Sparse Covariance Supervised Principal Component Analysis

Anonymous authors
Paper under double-blind review

Abstract

Principal component analysis (PCA) is one of the most well-studied machine learning methods of the last century. However, the principal components derived by PCA are not guaranteed to be response-informative and are usually dense, meaning they are hard to interpret in high dimensional settings. The former has led to the development of supervised PCA techniques where the response is usually incorporated in an objective function to guide informative projections and enhance predictive accuracy, while the latter to sparse PCA methods that seek to induce sparsity by shrinking non-significant variables to zero, effectively improving interpretability. Sparse supervised PCA methods seek to combine the two concepts as a means of simultaneous supervised dimensionality reduction and variable selection, but they usually depend on iteratively biconvex solutions of auxiliary objective functions, with no robust convergence guarantees and are sensitive to initialisation. In this paper, we propose a novel sparse supervised PCA method, sparse covariance supervised PCA (SCS-PCA), that seeks to trade-off prediction accuracy and sparsity performance. We impose an L_1 penalty on a supervised objective function and we employ manifold proximal gradient descent to solve the derived optimization problem, which guarantees global convergence to a stationary point. Numerical results from simulations and real-world microarray data illustrate that SCS-PCA provides competitive performance in prediction tasks and is able to select more features compared to existing supervised sparse methods.

1 Introduction

Principal component analysis (PCA) (Jolliffe, 2002) is one of the most popular data mining techniques used for dimensionality reduction, feature extraction, data visualisation etc., with applications in numerous fields such as genetics and bioinformatics (Elhaik, 2022), computer vision (Mehrabinezhad et al., 2024), chemometrics (Bin et al., 2013) and finance (Mavungu, 2023) to name a few. It projects the original high dimensional data into a lower dimensional, orthogonal subspace by considering linear combinations of the original data that preserve as much of the intrinsic variability as possible while also being uncorrelated (Jolliffe & Cadima, 2016).

By definition, PCA has two notable limitations. First, PCA is an unsupervised learning method and is not guaranteed to provide informative projections, with respect to the response variables, when the latter are available, leading to poor prediction performance in subsequent prediction models (Ritchie et al., 2019). The second shortcoming is that the loadings (coefficients) of the principal components are typically dense, meaning most or all original features contribute to each principal component, hindering interpretability, especially in high dimensional settings, where the number of features ranges from thousands to millions (Zou et al., 2006).

The former limitation has motivated the development of supervised PCA dimensionality reduction techniques, that seek to optimize a supervised objective function similar to that of PCA. As far as we are concerned, Bair & Tibshirani (2004) and Bair et al. (2006) were the first to propose a supervised PCA method. Specifically, they proposed a two-stage approach, incorporating the response during the first stage, where its correlation to the data is measured in order to define a subset of significant features. Then, a standard PCA step is performed on the created subset. However, this method does not integrate the response directly during learning. Barshan et al. (2011) addressed this issue by considering a single objective function

that uses the Hilbert-Schmidt independence criterion (HSIC) to maximise the dependence between kernels in reproducing kernel Hilbert spaces (RKHs). Both methods mentioned, however, oversee PCA's main aim which is maximising the variance explained by the projected data. Papazoglou & Yin (2025) recently proposed an ensemble method that seeks to trade-off maximising the covariance of the response and the data and the variance of the projected data. Since the two objectives are competing, the two objectives are balanced via a tunable regularisation parameter, improving interpretability and prediction accuracy. Many additional supervised PCA methods have been proposed recently, e.g. see Ritchie et al. (2019; 2020); Pascual & Yee (2022) with a good overview given in Xu et al. (2021); Chao et al. (2019).

The latter limitation is also well known and explored thoroughly in the literature, motivating the development of sparse PCA methods. The introduction of sparsity aids in improving interpretability by reducing the number of non-significant features in a dataset. For example, in DNA microarray experiments, gene expression data often consist of measurements from tens of thousands of genes, yet only a small fraction are biologically relevant for accurate classification of phenotypes for diagnosing a disease (Zhang & Deng, 2007). Principal components from PCA are linear combinations of nearly all genes, making it difficult to interpret and identify a subset of significant biomarkers. By incorporating sparsity, either in the form of a penalty (e.g. L_1) or a constraint (e.g. $\|\cdot\|_1 \leq \eta$), sparse PCA shrinks loadings of irrelevant genes to zero (Zou et al., 2006), effectively performing variable selection additionally to dimensionality reduction. The sparse principal components derived from sparse PCA are linear combinations of only a subset of genes, enhancing interpretability and potentially improving prediction accuracy. In the context of gene expression analysis, sparse PCA isolates the genes that contribute most to explaining the variance, thereby improving biological insight and facilitating the identification of key biomarkers. Many different versions of sparse PCA have been proposed over the years with a good overview found in Zou & Xue (2018) and Guerra-Urzola et al. (2021). Due to the non-convexity of the optimization problem, sparse PCA methods employ heuristic methods to provide approximate solutions to the objective function (Li & Xie, 2024) rather than solving the optimization problem directly. In our discussion, we will use the formulation in Zou et al. (2006) of sparse PCA where an elastic net penalty is imposed on traditional PCA.

Sparse methods are not limited to PCA, with similar extensions appearing, for example, in partial least squares (PLS) and linear discriminant analysis (LDA), namely sparse PLS (Chun & Keleş, 2010) and sparse LDA (Clemmensen et al., 2011) respectively. Most relevant to our work is the inclusion of sparsity into supervised PCA. Sharifzadeh et al. (2017) proposed supervised sparse PCA (SSPCA), where an L_1 norm constraint was imposed in the projection matrix of supervised PCA using HSIC (Barshan et al., 2011) to derive sparse supervised principal components using penalised matrix decomposition (PMD), while Feng et al. (2019) proposed supervised discriminant sparse PCA (SDSPCA), where discriminative information and sparsity are introduced directly into PCA, through an $L_{2,1}$ penalty directly on the principal components, for tumor classification. The latter has further been extended in Shi et al. (2020) to incorporate projected clustering with adaptive neighbours into SDSPCA to form SDSPCA with adaptive neighbours (SDSPCAAN).

SDSPCA and SDSPCAAN induce sparsity directly on the principal components rather than the loading matrix and hence cannot be used for variable selection, unlike SSPCA. Additionally, they can only be applied for binary classification tasks while SSPCA can also be applied in regression or multi-label tasks. SSPCA, on the other hand, does not address the trade-off between sparsity and prediction accuracy directly since sparsity is incorporated as a constraint rather as a penalty and also requires careful specification of a kernel for the response variables. A common feature among all three methods is the optimization scheme employed to solve the respective problem. Specifically, all sparse supervised PCA optimization problems are non-convex and non-smooth due to the orthogonality and sparsity constraints. As a result, sparse supervised PCA methods employ auxiliary objective functions to translate the original non-convex optimization problem into a biconvex problem that can be solved using iteratively alternating algorithms. Feng et al. (2019) and Shi et al. (2020) used an iteratively alternate algorithm to solve SDSPCA and SDSPCAAN respectively, while Sharifzadeh et al. (2017) solved SSPCA using PMD which has also been used to solve the sparse PCA problem (Witten et al., 2009) and is also based on an alternating algorithm. These algorithms, while simple to perform, do not provide robust theoretical guarantees for global convergence and are subject to proper initialisation and specification.

The lack of a rigorous optimization scheme in sparse methods lies in the non-convexity of the optimization problem. However, the objective function does not necessarily need be non-convex as well. Specifically, when the objective function is convex, the proximal gradient descent method (Combettes & Wajs, 2005) can be used to solve the non-smooth optimization problem as long as the objective function can be decomposed into two convex functions of which one is also smooth. However, the proximal gradient descent can be applied in Euclidean settings, while the optimization problem we consider lies in the Stiefel manifold, i.e. the set of all orthonormal p-frames in $\mathbb{R}^n$. Recent developments in manifold optimization algorithms have proposed transitioning from the Euclidean setting to manifolds while still preserving the optimal properties of proximal gradient descent. For example, a Riemannian proximal gradient method proposed in Huang & Wei (2022) was successfully employed to solve sparse PCA directly, without employing heuristic methods to approximate the solution. Huang & Wei (2022) established convergence analysis for their method, guaranteeing global convergence to a stationary point under minimum requirements. For the formulation presented in this paper, we consider a similar algorithm proposed in Chen et al. (2020) called manifold proximal gradient (ManPG), which is specifically designed to solve the non-smooth optimization problem over the Stiefel manifold, while ensuring global convergence to a stationary point when applied to our proposed optimization scheme.

In this paper, we propose a novel supervised sparse PCA method that seeks to perform simultaneous supervised dimensionality reduction and variable selection, by inducing sparsity on a supervised objective function. We solve the non-smooth optimization problem derived using manifold proximal gradient descent to establish global convergence to a stationary point. Our proposed method seeks to (1) derive interpretable supervised sparse projections, (2) establish global convergence to a stationary point for sparse supervised PCA through a robust optimization scheme, (3) identify a subset of relevant features through supervised dimensionality reduction and (4) improve prediction performance for subsequent prediction models.

