time.

2.1.4 Overall experimental workflow and parallel comparison structure

For each replicate, the experimental workflow proceeds as follows. StructuredSDR and SEAS are tuned, refit, and evaluated in parallel, sharing the same data splits and the same SIR statistics.

2.1.5 Detailed solution procedure of StructuredSDR (final version with warm start and post-refitting)

We summarize the StructuredSDR procedure used in our experiments. Given the training SIR statistics (M, U), the method estimates a sparse d-dimensional subspace by optimizing over a factorization $B = WV^\top$ with row-sparse $W \in \mathbb{R}^{p\times d}$ and $V \in \mathbb{R}^{H\times d}$ satisfying $V^\top V = I_d$. Hyperparameters are selected by validation distance correlation, and the final estimator is obtained by refitting on the combined Train+Validation set.

Algorithm 2: StructuredSDR: warm start, tuning, and refitting (main procedure)

Input : training-set SIR statistics (M, U); candidate dimension set $\mathcal{D}$ (singleton $\{d_{\text{true}}\}$ in the known-d setting); sparsity parameter grid $\Lambda = \{\lambda_1 > \cdots > \lambda_L\}$; validation data $(X_{\text{val}}, y_{\text{val}})$; ridge parameter $\rho > 0$; support threshold ε_{sup}; number of post-refit iterations T_{refit}.

Output: selected dimension $\hat{d}$, sparsity parameter $\hat{\lambda}$, subspace basis $\hat{\beta} \in \mathbb{R}^{p\times\hat{d}}$, support $\hat{S}$.

Initialization *(A) Warm start (precomputed once for all d):*

$B_0 \leftarrow (M + \rho I_p)^{-1} U \in \mathbb{R}^{p\times H}$

Compute truncated SVD (truncate at $d_{\max} = \max \mathcal{D}$): $B_0 \approx U_0 D_0 V_0^\top$

foreach $d \in \mathcal{D}$ **do**

  $W^{(0)}(d) \leftarrow U_{0,d} D_{0,d}, \quad V^{(0)}(d) \leftarrow V_{0,d}, \quad$ with $V^{(0)\top}(d) V^{(0)}(d) = I_d$

(B) Hyperparameter tuning (for each d, warm start along λ path):

foreach $d \in \mathcal{D}$ **do**

  $W_{\text{prev}} \leftarrow W^{(0)}(d), \quad V_{\text{prev}} \leftarrow V^{(0)}(d), \quad$ best_score $\leftarrow -\infty$

  foreach $\lambda \in \Lambda$ **do**

    (B.1) Alternating optimization:

    $(W, V) \leftarrow \textsc{AltOpt}(M, U, \lambda; W_{\text{prev}}, V_{\text{prev}})$

    (B.2) Support extraction:

    $\hat{S}(\lambda, d) \leftarrow \{i : \|W_{i\cdot}\|_2 > \varepsilon_{\text{sup}}\}$

    (B.3) Post-refitting:

    $(W, V) \leftarrow \textsc{PostRefit}(M, U, \hat{S}(\lambda, d), T_{\text{refit}}; W, V)$

    (B.4) Validation scoring:

    $\beta(\lambda, d) \leftarrow \text{orth}(W) \in \mathbb{R}^{p\times d}$

    $\text{score}(\lambda, d) \leftarrow \text{dCor}(X_{\text{val}}\beta(\lambda, d),\, y_{\text{val}})$

    (B.5) Path-wise warm start:

    $W_{\text{prev}} \leftarrow W, \quad V_{\text{prev}} \leftarrow V$

    if score$(\lambda, d) >$ best_score **then**

      Store $(\hat{\lambda}(d), \hat{\beta}(d), \hat{S}(d)) \leftarrow (\lambda, \beta(\lambda, d), \hat{S}(\lambda, d))$

      best_score $\leftarrow$ score(λ, d)

Select $(\hat{d}, \hat{\lambda})$ maximizing best_score over $d \in \mathcal{D}$

(C) Refitting (Train+Validation):

Recompute (M, U) using Train+Validation data, repeat step (A), and run steps (B.1)–(B.3) with $(\hat{d}, \hat{\lambda})$

$\hat{\beta} \leftarrow \text{orth}(W), \quad \hat{S} \leftarrow \{i : \|W_{i\cdot}\|_2 > \varepsilon_{\text{sup}}\}$

For presentation, Algorithm 2 is organized into three components. **(A) Warm start** computes $B_0 = (M + \rho I_p)^{-1} U$ and obtains an initialization $\left(W^{(0)}(d), V^{(0)}(d)\right)$ for each $d \in \mathcal{D}$ via a truncated SVD of B_0 (truncated at $d_{\max} = \max \mathcal{D}$). **(B) Hyperparameter tuning** evaluates candidate pairs (d, λ) using warm starts along the λ path. For each (d, λ), the core

Algorithm 3: AltOpt: Alternating optimization for fixed (d, λ)