The rest of the paper is organised as follows. In Section 2 we provide a brief overview of PCA, supervised PCA and sparse supervised PCA methods. We also include a brief overview of PLS and sparse PLS which we use later in our experiments as a baseline. Section 3 provides some important mathematical definitions related to our proposed method which is introduced in Section 4. Numerical experiments from simulation and real-world analyses are presented in Section 5, with Section 6 offering a closing discussion.

2 Background

In this section we explore briefly existing supervised and sparse supervised PCA based methods for dimensionality reduction. We begin with a short overview of PCA and sparse PCA and then discuss supervised PCA methods followed by sparse supervised PCA methods. The section ends with a short discussion on PLS and sparse PLS which will be used as baseline comparisons during our experiments. Throughout the paper, unless stated otherwise, we assume the following notation. We denote $X \in \mathbb{R}^{n\times p}$ the feature data matrix and Y as the $n \times k$ matrix containing the response variables. We assume that both matrices have already been centred.

2.1 PCA and Sparse PCA

PCA learns an orthogonal projection of the original data into a lower dimensional linear subspace, i.e. $Z = XW$, such that the variance of the projected data is maximized, or equivalently, the reconstruction error is minimized,

$$\min_{W:W^\top W=I_q} \|X - XWW^\top\|_F^2,$$

where $W \in \mathbb{R}^{p\times q}$ the projection matrix satisfying the orthonormality condition, $W^\top W = I_q$. The solution to PCA can be derived either by the eigenvalue decomposition of the covariance matrix $\Sigma = X^\top X$, or via singular value decomposition (SVD) directly on X. The new projections are linear combinations of the variables in the original dataset that are uncorrelated with each other (Jolliffe & Cadima, 2016). However, in (ultra-) high-dimensional settings, this can decrease interpretability significantly.

Sparse PCA (SPCA) improves interpretability by enforcing sparsity on the loadings, removing non-significant variables. Sparsity is usually integrated in the PCA framework in the form of an L_1 penalty or constraint

(Tibshirani, 1996), although other penalties can be used as well, e.g. L_0, elastic net etc.. The most notable SPCA method was proposed in Zou et al. (2006), where an elastic net penalty was utilized to produce modified principal components with sparse loadings, while maintaining computational efficiency and scalability. Zou et al. (2006) used a regression formulation for SPCA,

$$\left(\hat{P}, \hat{W}\right) := \arg\min_{P,W} \|X - XWP^\top\|_F^2 + \lambda \sum_{j=1}^{k} \|w_j\|^2 + \sum_{j=1}^{k} \lambda_{1,j} \|w_j\|_1,$$

where $P, W \in \mathbb{R}^{p\times k}$ are matrices of loadings, subject to $P^\top P = I_k$, w_j is the j-th column of W, $\lambda > 0$ is the ridge penalty and $\lambda_{1,j}$ controls sparsity for the j-th principal component. The solution is computed via an alternating algorithm, optimizing W and P iteratively.

2.2 Supervised PCA Methods

Numerous supervised PCA extensions have been proposed over the last two decades, with comprehensive reviews in Ghojogh & Crowley (2019); Xu et al. (2021). The first such method, introduced in Bair & Tibshirani (2004); Bair et al. (2006), employs a two-stage approach: selecting features correlated with the response via a threshold rule, followed by standard PCA. However, this method fails to integrate the response during learning of the projection, potentially yielding less informative projections (Ritchie et al., 2019).

Barshan et al. (2011) addressed this issue using the Hilbert-Schmidt independence criterion (HSIC) to measure the dependence between the data and the response variables in reproducing kernel Hilbert spaces (RKHs). They incorporate the labels directly during the learning stage using the following optimization problem,

$$\max_{W:W^\top W = I_q} W^\top X^\top K X W \tag{1}$$

where $K = K(Y)$ is a kernel of Y, e.g. radial basis function (RBF) kernel for regression. The solution to (1) is derived, similar to PCA, by considering the eigenvalue decomposition of the matrix $Q = XKX^\top$.

Recently, Papazoglou & Yin (2025) proposed covariance supervised PCA (CSPCA), which balances supervised relevance and interpretability by maximizing the covariance of the projected data with the response variables while also preserving the variability of the data,

$$\max_{W:W^\top W = I_q} W^\top C W, \tag{2}$$

where $C = X^\top Y Y^\top X + \kappa X^\top X$ combines the supervised and unsupervised objectives via a tunable hyperparameter $\kappa > 0$. The solution derives from the top-q eigenvectors of C. For classification tasks, the use of a delta kernel is recommended,

$$\delta(y, y') = \begin{cases} 1, \text{ if } y = y' \\ 0, \text{ if } y \neq y', \end{cases}$$

instead of $YY^\top$ to align the binary nature of the response with the model and also allow for non-linear relationships between the data and the response. CSPCA maintains data covariance structure, matches, approximately, PCA in variance explained, and outperforms other supervised PCA methods in prediction accuracy for both regression and classification tasks.

2.3 Sparse Supervised PCA Methods

The method proposed by Barshan et al. (2011) still faces the issue of generating dense supervised principal components which consist of all the original variables and hence are difficult to interpret. Using the formulation in (1), Sharifzadeh et al. (2017) imposed an L_1 norm constraint on the eigenvectors of W in (1) to induce sparsity and remove non-significant variables by defining sparse supervised PCA (SSPCA),

$$\max_{W} W^\top X K X^\top W, \text{ such that } \|W\|_1 \leq c, \tag{3}$$

where $1 \leq c \leq \sqrt{p}$ controls the sparsity. SSPCA preserves only the most relevant features improving interpretability and enhancing generalisation in high dimensional settings. To solve SSPCA, Sharifzadeh et al. (2017) used the penalised matrix decomposition (PMD) method by Witten et al. (2009) that has also been used in sparse PCA applications. SSPCA applies sparsity on the loadings and can be applied to both regression and classification settings, while also effectively performing variable selection. However, SSPCA does not solve (3) directly, instead it employs a number of approximation steps, reformulating the original objective function. It also requires careful specification of the kernel applied to the response, which can affect significantly the performance of the method. For the experiments conducted in this paper, we consider the RBF kernel for regression tasks and the delta kernel for classification tasks.

Feng et al. (2019) proposed supervised discriminative sparse PCA (SDSPCA), a sparse supervised PCA method that incorporates class labels on the PCA problem, combining discriminative information and sparsity. The major difference from existing methods is that sparsity is applied directly on the principal components instead of the loadings. Mathematically, Feng et al. (2019) defined SDSPCA as

$$\min_{W,Q,A} \|X^\top - WQ^\top\|_F^2 + \alpha\|Y - AQ^\top\|_F^2 + \beta\|Q\|_{2,1}, \quad \text{subject to } Q^\top Q = I,$$

where Q represents the sparse principal components, α and β are scale weights, A is a linear transformation matrix that maps the sparse principal components (Q) to the class labels (Y) and $\|\cdot\|_{2,1}$ corresponds to the $L_{2,1}$ norm. From (4) we can observe how sparsity (β) is imposed on the principal components rather than the loadings as is the case for SSPCA in (3). As a result, SDSPCA is not suitable for variable selection since we cannot identify non-significant features, rather principal components only. Additionally, Feng et al. (2019) did not solve (4) directly but instead defined an auxiliary objective function to optimize using $L_{2,1}$ norm optimization. The solution is then derived using an iterative algorithm that alternates between updating the loading matrix W, the transformation matrix A, the sparse principal components Q and a diagonal matrix V introduced in the auxiliary objective, complicating interpretation. In contrast to SPCA, SSPCA and the method we herein propose, the loading matrix derived from SDSPCA will not have sparse entries, since sparsity appears directly on the principal components in Q. SDSPCA can be applied only for classification settings, limiting its application spectrum. This method has further been extended to incorporate neighbour information during learning (Shi et al., 2020), but we stick to the original formulation by Feng et al. (2019).

2.4 PLS and Sparse PLS

PLS regression seeks a set of latent vectors that performs a simultaneous decomposition of the data and the response variables such that the covariance is maximised. A regression step is then performed to predict Y using the decomposition of X. Mathematically, PLS can be defined as

$$X = PU_x^\top + E_x,$$

$$Y = KU_y^\top + E_y,$$

where $P, K \in \mathbb{R}^{n\times q}$, for the q extracted components known as latent vectors, $U_x \in \mathbb{R}^{p\times q}$ and $U_y \in \mathbb{R}^{k\times q}$ are matrices of coefficients (loadings) and $E_x \in \mathbb{R}^{n\times p}$ and $E_y \in \mathbb{R}^{n\times k}$ are matrices of random errors. Sparse PLS (SPLS), proposed by Chun & Keleş (2010), extends PLS by imposing an L_1 constraint on the objective function of PLS. To formulate the SPLS optimization problem, Chun & Keleş (2010) generalised the regression formulation used in SPCA by Zou et al. (2006), imposing the L_1 penalty onto a surrogate of the direction vector instead of the original direction vector, i.e.

$$\min_{\alpha,\gamma} -\kappa\alpha^\top M\alpha + (1-\kappa)(\gamma-\alpha)^\top M(\gamma-\alpha) + \lambda_1|\gamma_1| + \lambda_2|\gamma_2| \quad \text{s.t. } \alpha^\top\alpha = 1,$$

where $M = X^\top YY^\top X$, λ_1 encourages sparsity on γ_1 and λ_2 accounts for potential singularity in M. The solution to (4) is derived by alternatively iterating between solving for α for γ fixed and solving for γ for α fixed.

3 Preliminaries

We first provide some key definitions. Throughout the PCA literature and its supervised extensions, the Frobenius norm is the most used norm as it is an essential part of the objective function of each method.

Definition 1 (Frobenius Norm)**.** *Assume an $n \times p$ real matrix Z. The Frobenius norm of Z is defined as the square root of the sum of the absolute squares of its elements,*

$$\|Z\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{p}|z_{ij}|^2} = \sqrt{tr(Z^\top Z)},$$

where $tr(\cdot)$ denotes the matrix trace operation.

In the example of PCA, the Frobenius norm is employed to calculate the reconstruction error when projecting the data onto a lower-dimensional subspace. The projection matrix used in PCA and all similar dimensionality reduction techniques is assumed to be orthonormal, or equivalently, it lies on the Stiefel manifold. The Stiefel manifold is an example of a smooth manifold. As noted by Lee (2012), a smooth manifold is a space that "locally looks like" $\mathbb{R}^n$ and we can perform calculus on it.

Definition 2 (Stiefel Manifold)**.** *The Stiefel manifold, herein denoted as $\mathcal{S}(p,q)$, is defined as the space consisting of all $p \times q$ orthonormal matrices,*

$$\mathcal{S}(p,q) = \{W \in \mathbb{R}^{p\times q} : W^\top W = I_q\}.$$

On a differential manifold, such as the Stiefel manifold, an important notion is the tangent space.

Definition 3 (Tangent Space)**.** *Let $\mathcal{M}$ be a smooth manifold and let $p \in \mathcal{M}$. The tangent space at p, denoted as $\mathcal{T}_p\mathcal{M}$, is the set of all possible derivatives of smooth curves passing through p. For the Stiefel manifold, this can be defined as*

$$\mathcal{T}_W\mathcal{S}(p,q) = \{V \in \mathbb{R}^{p\times q} : V^\top W + W^\top V = 0\}.$$

A recurrent term in manifold theory is that of the Riemannian gradient. First, recall that the Euclidean gradient, or standard gradient, is the vector consisting of the partial derivatives of a real function $f : \mathbb{R}^n \to \mathbb{R}$ with respect to each coordinate direction. The Riemannian gradient generalizes this concept to manifolds.

Definition 4 (Riemannian Gradient)**.** *Consider a smooth function $f : \mathcal{M} \to \mathbb{R}$, where $\mathcal{M}$ is a smooth manifold. The Riemannian gradient, denoted $grad_f$, is defined as the unique tangent vector in $\mathcal{T}_x\mathcal{M}$ satisfying*

$$\langle grad_f(x), v\rangle_x = Df(x)[v], \quad \forall v \in \mathcal{T}_x\mathcal{M},$$

where $Df(x)[v]$ is the directional derivative of f at x in the direction v.

When solving optimization problems constrained on a manifold it is crucial to ensure that each iterative step remains on the manifold regardless of the operation taking place. Retraction ensures that tangent vectors are "retracted" back to the manifold, ensuring feasibility of iterates.

Definition 5 (Retraction)**.** *A continuous map $r : X \to A$, where $A \subseteq X$ is called a retraction if and only if $r(\alpha) = \alpha$ for all $\alpha \in A$.*

Sparse methods that make use of the L_1 norm, e.g. LASSO regression, face the issue of non-smoothness since the L_1 norm is non-differentiable. To overcome this issue we make use of proximal mappings, which are well-defined under mild assumptions.

Definition 6 (Proximal Mapping)**.** *Let f denote a closed, convex function. The proximal mapping associated to f is defined as*

$$prox_f(Y) := \arg\min_X f(X) + \frac{1}{2}\|X - Y\|_F^2.$$

Theorem 1 (Proximal Mapping Existence-Uniqueness)**.** *If f is additionally lower- semicontinuous, then the proximal mapping, $prox_f$, is well-defined and is unique for all $Y \in \mathbb{R}^{n\times m}$.*

The proof for this Theorem is provided in the Appendix. The objective function we propose belongs in the category of convex but non-differentiable functions. As a result, the application of gradient descent becomes infeasible. Instead, we consider proximal gradient descent, specifically over the Stiefel manifold, that seeks to optimize a non-differentiable but decomposable function. The update rule for proximal gradient descent in the Euclidean setting (Combettes & Wajs, 2005) is provided in the following definition.

Definition 7 (Proximal Gradient Step)**.** *Assume f is a decomposable function, $f(X) = g(X) + h(X)$, where g is convex and differentiable and h is convex, non-differentiable. The proximal gradient update rule is defined as*

$$X_{k+1} := \arg\min_{Z} g(X_k) + \langle \nabla g(X_k), Z - X_k \rangle + \frac{1}{2t}\|Z - X_k\|_F^2 + h(Z), \tag{4}$$

where $t > 0$ is a stepsize parameter.

The update rule of the proximal gradient descent can be equivalently written in terms of the proximal mapping as

$$X_{k+1} = \text{prox}_{th}\left(X_k - t\nabla g(X_k)\right).$$

4 Sparse Covariance Supervised PCA (SCS-PCA)

In this section, we present our proposal for sparse supervised PCA which we call sparse covariance supervised PCA or sparse CSPCA (SCS-PCA). First, we discuss the motivation behind our method before proceeding with the mathematical formulation of SCS-PCA. Next, we present the optimization algorithm we employ to derive the projection matrix with sparse loadings and discuss the convergence of it. The section concludes with some practical considerations.

4.1 Motivation

The method we propose herein seeks to provide a direct trade-off between predictive accuracy and sparsity performance in a single objective function via a tunable hyperparameter and it is accompanied by a robust optimization scheme that provides theoretical guarantees for global convergence to a stationary point. The main innovation of our proposal lies in the direct solution of the objective function without relying on auxiliary objective functions, through the use of manifold optimization that confronts the non-convex optimization problem directly, without resorting to biconvex formulations. Specifically, we apply an L_1 penalty on the loadings of the projection matrix seeking to trade-off CSPCA's predictive performance and sparsity induced by the lasso penalty. To achieve this, we introduce a tunable parameter that controls the magnitude of sparsity imposed on CSPCA's objective function and hence controls the shrinkage of the loadings. Following CSPCA, our method can be applied for any type of response variable and does not require kernel tuning. The derived optimization problem is non-convex and non-smooth. We employ manifold proximal gradient descent (ManPG) (Chen et al., 2020) which can handle non-convexity and non-smoothness simultaneously by decomposing the objective function into a smooth and non-smooth function. The ManPG algorithm is an extension of traditional proximal gradient descent onto the Stiefel manifold, where the projection matrix of the proposed optimization problem lies. We provide theoretical guarantees for the global convergence of the algorithm to a stationary point.

4.2 Mathematical Formulation

Our proposal is based on CSPCA's formulation presented in (2). We introduce a slightly different notation to the optimization problem by specifying explicitly that the projection matrix subsides in the Stiefel manifold,

$$\max_{W \in S(p,q)} \text{tr}\left(W^\top C W\right),$$

where $C = X^\top Y Y^\top X + \kappa X^\top X$ defined as before. For classification tasks, a delta kernel, $\delta(y, y')$ is used to model the response instead of $YY^\top$ in the objective function. To induce sparsity, we apply an L_1 penalty (Tibshirani, 1996) directly on the loadings of the projection matrix, which allows for a direct trade-off between prediction accuracy and sparsity. This way we expect a large number of non-significant features is

eliminated from the projected data, through shrinkage, enhancing interpretability. The sparse optimization problem can thus be defined as

$$\max_{W \in S(p,q)} \operatorname{tr}\left(W^\top C W\right) - \eta \|W\|_1, \tag{5}$$

where $\eta > 0$ is the parameter controlling the magnitude of sparsity. The proposed method can be applied for any type of response variable and applies sparsity directly on the loadings in the form of an L_1 penalty rather a constrained optimization problem. Unlike existent methods, we seek to optimize (5) directly without using auxiliary objectives or approximating reformulations. Specifically, we employ the ManPG algorithm, which combines manifold optimization and proximal gradient descent theory to simultaneously handle the orthogonality and sparsity constraints and prove it leads to global convergence on a stationary point.