Input : M, U, λ; initialization $(W_{\text{init}}, V_{\text{init}})$
Output: (W, V) at convergence
Initialize:
$(W^{(t)}, V^{(t)}) \leftarrow (W_{\text{init}}, V_{\text{init}})$
repeat
  W-step (proximal gradient with FISTA):
  Set step size $\eta \approx (2\|M\|_2)^{-1}$, with $\|M\|_2$ estimated by power iteration
  Perform K inner FISTA iterations:
  for $k = 0$ **to** $K - 1$ **do**
    $G \leftarrow 2MW^{(t,k)} - 2UV^{(t)}$
    $Z \leftarrow W^{(t,k)} - \eta G$
    for $i = 1$ **to** p **do**
      $W_{i\cdot}^{(t,k+1)} \leftarrow \left(1 - \frac{\eta\lambda}{\|Z_{i\cdot}\|_2}\right)_+ Z_{i\cdot}$
  Set $W^{(t+1)} \leftarrow W^{(t,K)}$
  V-step (Procrustes update):
  $M_V \leftarrow U^\top W^{(t+1)}$
  Compute SVD $M_V = P\Lambda Q^\top$
  $V^{(t+1)} \leftarrow PQ^\top$
until *converged*
return $(W^{(t)}, V^{(t)})$

Algorithm 4: PostRefit: Post-refitting on extracted support $\hat{S}$

Input : M, U; support set $\hat{S}$; iterations T_{refit}; initial (W, V)
Output: refitted (W, V) after T_{refit} iterations
for $r = 1$ **to** T_{refit} **do**
  Solve $M_{\hat{S},\hat{S}} W_{\hat{S}} = U_{\hat{S}} V$
  Set $W_{\hat{S}^c} = 0$
  $M_V \leftarrow U^\top W$
  Compute SVD $M_V = P\Lambda Q^\top$
  $V \leftarrow PQ^\top$
return (W, V)

fitting routine AltOpt (Algorithm 2a) alternates between: (i) a W-update using proximal gradient with K inner FISTA iterations under a row-wise group-lasso penalty, and (ii) a Procrustes update of V based on the SVD of $U^\top W$. After convergence, a support set $\hat{S}(\lambda, d)$ is extracted by thresholding row norms of W. Conditioned on $\hat{S}(\lambda, d)$, the post-refitting routine PostRefit (Algorithm 2b) performs T_{refit} iterations of restricted least-squares updates $M_{\hat{S},\hat{S}} W_{\hat{S}} = U_{\hat{S}} V$ with $W_{\hat{S}^c} = 0$, followed by Procrustes updates of V. The resulting orthonormal basis $\beta(\lambda, d) = \text{orth}(W)$ is scored by $\text{dCor}(X_{\text{val}}\beta(\lambda, d), y_{\text{val}})$, and the best $(\hat{d}, \hat{\lambda})$ is selected.

Finally, **(C) Refitting** recomputes (M, U) on Train+Validation data, repeats the warm-start initialization, and reruns AltOpt and PostRefit at $(\hat{d}, \hat{\lambda})$ to output $\hat{\beta} = \text{orth}(W)$ and $\hat{S} = \{i : \|W_{i\cdot}\|_2 > \varepsilon_{\text{sup}}\}$. The known-$d$ setting corresponds to $\mathcal{D} = \{d_{\text{true}}\}$; otherwise the procedure selects d by maximizing the validation score over $d \in \mathcal{D}$.

2.1.6 Key implementation details and discussion

This subsection discusses implementation aspects that have a direct impact on numerical stability, reproducibility, and fairness of comparison, including initialization and warm starts, step-size selection and stopping criteria, post-refitting for bias reduction, and the definition of support in SEAS together with sparsity-budget alignment.

(1)Base version versus final version: warm start and post-refitting. StructuredSDR alternates between updates of (W, V), where the W-step solves a convex subproblem with a row-wise group-lasso penalty and the V-step admits a closed-form Procrustes update. The overall problem remains nonconvex due to the orthogonality constraint $V^\top V = I_d$ and the alternating structure. In the base version, a common choice is to initialize $W^{(0)} = 0$ and to specify $V^{(0)}$ using SIR. Under strong sparsity penalties in high dimensions, this initialization has three limitations. First, when λ is large, the proximal operator may shrink many rows to zero at early iterations; with a weak initial point, the algorithm can quickly enter regions with predominantly zero rows, making recovery of correct directions difficult and amplifying sensitivity to λ. Second, due to nonconvexity, different initializations can lead to different local solutions or sparsity patterns, increasing variability across Monte Carlo replicates and reducing stability. Third, grid-based tuning over λ requires repeated fitting, and restarting from weak initial points incurs cumulative computational cost and prevents information reuse across candidates.

The final procedure incorporates two forms of warm start together with post-refitting: ridge-based initialization followed by truncated SVD, continuation along the λ path, and post-refitting on the selected support to mitigate shrinkage bias and improve subspace estimation accuracy. These modifications do not alter the model or objective function but substantially improve numerical robustness and estimation quality.

(2)Step-size selection and stopping criteria. With V fixed, the W-step solves

$$\min_W \ \mathrm{tr}(W^\top M W) - 2\mathrm{tr}(W^\top U V) + \lambda \|W\|_{2,1},$$

whose smooth part has gradient $\nabla h(W) = 2MW - 2UV$ and Lipschitz constant proportional to $\|M\|_2$. The implementation uses $\eta \approx (2\|M\|_2)^{-1}$, with $\|M\|_2$ estimated by power iteration. This choice avoids oscillation or divergence in high dimensions, yields comparable iteration behavior across replicates, and supports efficient continuation along the λ path. Outer alternating updates use relative changes in objective values or variables as stopping criteria, while inner FISTA iterations use relative changes in W. Thresholds are chosen to ensure sufficient convergence for fair comparison without excessive computational cost.