4.3 Optimization

Since the objective function in (5) is non-differentiable, proximal gradient descent algorithms are required to derive the projection matrix with sparse loadings. Hence we need to decompose the objective into a smooth, convex part and a non-smooth but convex part to define a problem of the form,

$$\min F(W) := f(W) + h(W),$$

where f is smooth and convex and h is convex, non-smooth. By rewriting (5) as a minimisation problem,

$$\min_{W \in S(p,q)} F(W) := -\operatorname{tr}(W^\top C W) + \eta \|W\|_1,$$

we can observe that $f(W) = -\operatorname{tr}(W^\top C W)$ is smooth and convex, while $h(W) = \eta\|W\|_1$ is convex and non-smooth, since h is non-differentiable at 0. The Euclidean gradient of the smooth part can easily be calculated as

$$\nabla f(W) = -2CW = -2(X^\top Y Y^\top X + \kappa X^\top X)W.$$

Using the proximal gradient update rule from Definition 7 on (5), we derive

$$W_{k+1} = \arg\min_Z -\operatorname{tr}(W_k^\top C W_k) + \langle -2CW_k, Z - W_k\rangle + \frac{1}{2t}\|Z - W_k\|_F^2 + \eta\|Z\|_1, \tag{6}$$

where $t > 0$ is a stepsize parameter. Since W lies on the Stiefel manifold, $\mathcal{M} = \mathcal{S}(p,q)$, we need to ensure that the descent direction lies in the tangent space, $\mathcal{T}_W\mathcal{M}$, hence we need to employ a manifold proximal gradient descent algorithm instead. Chen et al. (2020) propose the following subproblem to transition from the Euclidean setting to the Stiefel manifold,

$$V_k := \arg\min_V \langle \operatorname{grad} f(W_k), V\rangle + \frac{1}{2t}\|V\|_F^2 + \eta\|W_k + V\|_1, \tag{7}$$

such that $V \in \mathcal{T}_{W_k}\mathcal{M}$, where $\operatorname{grad} f$ is the Riemannian gradient. By the definition of the Riemannian gradient, we have

$$\langle \operatorname{grad} f(W_k), V\rangle = \langle \nabla f(W_k), V\rangle = \langle -2CW_k, V\rangle,$$

$\forall\ V \in \mathcal{T}_{W_k}\mathcal{M}$, which allows to rewrite the subproblem in (7) as

$$V_k := \arg\min_V \langle -2CW_k, V\rangle + \frac{1}{2t}\|V\|_F^2 + \eta\|W_k + V\|_1, \tag{8}$$

such that $V \in \mathcal{T}_{W_k}\mathcal{M}$. As a result, there is no need to calculate the Riemannian gradient, but the Euclidean. The corresponding projection matrix at step $k+1$ can be derived by retraction,

$$W_{k+1} = \operatorname{Retr}(\alpha V_k),$$

where α is determined by an Armijo line search. Popular choices of retraction for the Stiefel manifold include the exponential mapping, the polar decomposition, the QR decomposition and the Caley transformation.

4.4 Solving SCS-PCA using RSSN

To solve a subproblem of the form as in (8), Chen et al. (2020) suggested the use of the Regularised Semi-Smooth Newton (RSSN) method, which, applied to SCS-PCA, is guaranteed to converge. First, we denote $A_k(V) = V^\top W_k + W_k^\top V$, and then rewrite the subproblem in (7) as

$$V_k := \arg\min_V \langle -2CW_k, V\rangle + \frac{1}{2t}\|V\|_F^2 + \eta\|W_k + V\|_1$$

such that $A_k(V) = 0$. Let $\mathcal{L}(V;\Lambda)$ denote the Langrangian function of the subproblem's objective function,

$$\mathcal{L}(V;\Lambda) = \langle -2CW_k, V\rangle + \frac{1}{2t}\|V\|_F^2 + \eta\|W_k + V\|_1 - \langle A_k(V), \Lambda\rangle.$$

To solve the subproblem, we define the Karush-Kuhn-Tucker (KKT) conditions,

$$0 \in \partial_V \mathcal{L}(V;\Lambda) \quad \text{and} \quad A_k(V) = 0.$$

The proximal mapping point of $h = \eta\|\cdot\|_1$ at point W_k is defined as

$$\text{prox}_{t\eta\|\cdot\|_1}(W_k) = \arg\min_Z \frac{1}{2}\|Z - W_k\|_F^2 + t\eta\|Z\|_1. \tag{9}$$

Using the proximal mapping we can write

$$V(\Lambda) = \text{prox}_{th}(B(\Lambda)) - W_k,$$

where $B(\Lambda) = W_k + t(2CW_k - A_k^*(\Lambda))$, and $A_k^*(\Lambda)$ is the adjoint operator of A_k.

In the case of the L_1 norm, the proximal mapping is defined as the soft-thresholding operator, hence there is no need to solve (9). The soft-thresholding operator is defined as

$$\text{prox}_{t\eta\|\cdot\|_1}(Y)_{ij} = \text{sign}(Y_{ij})\max(|Y_{ij}| - t\eta, 0).$$

We use the RSSN method to solve $\mathcal{E}(\Lambda) = A_k(V(\Lambda)) = 0$ to derive the k-th iterate of the subproblem, V_k. Specifically, we compute the Newton direction d by solving

$$(G(\Lambda) + \rho I)d = -\text{vec}(\mathcal{E}(\Lambda)),$$

where $G(\Lambda)$ is a representation of the generalised Jacobian of $\mathcal{E}$ and $\rho > 0$ is a regularisation parameter. The updates for Λ are calculated using the following rule

$$\Lambda_{k+1} \leftarrow \Lambda_k + d.$$

Thus, we can easily solve the subproblem and derive V_k. We can then update W_k using a retraction step, i.e. $W_{k+1} = \text{Retr}(\alpha V_k)$, where α is determined by an Armijo line search. An overview of the steps to solve SCS-PCA is given in Algorithm 1.

Algorithm 1 SCS-PCA using ManPG Algorithm

Require: Initial $W_0 \in \mathcal{M}$, $\gamma \in (0,1)$, stepsize $t > 0$
Ensure: Projection matrix W.
1: **for** $k = 0, 1, \ldots$ **do**
2: Obtain V_k by solving the subproblem,
$V_k := \arg\min_{V \in \mathcal{T}_{W_k}\mathcal{M}} \langle -2CW_k, V\rangle + \frac{1}{2t}\|V\|_F^2 + \eta\|W_k + V\|_1$, where $C = X^\top YY^\top X + \kappa X^\top X$.
3: Set $\alpha = 1$
4: **while** $F(\text{Retr}_{W_k}(\alpha V_k)) > F(W_k) - \frac{\alpha\|V_k\|_F^2}{2t}$ **do**
5: $\alpha \leftarrow \gamma\alpha$
6: **end while**
7: Update the iterate using retraction:
$W_{k+1} = \text{Retr}_{W_k}(\alpha V_k)$.
8: **end for**

4.5 Convergence of SCS-PCA and practical considerations

The proximal gradient method converges globally with rate $\mathcal{O}(1/k)$ if and only if both parts of the objective function are convex and additionally the gradient of the smooth part, i.e. the Euclidean gradient ∇f, is Lipschitz continuous (Parikh et al., 2014). This also extends to SCS-PCA as indicated by the following Theorem.

Theorem 2 (SCS-PCA Convergence)**.** *Under these assumptions, Algorithm 1 will also converge globally to a stationary point.*

The detailed proof is provided in the Appendix. The Euclidean gradient of the smooth part is defined as

$$\nabla f(W) = -2CW = -2X^\top YY^\top XW - 2\kappa X^\top XW,$$

and we can easily show that ∇f is indeed Lipschitz continuous since

$$\|\nabla f(W_1) - \nabla f(W_2)\|_F = \| - 2CW_1 + 2CW_2\|_F = 2\|C(W_1 - W_2)\|_F \leq 2\|C\|_{\text{op}}\|W_1 - W_2\|_F,$$

where $\|C\|_{\text{op}}$ is the operator norm corresponding to the largest singular value of the symmetric and positive semi-definite matrix C. Following the definition of Lipschitz continuous function in Section 3, we have $L = 2\|C\|_{\text{op}}$ as the Lipschitz constant which is finite since C is positive semi-definite. Thus, ∇f is indeed Lipschitz continuous and hence Algorithm 1 converges globally to a stationary point for SCS-PCA.

In practice, we suggest to initialise Algorithm 1 using the solution to the standard CSPCA which consists of the top q eigenvectors corresponding to the top q eigenvalues of C. The computational burden of the algorithm lies in solving the subproblem to obtain V_k, while the RSSN method for finding the descent direction performs robustly and efficiently (Chen et al., 2020).

4.6 Tuning the sparsity parameter

The proposed formulation in (5) consists of two parameters, the balancing parameter, κ and the sparsity inducing parameter η. CSPCA is not very sensitive to the choice of κ, and large fluctuations are needed to observe a significant difference. We suggest tuning κ at the initialisation step where W_0 is taken as the standard CSPCA solution and use this value as fixed. For η, we suggest the use of K-fold cross-validation. Algorithm 2 outlines the tuning procedure.

Algorithm 2 K-fold Cross-Validation for Tuning the Sparsity Parameter η

Require: Data (X, Y), candidate values $\{\eta_1, \ldots, \eta_L\}$, number of folds K
Ensure: Optimal sparsity parameter η^*
Partition data into K equal folds
for $i = 1, \ldots, L$ **do** ▷ Loop over grid of η values
  Set $\eta = \eta_i$
  Initialize validation error list: $E_i = []$
  for $k = 1, \ldots, K$ **do** ▷ K-fold cross-validation
    Use fold k as validation set; remaining $K-1$ folds as training set
    Train Algorithm 1 on training data with current η
    Compute prediction $\hat{Y}_{\text{val}}$ on validation set
    Compute MSE: $e_k = \frac{1}{n_k}\|Y_{\text{val}} - \hat{Y}_{\text{val}}\|_F^2$
    Append e_k to E_i
  end for
  Compute mean validation error: $\bar{E}_i = \frac{1}{K}\sum_{j=1}^{K} E_i[j]$
end for
Select $\eta^* = \arg\min_{\eta_i} \bar{E}_i$

Table 1: Mean squared error (MSE) and number of non-zero variables (Non-zero) with standard errors for $q = 2, 3$, and 4 components for Simulation 1.

Method	**2 Components**		**3 Components**		**4 Components**	
	MSE	**Non-zero**	**MSE**	**Non-zero**	**MSE**	**Non-zero**
			I.I.D. Scenario			
PCA	0.9517 (0.0681)	500.0 (0.0)	0.9409 (0.0703)	500.0 (0.0)	0.9479 (0.0720)	500.0 (0.0)
PLS	0.8499 (0.0615)	500.0 (0.0)	0.8438 (0.0610)	500.0 (0.0)	0.8448 (0.0614)	500.0 (0.0)
HSIC	0.9471 (0.0694)	499.2 (0.3)	0.9403 (0.0695)	499.1 (0.2)	0.9289 (0.0681)	499.1 (0.1)
Bair	0.9259 (0.0645)	222.4 (2.9)	0.9168 (0.0632)	222.4 (2.9)	0.9184 (0.0641)	222.4 (2.9)
CSPCA	0.8735 (0.0654)	499.2 (0.3)	0.8743 (0.0656)	499.2 (0.2)	0.8749 (0.0652)	499.2 (0.2)
SPLS	0.3960 (0.0923)	71.1 (31.9)	0.4351 (0.0834)	90.3 (17.2)	0.5794 (0.0903)	182.1 (33.5)
SPCA	0.9780 (0.0660)	34.3 (1.5)	0.9980 (0.0749)	33.4 (1.0)	0.9569 (0.0725)	33.8 (0.9)
SSPCA	0.9356 (0.0652)	127.0 (37.2)	0.9270 (0.0796)	58.2 (20.2)	0.9266 (0.1062)	61.1 (24.6)
SCS-PCA	0.5579 (0.0733)	35.1 (7.0)	0.5779 (0.0719)	26.1 (4.9)	0.5461 (0.0650)	16.7 (2.7)
			Correlated Scenario			
PCA	1.0219 (0.1384)	500.0 (0.0)	1.0215 (0.1400)	500.0 (0.0)	1.0134 (0.1397)	500.0 (0.0)
PLS	1.1740 (0.1385)	500.0 (0.0)	1.1936 (0.1393)	500.0 (0.0)	1.2020 (0.1440)	500.0 (0.0)
HSIC	1.0488 (0.1423)	499.1 (0.1)	1.0503 (0.1423)	499.0 (0.1)	1.0456 (0.1413)	499.1 (0.1)
Bair	1.0717 (0.1280)	224.1 (3.6)	1.1037 (0.1350)	224.1 (3.6)	1.1073 (0.1294)	224.1 (3.6)
CSPCA	1.1282 (0.1322)	499.1 (0.2)	1.1288 (0.1321)	499.2 (0.2)	1.1288 (0.1317)	499.2 (0.1)
SPLS	1.0021 (0.0594)	67.2 (16.5)	1.1819 (0.1121)	164.2 (54.2)	1.1696 (0.1531)	179.5 (51.4)
SPCA	1.0210 (0.1300)	25.6 (0.9)	1.0428 (0.1390)	27.8 (0.7)	1.0351 (0.1378)	27.1 (1.0)
SSPCA	1.0787 (0.1586)	83.0 (47.3)	1.0926 (0.1539)	78.0 (29.9)	1.0853 (0.1449)	73.2 (33.2)
SCS-PCA	0.9461 (0.0911)	44.6 (10.1)	1.0161 (0.1187)	39.6 (5.3)	1.0183 (0.1146)	31.9 (4.2)

5 Experimental Results

5.1 Simulations

We conduct a series of simulation analyses to examine the performance of SCS-PCA against existing methods in sparse supervised PCA, supervised PCA as well as state-of-the-art dimensionality reduction methods such as PCA and PLS. Specifically, for supervised PCA, we consider Bair's method Bair et al. (2006), supervised PCA using HSIC Barshan et al. (2011) and CSPCA Papazoglou & Yin (2025), while for sparse supervised PCA we consider, sparse PLS (SPLS) Chun & Keleş (2010), sparse PCA (SPCA) Zou et al. (2006) and sparse supervised PCA (SSPCA) Sharifzadeh et al. (2017). We define as evaluation criteria the mean squared error (MSE) and the number of non-zero variables, as an indicator of variable selection performance by each method. For parameter tuning, cross validation was adopted with MSE as optimization metric.

We consider three different simulation models. In each setting, we generate a dataset with $n = 100$ observations and $p = 500$ features, where we assume each true underlying model depends only on the first four features, creating a high-dimensional sparse setting. The specific models are defined as

- Simulation 1: $Y = 3X_1 - 2X_2 - 5X_3 + 4X_4 + \epsilon$,
- Simulation 2: $Y = e^{X_1} + 4\sin(X_2) - 3X_3 + X_4\epsilon$,
- Simulation 3: $Y = X_1^2 + 3\sin(X_2) - e^{(X_3+1)} + \frac{1}{1+X_4} + \epsilon$,

where $\epsilon \sim N(0, 0.1^2)$ corresponds to random noise. For the data generating mechanism of all 3 simulations we consider two scenarios. First, the data are independently and identically distributed (i.i.d.) from a standard normal distribution, i.e. $X \sim N(0, I_p)$. Second, we introduce correlation between features to align closer with realistic scenarios, e.g. microarray data. Specifically, we generate the data from a normal distribution with mean zero and covariance matrix Σ, defined as a Toeplitz matrix, that is $\Sigma = \Sigma_{ij} = \rho^{|i-j|}$. We set $\rho = 0.7$, inducing strong correlation between features, such that the correlation between X_i and X_j decays exponentially with $|i-j|$. All simulations were performed for $q = 2, 3$ and 4 components and for each dataset, $60\% - 20\% - 20\%$ splits into training, validation and test sets were performed.

Table 2: Mean squared error (MSE) and number of non-zero variables (Non-zero) with standard errors for $q = 2, 3$, and 4 components for Simulation 2.