(3)Post-refitting and shrinkage bias. Both group-lasso and nuclear-norm penalties introduce systematic shrinkage bias. Even when the support is accurately identified, coefficient

magnitudes may be underestimated, affecting the estimated subspace and downstream dependence measures. Post-refitting follows a "select-then-estimate" principle: a penalized solution yields $\hat{S}$, after which an unpenalized linear system is solved on $\hat{S}$,

$$M_{\hat{S},\hat{S}} W_{\hat{S}} = U_{\hat{S}} V, \qquad W_{\hat{S}^c} = 0,$$

followed by a Procrustes update. This step operates on a reduced subproblem and often improves the quality of $\hat{\beta} = \mathrm{orth}(W)$, as reflected in both subspace distance and test-set distance correlation.

(4) Numerical thresholds and support definition in SEAS. In SEAS, the numerical threshold ε serves both as a stopping criterion for ADMM and as an elementwise cutoff applied to the estimated basis $\hat{\beta}$. Support must therefore be defined after elementwise thresholding, based on whether a row of $\hat{\beta}$ is nonzero. Defining support directly from the raw matrix B using extremely small thresholds can misclassify numerical noise as signal, leading to inflated support size and distorted FPR, particularly in high dimensions. A common threshold level is used across methods, with $\varepsilon = 10^{-3}$ for SEAS and $\varepsilon_{\mathrm{sup}} = 10^{-3}$ for StructuredSDR.

(5)Sparsity-budget alignment via top-k truncation. In variable selection evaluation, SEAS outputs are truncated to retain the top-k rows ranked by row norms, where

$$k = |\hat{S}_{\mathrm{struct}}|.$$

The resulting row set defines $\hat{S}_{\mathrm{seas}}$, which is used to compute TPR and FPR. For directional evaluation, $\hat{\beta}$ can be recomputed from the truncated matrix. This protocol ensures that comparisons focus on directional recovery under a fixed sparsity budget rather than differences driven by support size alone. It does not modify the optimization or tuning procedures of SEAS, and serves solely to standardize evaluation of variable selection and subspace recovery.

2.2 Simulation results

This subsection reports Monte Carlo summaries for the four simulation models (M1–M4) under the Known-d and Unknown-d settings, comparing StructuredSDR with SEAS. For M1–M3, SEAS is used in the SEAS-SIR form, while M4 uses SEAS-PFC. All metrics are computed on the test set and aggregated across $R = 50$ replicates in terms of means and standard deviations. Alongside the mean and standard deviation, we also report the standard error (SE), computed as

$$\mathrm{SE} = \mathrm{SD}/\sqrt{R}.$$

To make the variable-selection metrics comparable across methods, all experiments adopt a sparsity-budget alignment scheme. For each replicate, the SEAS solution is truncated by row norm, keeping only the top-k rows, where $k = |\widehat{S}_{\mathrm{struct}}|$ is the final support size selected by

StructuredSDR for that replicate. Both TPR and FPR are then computed with respect to this aligned support set. In this way, the variable-selection metrics primarily reflect the ability to identify the correct variables rather than differences in the overall support size.

Evaluation metrics. The subspace error Dist is smaller for better performance. TPR should be as large as possible and FPR as small as possible; in the tables TPR and FPR are reported in percentages. The test-set distance correlation (dCor) is larger for better dependence preservation. In the Unknown-d setting, we also report the mean and variability of the selected dimension $\hat{d}$.

Setting	Method	Dist	TPR (%)	FPR (%)	dCor
Known-d	StructuredSDR	0.394 ± 0.141 (0.031)	91.5 ± 11.8 (2.6)	2.3 ± 4.8 (1.1)	0.721 ± 0.078 (0.017)
Known-d	SEAS-SIR	0.451 ± 0.112 (0.025)	91.5 ± 10.9 (2.4)	2.3 ± 5.0 (1.1)	0.701 ± 0.080 (0.018)
Unknown-d	StructuredSDR	0.413 ± 0.188 (0.042)	91.0 ± 11.7 (2.6)	1.8 ± 3.7 (0.8)	0.723 ± 0.075 (0.017)
Unknown-d	SEAS-SIR	0.397 ± 0.098 (0.022)	95.0 ± 7.6 (1.7)	1.6 ± 4.0 (0.9)	0.715 ± 0.077 (0.017)

Table 1: M1 (linear SIM): mean ± SD (SE in parentheses), $R = 50$.

Setting	Method	Dist	TPR (%)	FPR (%)	dCor
Known-d	StructuredSDR	0.273 ± 0.095 (0.021)	95.5 ± 6.9 (1.5)	0.7 ± 1.4 (0.3)	0.811 ± 0.056 (0.013)
Known-d	SEAS-SIR	0.413 ± 0.105 (0.023)	93.0 ± 9.2 (2.1)	0.8 ± 1.4 (0.3)	0.779 ± 0.063 (0.014)
Unknown-d	StructuredSDR	0.273 ± 0.095 (0.021)	95.5 ± 6.9 (1.5)	0.7 ± 1.4 (0.3)	0.811 ± 0.056 (0.013)
Unknown-d	SEAS-SIR	0.362 ± 0.092 (0.020)	95.5 ± 6.0 (1.4)	0.7 ± 1.4 (0.3)	0.792 ± 0.058 (0.013)

Table 2: M2 (nonlinear SIM): mean ± SD (SE in parentheses), $R = 50$.