Method	**2 Components**		**3 Components**		**4 Components**	
	MSE	**Non-zero**	**MSE**	**Non-zero**	**MSE**	**Non-zero**
			I.I.D. Scenario			
PCA	0.8721 (0.0992)	500.0 (0.0)	0.8647 (0.0993)	500.0 (0.0)	0.8749 (0.1004)	500.0 (0.0)
PLS	0.8240 (0.0771)	500.0 (0.0)	0.8260 (0.0793)	500.0 (0.0)	0.8278 (0.0790)	500.0 (0.0)
HSIC	0.8758 (0.1032)	499.2 (0.2)	0.8749 (0.0999)	499.2 (0.3)	0.8723 (0.1024)	499.2 (0.2)
Bair	0.8764 (0.0917)	220.9 (2.4)	0.8737 (0.0922)	220.9 (2.4)	0.8784 (0.0913)	220.9 (2.4)
CSPCA	0.8275 (0.0776)	498.8 (0.2)	0.8278 (0.0775)	498.8 (0.2)	0.8303 (0.0770)	498.9 (0.1)
SPLS	0.6576 (0.1007)	67.5 (15.3)	0.7645 (0.0926)	145.9 (20.2)	0.7990 (0.0688)	237.5 (19.6)
SPCA	0.8759 (0.0978)	34.2 (1.2)	0.8591 (0.0936)	34.2 (1.2)	0.8956 (0.0992)	34.2 (1.0)
SSPCA	0.7950 (0.1139)	108.2 (61.7)	0.8681 (0.1260)	131.5 (47.5)	0.7594 (0.1127)	136.1 (60.7)
SCS-PCA	0.6150 (0.0769)	59.8 (7.9)	0.6202 (0.0777)	40.1 (5.2)	0.6221 (0.0802)	29.9 (4.0)
			Correlated Scenario			
PCA	0.9769 (0.0987)	500.0 (0.0)	0.9774 (0.0976)	500.0 (0.0)	0.9732 (0.0942)	500.0 (0.0)
PLS	1.1240 (0.0958)	500.0 (0.0)	1.1226 (0.0965)	500.0 (0.0)	1.1249 (0.0953)	500.0 (0.0)
HSIC	0.9746 (0.0974)	500.0 (0.0)	0.9905 (0.0999)	500.0 (0.0)	1.0187 (0.0948)	500.0 (0.0)
Bair	1.0047 (0.0911)	220.3 (4.0)	1.0136 (0.0912)	220.3 (4.0)	1.0184 (0.0882)	220.3 (4.0)
CSPCA	1.0018 (0.0913)	500.0 (0.0)	1.0083 (0.0924)	500.0 (0.0)	1.0031 (0.0937)	500.0 (0.0)
SPLS	0.9295 (0.1166)	139.0 (31.9)	0.9734 (0.0932)	222.4 (32.0)	1.0372 (0.0916)	265.2 (31.0)
SPCA	1.0015 (0.1002)	26.7 (0.8)	0.9782 (0.0962)	26.6 (0.7)	1.0231 (0.0953)	26.9 (0.4)
SSPCA	1.0151 (0.1079)	76.6 (23.9)	1.0193 (0.1071)	111.6 (24.8)	1.0167 (0.1021)	81.9 (26.7)
SCS-PCA	0.9202 (0.0905)	83.0 (9.0)	0.9169 (0.0897)	53.8 (5.9)	0.9343 (0.0900)	40.4 (4.5)

Table 1 presents the results for the first simulation for the i.i.d. (top) and correlated (bottom) scenarios. SPLS provides the best performance in terms of prediction error for $q = 2$ and 3 components with SCS-PCA outperforming SPLS for $q = 4$ components. The two methods significantly outperform all existing methods across all components. PCA and SPCA are the worst performing methods with the highest prediction error across all components. SCS-PCA also provides the strongest shrinkage effect across methods with a supervised objective, only surpassed by SPCA which has an unsupervised objective. SSPCA follows closely with a strong shrinking effect across all components.

For the correlated scenario, SCS-PCA outperforms all methods in terms of MSE — including SPLS — adjusting well to high correlated features and is followed by SPCA, PCA and SPLS for $q = 2$ components. SPLS struggles to adjust to the correlation as the number of components increases, offering the worst MSE performance for $q = 3$ and 4 components, along with PLS across all components. SCS-PCA, followed by SSPCA, are effective in performing variable selection with more than 90% of the original features eliminated across all components, while SPLS also illustrates a strong shrinking effect but it is limited for $q = 2$ components only. SPCA had once again the strongest shrinking effect but for an unsupervised objective.

Numerical results for the i.i.d. and correlated scenarios of Simulation 2 are presented in Table 2. For the i.i.d. scenario SCS-PCA consistently outperforms all methods in terms of MSE across all components, followed by SPLS and SSPCA. The same conclusion can be extended for the sparsity effect, where SCS-PCA offers greater shrinkage than the two methods, only outperformed by SPCA. For the correlated scenario, SCS-PCA continues to outperform all existing method in prediction performance, followed by SPLS, PCA and SPCA. The three methods are followed by SPCA using HSIC and SSPCA while PLS performs significantly worse. In terms of shrinkage, SCS-PCA offers a strong effect, outperforming supervised methods, except for $q = 2$ components where SSPCA outperforms SCS-PCA.

Numerical results from the third simulation are presented in Table 3 for the i.i.d. scenario (top) and the correlated scenario (bottom). In general, conclusions align with the previous two simulations, with SCS-PCA offering the best MSE performance across all components. For both scenarios, PCA and SPCA are the next best performing methods, adapting to the complex non-linear relationship between the features and the data. In terms of sparsity, SCS-PCA offers the strongest shrinkage effect for all supervised methods for the majority of components, with SSPCA following. SPCA still has a stronger shrinkage effect due to its unsupervised objective.

Table 3: Mean squared error (MSE) and number of non-zero variables (Non-zero) with standard errors for $q = 2, 3$, and 4 components for Simulation 3.

Method	**2 Components**		**3 Components**		**4 Components**	
	MSE	**Non-zero**	**MSE**	**Non-zero**	**MSE**	**Non-zero**
			I.I.D. Scenario			
PCA	1.0419 (0.2488)	500.0 (0.0)	1.0368 (0.2488)	500.0 (0.0)	1.0343 (0.2495)	500.0 (0.0)
PLS	1.0698 (0.2382)	500.0 (0.0)	1.0728 (0.2390)	500.0 (0.0)	1.0737 (0.2391)	500.0 (0.0)
HSIC	1.0562 (0.2480)	500.0 (0.0)	1.0504 (0.2487)	500.0 (0.0)	1.0393 (0.2473)	500.0 (0.0)
Bair	1.0362 (0.2387)	228.5 (2.8)	1.0374 (0.2408)	228.5 (2.8)	1.0344 (0.2392)	228.5 (2.8)
CSPCA	1.0525 (0.2396)	500.0 (0.0)	1.0526 (0.2395)	500.0 (0.0)	1.0529 (0.2396)	500.0 (0.0)
SPLS	1.0432 (0.2230)	180.8 (32.4)	1.0574 (0.2383)	265.3 (31.9)	1.0701 (0.2257)	283.9 (29.2)
SPCA	1.0454 (0.2423)	35.0 (0.7)	1.0469 (0.2443)	34.3 (0.9)	1.0895 (0.2498)	34.3 (0.6)
SSPCA	1.0708 (0.2705)	34.5 (11.4)	1.0202 (0.2442)	102.8 (25.4)	1.0537 (0.2310)	115.5 (27.5)
SCS-PCA	**0.8874** (**0.1976**)	**46.1** (**11.2**)	**0.9684** (**0.2250**)	**53.6** (**7.3**)	**0.9769** (**0.2245**)	**43.5** (**5.0**)
			Correlated Scenario			
PCA	1.3304 (0.5388)	500.0 (0.0)	1.3146 (0.5298)	500.0 (0.0)	1.3007 (0.5297)	500.0 (0.0)
PLS	1.4397 (0.4344)	500.0 (0.0)	1.4909 (0.4555)	500.0 (0.0)	1.5072 (0.4636)	500.0 (0.0)
HSIC	1.3190 (0.5267)	500.0 (0.0)	1.3579 (0.5210)	500.0 (0.0)	1.3790 (0.5240)	500.0 (0.0)
Bair	1.3558 (0.4757)	243.7 (5.1)	1.3546 (0.4767)	243.7 (5.1)	1.3457 (0.4796)	243.7 (5.1)
CSPCA	1.3590 (0.4348)	500.0 (0.0)	1.3529 (0.4331)	500.0 (0.0)	1.3495 (0.4335)	500.0 (0.0)
SPLS	1.4777 (0.4279)	220.6 (36.5)	1.5967 (0.4255)	317.4 (56.8)	1.5155 (0.4515)	348.6 (48.9)
SPCA	1.3140 (0.5506)	26.5 (1.1)	1.3517 (0.5474)	26.6 (1.2)	1.3744 (0.5460)	29.2 (1.2)
SSPCA	1.3954 (0.5730)	88.4 (42.4)	1.3590 (0.5252)	73.0 (35.9)	1.4519 (0.5150)	93.7 (37.2)
SCS-PCA	**1.2905** (**0.4053**)	**78.6** (**14.2**)	**1.2745** (**0.4062**)	**58.3** (**9.4**)	**1.2673** (**0.4096**)	**44.0** (**7.1**)

Table 4: Numerical Results from the Mice Toxicity Dataset for $q = 2$ components.

Method	**MSE (s.e.)**	**Non-zero variables (s.e.)**
PCA	1.0681 (0.1701)	3116.0 (0.0)
PLS	0.6194 (0.0725)	3116.0 (0.0)
HSIC	0.9878 (0.1502)	3115.1 (0.2)
Bair	0.9545 (0.1650)	2123.9 (34.3)
CSPCA	0.7249 (0.1164)	3115.1 (0.2)
SPCA	1.1087 (0.1766)	11.4 (0.9)
SPLS	0.6978 (0.1042)	982.4 (355.1)
SSPCA	0.9285 (0.1367)	417.6 (152.8)
SCS-PCA	0.6009 (0.0835)	246.1 (34.8)

Table 5: Numerical Results with respective standard errors for Alon's Colon Cancer Dataset for $q = 2$ components.

Method	**Precision**	**Accuracy**	**AUC**	**Non-zero vars**
PCA	0.6352 (0.0443)	0.6000 (0.0441)	0.5 (0.0)	2000.0 (0.0)
LDA	0.8308 (0.0480)	0.7846 (0.0251)	0.9 (0.0)	2000.0 (0.0)
HSIC	0.8380 (0.0508)	0.7846 (0.0377)	0.9 (0.0)	2000.0 (0.0)
Bair	0.6963 (0.0489)	0.6538 (0.0417)	0.7 (0.1)	1342.9 (50.5)
CSPCA	0.8380 (0.0508)	0.7846 (0.0377)	0.9 (0.0)	2000.0 (0.0)
SDSPCA	0.7973 (0.0965)	0.6692 (0.0364)	0.7 (0.1)	—
SPCA	0.6563 (0.0415)	0.6462 (0.0348)	0.6 (0.0)	13.5 (0.5)
SSPCA	0.8394 (0.0571)	0.8000 (0.0328)	0.8 (0.0)	225.3 (106.8)
SCS-PCA	0.8421 (0.0446)	0.8077 (0.0367)	0.9 (0.0)	275.2 (72.0)

A general pattern observed across all simulations is that sparse methods consistently outperformed the non-sparse supervised methods in prediction error performance, indicating the benefits of removing non-significant variables in prediction tasks. Our proposed method, SCS-PCA offers competitive performance in both prediction accuracy and variable selection, which can be attributed to the direct trade-off offered by the proposed objective function. We have considered a spectrum of different relationships between the response and the variables including linear and non-linear scenarios, as well as correlated data generating mechanisms, which is a more representative example of real-world cases, e.g. genes in linkage disequilibrium.

5.2 Applications on Real Datasets

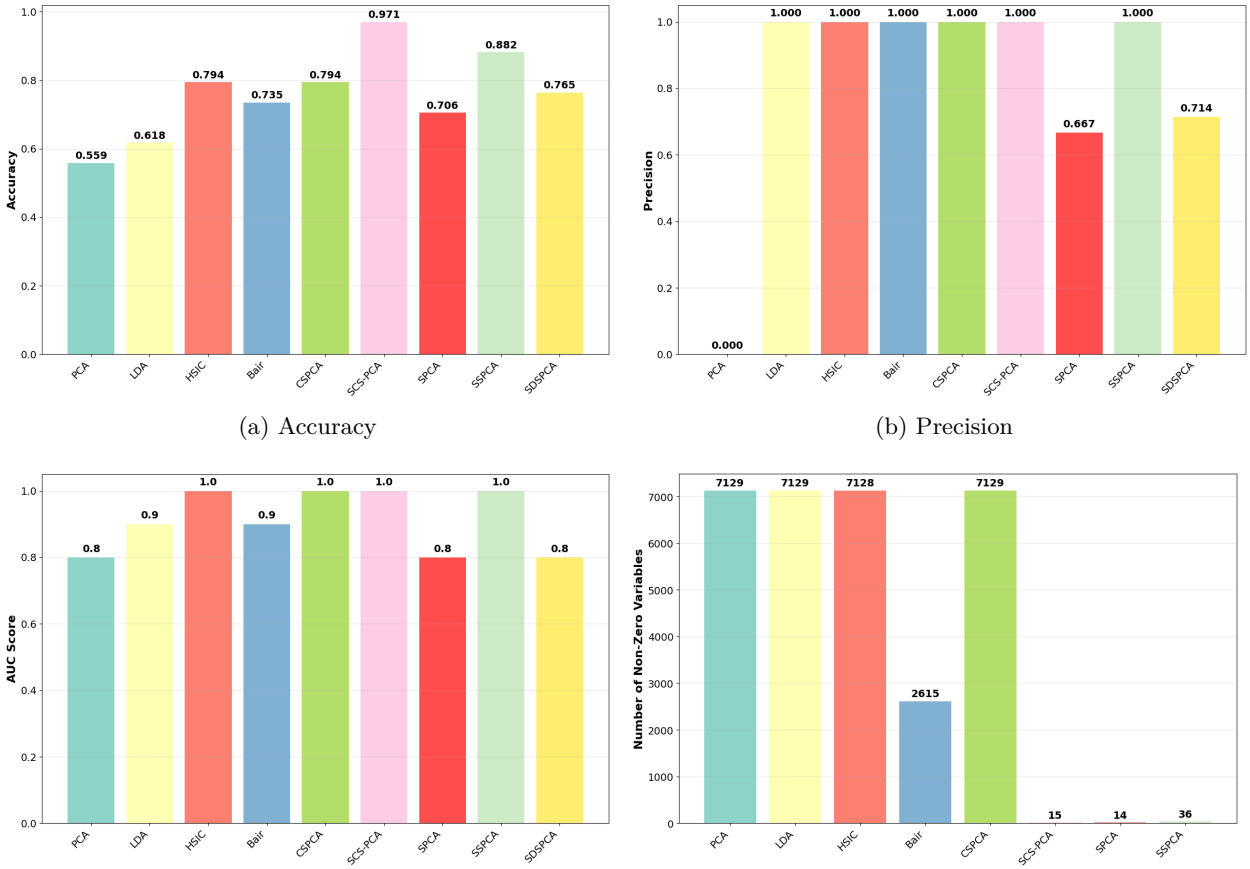
We use three real world, high-dimensional genetic datasets, one with a continuous phenotype and two with a binary phenotype, to examine SCS-PCA's performance against existing methods discussed in Section 2. For regression tasks, we employ the mice liver toxicity genetic dataset from Bushel et al. (2007), which is publicly available from R's Bioconductor package. The same methods used during the simulations in the previous section were also considered for this analysis. For classification, we employ the well-known leukemia dataset by Golub et al. (1999) — publicly available in Kaggle — which seeks to classify patients into acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL), along with a colon cancer dataset extracted from the colonCA library from the Bioconductor R package, seeking to classify individuals into two types of cancer tissue and was originally presented in Alon et al. (1999). For the former, methods were compared based on MSE and number of non-zero variables, while for the latter two, methods were compared according to accuracy, precision, area under the curve (AUC) and number of non-zero variables. For the two classification analyses, linear discriminant analysis (LDA) and SDSPCA were used instead of PLS and SPLS.

The mice dataset Bushel et al. (2007) consists of $p = 3116$ genes and $n = 64$ male rats, exposed to varying doses of acetaminophen (paracetamol). The response variable used is levels of Albumin (ALB.g.dl). A $60\% - 20\% - 20\%$ split into training, validation — for hyperparameter tuning — and test sets was considered 10 different times and results were averaged across all splits. For hyperparameter tuning, cross-validation was used with MSE as optimization metric. We consider the top $q = 2$ components for all projections. Table 4 illustrates the Monte Carlo estimates derived from the 10 splits. SCS-PCA achieves simultaneously the lowest MSE (0.6009) and strongest shrinking effect among all methods with a supervised objective — SPCA imposes sparsity on an unsupervised objective. PLS (0.6194) and SPLS (0.6978) follow in terms of prediction performance, with CSPCA also performing well (0.7249). SSPCA is the second most effective method in variable selection across supervised methods, however the prediction performance is not satisfactory (0.9285). PCA and SPCA are the worst performing methods in terms of prediction error, however, the latter has the strongest shrinking effect among all methods.

The leukemia dataset Golub et al. (1999) comprises $p = 7129$ genes and $n = 72$ individuals. The dataset is already divided into training ($n_{tr} = 38$) and test ($n_{ts} = 34$) sets, seeking to classify patients into two types of leukemia, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). No splitting is required for this dataset, however, 20% of the training data were used as validation set for tuning the hyperparameters where required. Results for accuracy, precision, AUC score and number of non-zero variables for $q = 2$ components are presented in barplots in Figure 1. SCS-PCA achieves the highest accuracy score (0.9706), while also providing the strongest shrinkage effect across supervised methods (15 non-zero variables). It achieves, along with SPCA using HSIC, CSPCA and SSPCA a perfect 1.0 score in both precision and AUC. SSPCA follows in both accuracy and sparsity performance (0.8824 and 36 non-zero variables). In general, all supervised methods, as well as PCA, were outperformed by their sparse variants in terms of accuracy. SPCA had 14 non-zero variables that contributed most in variance explained by the first two components, while for SDSPCA, recovery of non-zero variables is inapplicable since sparsity is imposed directly on the principal components and not the loadings. Overall, for the original split into training and test, used by Golub et al. (1999), SCS-PCA provides the strongest performance across all metrics.

The final dataset we employ in our analysis, was introduced by Alon et al. (1999) and it is publicly available on the R Bioconductor. It is an expression set consisting of $p = 2000$ genes and $n = 62$ samples, 40 of which are from tumors and 22 are from normal biopsies from healthy parts of patients' colons. A $60\% - 20\% - 20\%$ split into training, validation — for hyperparameter tuning — and test sets was considered 10 different times and results were averaged across all splits. For hyperparameter tuning, cross-validation was used with the

Figure 1: Results for Golub's Leukaemia Dataset for $q = 2$ components.