Setting	Method	Dist	TPR (%)	FPR (%)	dCor
Known-d	SEAS-SIR	0.273 ± 0.058 (0.013)	100.0 ± 0.0 (0.0)	1.8 ± 2.2 (0.5)	0.696 ± 0.054 (0.012)
Known-d	StructuredSDR	0.283 ± 0.076 (0.017)	100.0 ± 0.0 (0.0)	1.8 ± 2.2 (0.5)	0.689 ± 0.066 (0.015)
Unknown-d	SEAS-SIR	0.314 ± 0.130 (0.029)	96.7 ± 11.6 (2.6)	1.6 ± 2.3 (0.5)	0.689 ± 0.055 (0.012)
Unknown-d	StructuredSDR	0.289 ± 0.106 (0.024)	97.5 ± 11.2 (2.5)	1.6 ± 2.3 (0.5)	0.680 ± 0.072 (0.016)

Table 3: M3a (elliptical): mean ± SD (SE in parentheses), $R = 50$.

Setting	Method	Dist	TPR (%)	FPR (%)	dCor
Known-d	SEAS-SIR	0.242 ± 0.198 (0.044)	100.0 ± 0.0 (0.0)	4.0 ± 3.5 (0.8)	0.777 ± 0.059 (0.013)
Known-d	StructuredSDR	0.387 ± 0.098 (0.022)	100.0 ± 0.0 (0.0)	4.0 ± 3.5 (0.8)	0.775 ± 0.055 (0.012)
Unknown-d	SEAS-SIR	0.329 ± 0.284 (0.064)	99.2 ± 3.7 (0.8)	3.0 ± 3.0 (0.7)	0.775 ± 0.058 (0.013)
Unknown-d	StructuredSDR	0.402 ± 0.115 (0.026)	98.3 ± 5.1 (1.1)	3.0 ± 3.0 (0.7)	0.775 ± 0.053 (0.012)

Table 4: M3b (mixture): mean ± SD (SE in parentheses), $R = 50$.

Setting	Method	Dist	TPR (%)	FPR (%)	dCor
Known-d	StructuredSDR	0.407 ± 0.079 (0.018)	99.2 ± 3.7 (0.8)	0.5 ± 0.8 (0.2)	0.910 ± 0.016 (0.004)
Known-d	SEAS-PFC	0.080 ± 0.078 (0.018)	99.2 ± 3.7 (0.8)	0.5 ± 0.8 (0.2)	0.909 ± 0.017 (0.004)
Unknown-d	StructuredSDR	0.416 ± 0.092 (0.020)	99.2 ± 3.7 (0.8)	0.5 ± 0.8 (0.2)	0.910 ± 0.016 (0.004)
Unknown-d	SEAS-PFC	0.086 ± 0.078 (0.017)	99.2 ± 3.7 (0.8)	0.5 ± 0.8 (0.2)	0.909 ± 0.017 (0.004)

Table 5: M4 (PFC inverse): mean ± SD (SE in parentheses), $R = 50$.

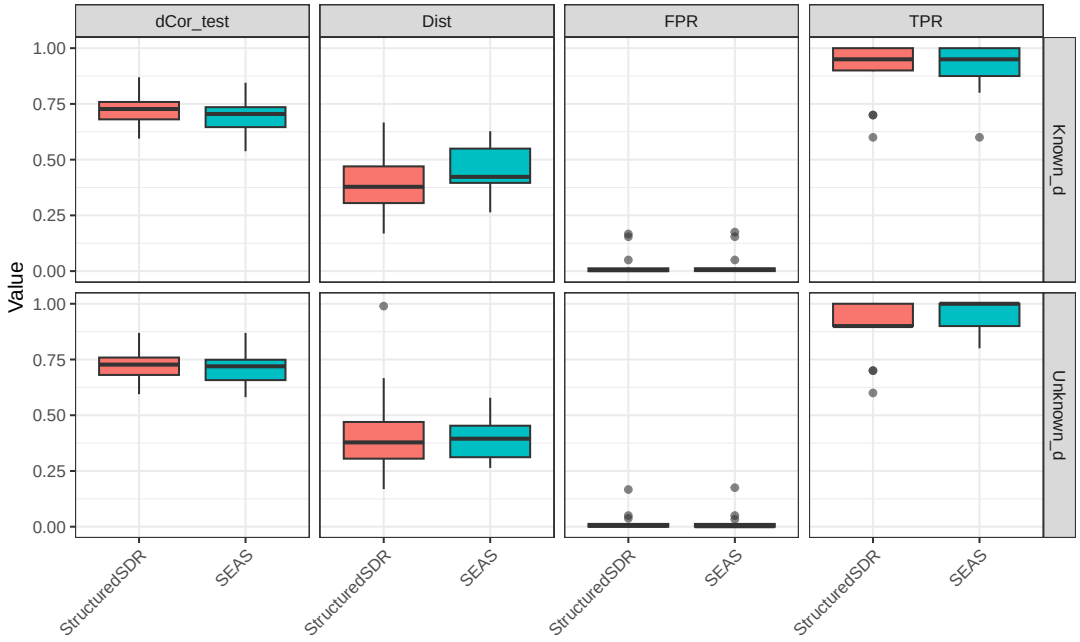

Figure 1: Test performance for M1

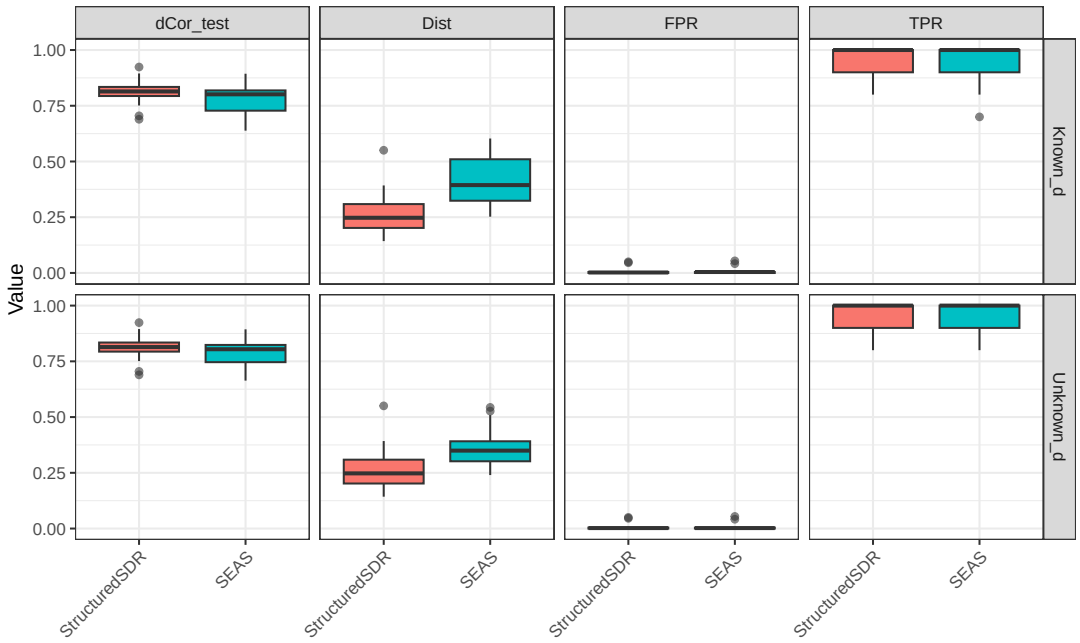

Figure 2: Test performance for M2

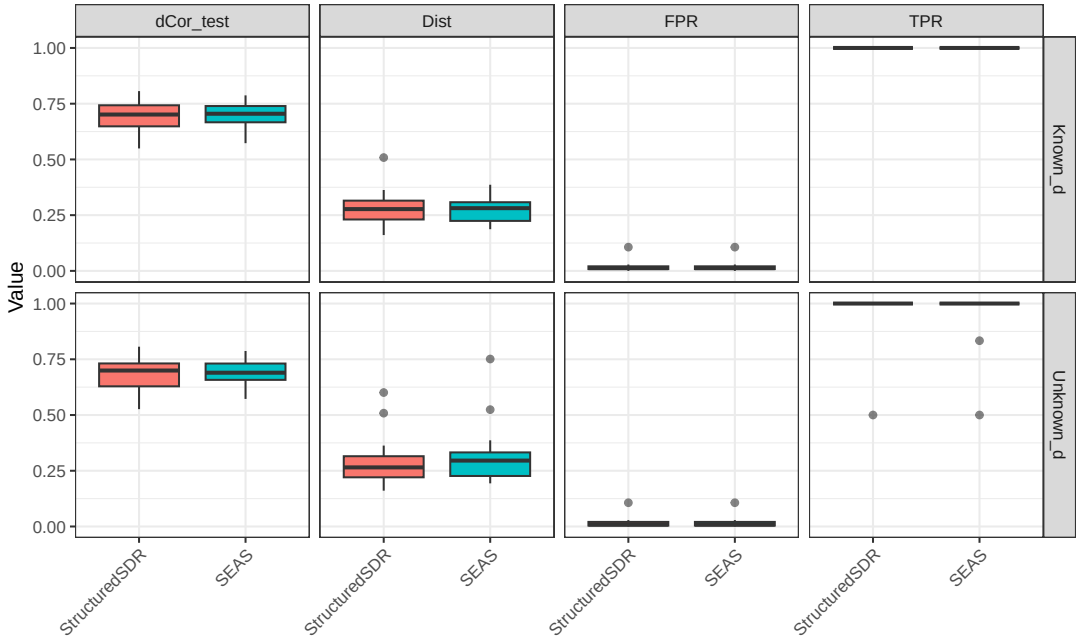

Figure 3: Test performance for M3a

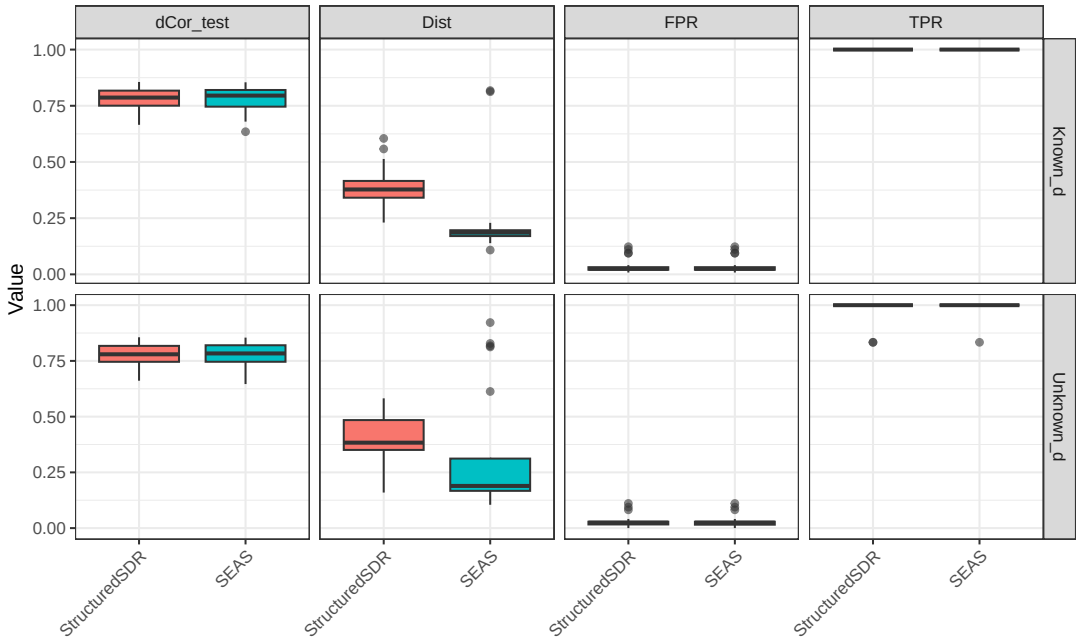

Figure 4: Test performance for M3b

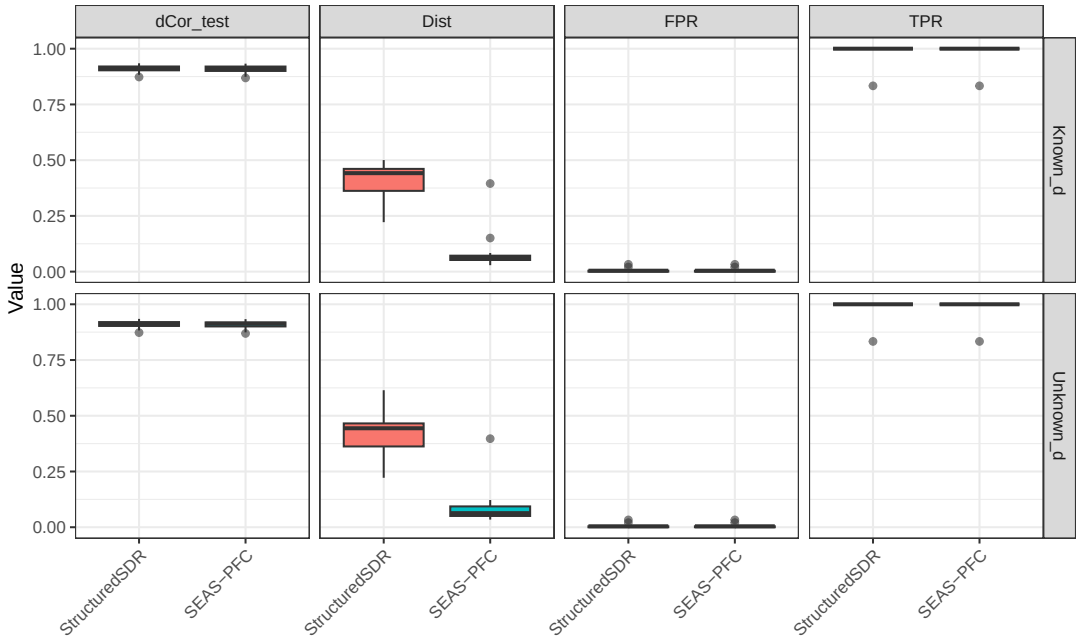

Figure 5: Test performance for M4

2.3 Analysis and discussion

2.3.1 Summary of empirical findings

(1) Single-index models (M1/M2): advantages of StructuredSDR become more pronounced in nonlinear settings. For M1 (linear single-index), the two methods perform similarly overall. Under Known-d, StructuredSDR achieves a smaller subspace error (Dist 0.394 vs. 0.451 in Table 1) and a slightly higher test dCor (0.721 vs. 0.701). Under Unknown-d, SEAS-SIR has a modest advantage in Dist and TPR (Dist 0.397 vs. 0.413, TPR 95.0% vs. 91.0%), while StructuredSDR attains a slightly higher test dCor (0.723 vs. 0.715). These patterns suggest that, in a linear model where SIR is well aligned with the central subspace, both methods can stably recover the relevant directions, and the differences are mainly in subspace accuracy and how closely they match the validation criterion (dCor).

In M2 (nonlinear single-index), StructuredSDR shows a clear and robust advantage. Under both Known-d and Unknown-d, Dist is substantially smaller (0.273 vs. 0.413 or 0.273 vs. 0.362), test dCor is higher (0.811 vs. 0.779 or 0.811 vs. 0.792), and TPR is also higher (95.5% vs. 93.0%). This is consistent with the role of dCor as a tuning and evaluation criterion that is sensitive to nonlinear dependence. When $y = \sinh(x^\top \beta^\star) + \varepsilon$, the "row-sparse + orthogonality" structure of StructuredSDR appears to favor directions that better preserve the nonlinear dependence, leading to lower subspace error and stronger dependence preservation in this nonlinear single-index setting.

(2) Multi-index models (M3a/M3b): StructuredSDR is competitive in elliptical settings but loses subspace precision under non-elliptical mixtures. In M3a (elliptical distribution), SEAS-SIR has a slight edge under Known-d: Dist is smaller (0.273 vs. 0.283)

and dCor is higher (0.696 vs. 0.689). Under Unknown-d, StructuredSDR attains a smaller Dist (0.289 vs. 0.314), while dCor is slightly lower (0.680 vs. 0.689), and the method tends to slightly underestimate the dimension (average $\hat{d} \approx 1.85$). This suggests that, in multi-index problems, the dCor-based tuning of StructuredSDR may favor somewhat lower-dimensional representations that trade a small loss in dCor for potentially more stable generalization, producing a mild tension between Dist and dCor.

In M3b (non-elliptical Gaussian mixture), SEAS-SIR is clearly stronger in subspace recovery. Under Known-d, Dist is 0.242 for SEAS-SIR compared with 0.387 for StructuredSDR; under Unknown-d, the pattern persists (0.329 vs. 0.402). However, the test dCor values are very similar for the two methods and both are close to 0.775. This indicates that, under non-elliptical distributions, StructuredSDR still learns a low-dimensional representation that captures most of the dependence between X and y (as reflected in dCor), but the recovered directions deviate more from the true central subspace in geometric terms. A natural interpretation is that non-elliptical distributions distort the ability of slice means (the SIR statistics) to identify the true subspace. In this setting, the alternating row-sparse optimization of StructuredSDR may be more exposed to statistical bias or local minima, while SEAS yields more robust subspace estimates.

(3) Inverse regression model (M4): structural advantage of SEAS-PFC in subspace recovery. Model M4 follows the structure of a PFC (principal fitted components) inverse model. SEAS-PFC is tailored to this generative mechanism through its objective and regularization, and it shows a strong advantage in Dist. Under Known-d, Dist is 0.080 for SEAS-PFC compared with 0.407 for StructuredSDR; under Unknown-d, the numbers are 0.086 and 0.416, respectively (Table 5). At the same time, the two methods have nearly identical TPR, FPR, and test dCor (around 0.909–0.910). This indicates that, although StructuredSDR does not match the true PFC directions as closely in geometric terms, it still extracts a low-dimensional representation $X\hat{\beta}$ that maintains a strong dependence with y. If the primary goal is to recover the true central subspace up to its projection geometry, methods designed for the inverse model structure (such as SEAS-PFC) are preferable.

(4) TPR/FPR under aligned sparsity budgets: differences arise from which variables are selected rather than how many. Because all experiments enforce top-k truncation with the same $k = |\widehat{S}_{\text{struct}}|$, the two methods often exhibit very similar or even identical TPR and FPR (for instance, in M2 under both Known-d and Unknown-d). This shows that, under a common sparsity budget, their ability to identify the set of signal variables is comparable. In this regime, performance differences are reflected primarily in the subspace geometry (Dist) and dependence preservation (dCor), which highlights more directly the differences in the quality of the estimated directions.

2.3.2 Answer to the research question based on the results

The central question of this project is whether, in high-dimensional sparse settings, one can construct a StructuredSDR method that takes SIR statistics as input, incorporates a tractable row-sparse structure, and stably outputs a dimension-reduction subspace estimate, thereby serving as a viable alternative to the SEAS family.

The simulation results support an affirmative answer. In the classical single-index models (M1/M2) and in the elliptical multi-index model (M3a), StructuredSDR consistently recovers statistically meaningful low-dimensional representations. In several cases it achieves smaller subspace errors and higher dependence preservation, with particularly clear advantages in the nonlinear single-index model M2. Together with the discussion in Section 2.1 on warm start and post-refit, the empirical behavior of StructuredSDR can be summarized as follows. First, the warm-start scheme (ridge + SVD, combined with λ-path initialization) improves the numerical stability of the alternating nonconvex optimization, leading to more consistent convergence to effective directions across regularization strengths and replicates. Second, the post-refit step reduces the shrinkage bias induced by the penalty once the support has been identified, which in turn strengthens test-set dependence (dCor) between $X\hat{\beta}$ and y and, in some models, simultaneously improves Dist.

At the same time, the experiments clarify the limits of this implementation. In the non-elliptical mixture model M3b and in the inverse regression model M4, StructuredSDR is weaker than SEAS-SIR or SEAS-PFC in terms of geometric subspace recovery, although its test dCor remains competitive. This indicates that StructuredSDR still learns low-dimensional representations that are useful for prediction or dependence preservation, but that it is less reliable when the goal is precise recovery of the central subspace geometry under model structures that are better matched by specialized SEAS variants.

Overall, the project establishes, through a reproducible Monte Carlo framework, that StructuredSDR is a feasible algorithmic realization of structured sparse SDR. The cross-model comparison further delineates its strengths (most notably in nonlinear single-index settings) and its limitations (non-elliptical mixtures and inverse models), thereby providing an empirical answer to the research question posed at the outset.

3 Conclusion

3.1 Summary of findings

This project focuses on a central challenge in high-dimensional sufficient dimension reduction (SDR): when the ambient dimension p is large but the effective signal is confined to a small subset of variables, how to achieve both subspace recovery and variable selection, while still preserving the explanatory power of the reduced representation for nonlinear regression relationships. To address this, we propose and implement a new method, StructuredSDR. The method takes the SIR-based statistics $(\hat{\Sigma}, \hat{U})$ as a unified input, explicitly introduces a struc-

tured factorization and row-sparsity constraint in the matrix-valued direction estimator, and uses alternating optimization to obtain an estimate of the central subspace. The method is then compared in a parallel fashion with the SEAS family on the standard simulation models (M1–M4) from the SEAS paper.

From a methodological perspective, the main characteristics of StructuredSDR relative to SEAS can be summarized as follows:

- StructuredSDR adopts a more direct structural modeling strategy. Through the factorization $B = WV^\top$, it decouples "low-dimensional subspace" and "variable selection" at the parameter level: V controls the orthogonal structure of the subspace directions, while row-sparsity of W corresponds directly to variable selection. In contrast, SEAS imposes both row-sparsity and nuclear-norm penalties directly on B, and uses ADMM to iterate over split variables, which leads to tighter coupling between structure and computation.

- The definition of the output and post-processing steps in StructuredSDR is more transparent. The support is defined via a threshold on the row norms of W, and the resulting subspace basis can be directly orthogonalized for distance and dCor evaluation. For SEAS, the support is defined from the output $\hat{\beta}$ (after elementwise thresholding). In the experiments we additionally adopt a sparsity-budget alignment via top-k truncation, so that both methods are evaluated under the same support size. This avoids distortions in FPR that would arise simply because one method tends to select denser models.

- The implementation emphasizes reproducibility and numerical stability. The full StructuredSDR algorithm uses warm start and post-refit as key numerical strategies: warm start provides a stable and directionally reasonable initial point and is propagated along the λ-grid; post-refit performs debiased refitting on the selected support to mitigate shrinkage bias from the penalty. These components improve the stability of high-dimensional nonconvex alternating optimization across Monte Carlo replicates.

The simulation results show that the performance of StructuredSDR exhibits clear model-dependent patterns.

- In single-index settings (M1/M2), StructuredSDR is highly competitive, and in the nonlinear single-index model M2 it performs particularly well. It achieves markedly smaller subspace error (Dist) and higher test dCor, indicating that it is more effective in recovering the central subspace and preserving the dependence structure between the response and covariates.

- In multi-index settings (M3a/M3b), the two methods are similar under elliptical distributions (M3a), while in the non-elliptical mixture model (M3b) the Dist for StructuredSDR is clearly larger. This suggests that when the covariate distribution deviates from typical SIR assumptions, estimators based on the same $(\hat{\Sigma}, \hat{U})$ input become more sensitive to the underlying distributional conditions.

- In the inverse regression model (M4), SEAS-PFC is structurally better aligned with the model and therefore has a clear advantage in subspace recovery (Dist). However, the two methods are close in terms of TPR, FPR, and dCor, showing that StructuredSDR remains usable at the level of dependence preservation, even though its directional accuracy is inferior to the specialized PFC variant.

Overall, by proposing and implementing StructuredSDR, this project demonstrates that an SDR approach based on structured factorization, row-sparsity, and a computable alternating optimization scheme is feasible within the standard simulation framework. In some representative tasks, especially in the nonlinear single-index case, StructuredSDR attains better subspace recovery and dependence preservation. At the same time, the results on M3b and M4 reveal performance boundaries under distributional deviations and inverse regression structures, providing concrete empirical guidance for future methodological refinements.

3.2 Open questions and directions for future work

This project has completed the construction of the method, an implementable algorithm, and parallel simulation studies on M1–M4. Several directions remain for further investigation.

1. A more systematic characterization of the range of applicability. The current results suggest that StructuredSDR is more advantageous in single-index models and under relatively standard SIR conditions, while non-elliptical distributions and inverse models are more challenging. Future work can design more finely stratified experiments along dimensions such as degree of distributional deviation, signal-to-noise ratio, correlation strength, and the number of slices, in order to summarize clearer empirical rules for when StructuredSDR should be preferred.

2. Structured extensions tailored to inverse models. The results for M4 indicate that SEAS-PFC has a structural advantage in direction recovery for inverse models. A natural extension is to adapt the "factorization + row-sparsity" framework of StructuredSDR to the statistical inputs or objective function corresponding to PFC, thereby obtaining a structured version better matched to inverse regression settings.

3. Further standardization of numerical mechanisms. Warm start, post-refit, and sparsity-budget alignment have already improved the stability and comparability of the experiments. Future work can further standardize stopping rules, tuning strategies, and output thresholds, in order to reduce the influence of implementation details on the conclusions and to enhance reproducibility.

These directions do not change the core conclusion of this project: StructuredSDR provides an alternative high-dimensional SDR scheme to SEAS, with a more transparent structural interpretation and controllable implementation, and exhibits clear usability and distinctive behavior within the standard simulation framework.

References

[1] Jing Zeng, Qing Mai, and Xin Zhang. Subspace Estimation with Automatic Dimension and Variable Selection in Sufficient Dimension Reduction. *Journal of the American Statistical Association*, 119(545):343–355, 2024. DOI: 10.1080/01621459.2022.2118601.