(a) Accuracy

(b) Precision

(c) Area Under Curve (AUC)

(d) Non-Zero Variables

logistic loss as optimization metric. We used $q = 2$ components for all projections. Numerical results for accuracy, precision, AUC and number of non-zero variables — along with the corresponding standard errors — are presented in Table 5. SCS-PCA was the best performing method in terms of accuracy (0.8421 ± 0.0446) and precision (0.8077 ± 0.0367), with SSPCA closely behind (0.8394 ± 0.0571 and 0.8000 ± 0.0328 respectively), with the former also having a higher AUC score over the latter (0.9 and 0.8 respectively). Strong performance in both precision and accuracy was also provided by their non-sparse counterparts, CSPCA and supervised PCA using HSIC, while PCA and SPCA were the poorest performing methods. SDSPCA and LDA also provided competitive results but still inferior to the top methods. In terms of shrinkage, SSPCA had a stronger effect than SCS-PCA (225 against 275 non-zero variables), although the standard error was higher (107 against 72 respectively). Overall, SCS-PCA provided strong predictive performance while also balancing well the sparsity effect, aiding in interpretability.

6 Discussion

In this paper, we have proposed a novel sparse supervised principal component analysis method, called sparse covariance supervised PCA (SCS-PCA), that incorporates sparsity into supervised PCA in the form of an L_1 penalty on the loadings of the projection matrix and performs simultaneous supervised dimensionality reduction and variable selection. We employ manifold proximal gradient descent to solve the proposed non-convex, non-smooth optimization problem, while also providing the theoretical guarantees for global convergence to a stationary point under minimum requirements.

Our proposal seeks for a trade-off between response relevant and interpretable projections within a single objective function, while solving the non-convex optimization problem directly. It is applicable to all types of response variables and offers guaranteed global convergence to a stationary point.

We have compared our method against existing sparse supervised PCA methods, supervised PCA methods, as well as state-of-the-art baselines such as sparse PCA and sparse PLS for both regression and classification tasks. SCS-PCA offers competitive performance, improving prediction accuracy, while also enhancing interpretability due to the strong shrinkage effect it offers. Simulations and applications on real-world genetic datasets illustrate SCS-PCA's practical strength.

Our work fills a significant gap in sparse supervised PCA's literature, namely the direct trade-off between prediction accuracy and sparsity in a single objective function and the robust optimization framework, along with the theoretical proofs of convergence. Some future directions include the extension of SCS-PCA to impose sparsity locally, or component-wise, since the current formulation applies sparsity globally on the loadings of the projection matrix. The framework presented here is deterministic, i.e. no probability distribution assumptions are made. It is warranted to consider a probabilistic formula for our method in order to apply Bayesian methods to quantify the uncertainty imposed by sparsity. This extension may also aid in reducing the tuning cost created from cross-validation through a prior distribution that will naturally perform sparsity.

Data Availability Statement

All data generating mechanisms used for the simulations, as well as the respective code used for the analysis are publicly available at the following GitHub repository `https://github.com/theopapazoglou/SCSPCA`. The real-world datasets used during the experimental section are also publicly available. The liver toxicity data can be accessed in `https://www.rdocumentation.org/packages/mixOmics/versions/6.3.2/topics/liver.toxicity`, Golub's leukemia dataset is available on Kaggle `https://www.kaggle.com/datasets/crawford/gene-expression` and the colon cancer data can be acquired in `https://bioconductor.org/packages/release/data/experiment/html/colonCA.html`.

Competing Interests

The authors have no competing interests to declare.

References

U Alon, N Barkai, D A Notterman, K Gish, S Ybarra, D Mack, and A J Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci U S A*, 96(12):6745–6750, Jun 1999.

Eric Bair and Robert Tibshirani. Semi-supervised methods to predict patient survival from gene expression data. *PLOS Biology*, 2(4), 04 2004.

Eric Bair, Trevor Hastie, Debashis Paul, and Robert Tibshirani. Prediction by supervised principal components. *Journal of the American Statistical Association*, 101(473):119–137, 03 2006.

Elnaz Barshan, Ali Ghodsi, Zohreh Azimifar, and Mansoor Zolghadri Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, 44(7):1357–1371, 2011.

Jun Bin, Fang-Fang Ai, Nian Liu, Zhi-Min Zhang, Yi-Zeng Liang, Ru-Xin Shu, and Kai Yang. Supervised principal components: a new method for multivariate spectral analysis. *Journal of Chemometrics*, 27(12): 457–465, 2013.

Pierre R. Bushel, Russell D. Wolfinger, and Greg Gibson. Simultaneous clustering of gene expression data with clinical chemistry and pathological evaluations reveals phenotypic prototypes. *BMC Systems Biology*, 1(1):15, 2007.

Guoqing Chao, Yuan Luo, and Weiping Ding. Recent advances in supervised dimension reduction: A survey. *Machine Learning and Knowledge Extraction*, 1(1):341–358, 2019.

Shixiang Chen, Shiqian Ma, Anthony Man-Cho So, and Tong Zhang. Proximal gradient method for nonsmooth optimization over the stiefel manifold. *SIAM Journal on Optimization*, 30(1):210–239, 2020.

Hyonho Chun and Sündüz Keleş. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *J R Stat Soc Series B Stat Methodol*, 72(1):3–25, Jan 2010.

Line Clemmensen, Trevor Hastie, Daniela Witten, and Bjarne Ersbøll. Sparse discriminant analysis. *Technometrics*, 53(4):406–413, 2011.

Patrick L. Combettes and Valérie R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

Eran Elhaik. Principal component analyses (pca)-based findings in population genetic studies are highly biased and must be reevaluated. *Scientific Reports*, 12(1):14683, 2022.

Chun-Mei Feng, Yong Xu, Jin-Xing Liu, Ying-Lian Gao, and Chun-Hou Zheng. Supervised discriminative sparse pca for com-characteristic gene selection and tumor classification on multiview biological data. *IEEE Transactions on Neural Networks and Learning Systems*, 30(10):2926–2937, 2019.

Benyamin Ghojogh and Mark Crowley. Unsupervised and supervised principal component analysis: Tutorial. *ArXiv*, 2019.

T R Golub, D K Slonim, P Tamayo, C Huard, M Gaasenbeek, J P Mesirov, H Coller, M L Loh, J R Downing, M A Caligiuri, C D Bloomfield, and E S Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, Oct 1999. ISSN 0036-8075 (Print); 0036-8075 (Linking).

Rosember Guerra-Urzola, Katrijn Van Deun, Juan C. Vera, and Klaas Sijtsma. A guide for sparse pca: Model comparison and applications. *Psychometrika*, 86(4):893–919, 2021.

Wen Huang and Ke Wei. Riemannian proximal gradient methods. *Mathematical Programming*, 194(1): 371–413, 2022.

Ian T Jolliffe. *Introduction*, pp. 1–9. Springer New York, New York, NY, 2002.

Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philos Trans A Math Phys Eng Sci*, 374(2065), Apr 2016.

John M. Lee. *Smooth Manifolds*, pp. 1–31. Springer New York, New York, NY, 2012.

Yongchun Li and Weijun Xie. Exact and approximation algorithms for sparse principal component analysis. *INFORMS Journal on Computing*, 37, 07 2024.

Masiala Mavungu. Computation of financial risk using principal component analysis. *Algorithmic Finance*, 10(1-2):1–20, 2023.

Amir Mehrabinezhad, Mohammad Teshnehlab, and Arash Sharifi. A comparative study to examine principal component analysis and kernel principal component analysis-based weighting layer for convolutional neural networks. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 12 (1), 2024.

Theodosios Papazoglou and Guosheng Yin. Covariance supervised principal component analysis. *arXiv*, 2025. URL `https://arxiv.org/abs/2506.19247`.

Neal Parikh, Stephen P. Boyd, and publisher. Now Publishers. *Proximal algorithms*. Foundations and trends in optimization, volume 1, issue 3, pages 127-239. Now Publishers, Hanover, Massachusetts, 2014.

Hector Pascual and Xin C. Yee. Least squares regression principal component analysis: A supervised dimensionality reduction method. *Numerical Linear Algebra with Applications*, 29(1), 2022.

Alexander Ritchie, Clayton Scott, Laura Balzano, Daniel Kessler, and Chandra S. Sripada. Supervised principal component analysis via manifold optimization. In *2019 IEEE Data Science Workshop (DSW)*, pp. 6–10, 2019.

Alexander Ritchie, Laura Balzano, and Clayton Scott. Supervised PCA: A multiobjective approach. *ArXiv*, 2020.

Sara Sharifzadeh, Ali Ghodsi, Line H. Clemmensen, and Bjarne K. Ersbøll. Sparse supervised principal component analysis (sspca) for dimension reduction and variable selection. *Engineering Applications of Artificial Intelligence*, 65:168–177, 2017.

Zhenhua Shi, Dongrui Wu, Jian Huang, Yu-Kai Wang, and Chin-Teng Lin. Supervised discriminative sparse pca with adaptive neighbors for dimensionality reduction. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2020.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 12 1996.

Daniela M Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, Jul 2009.

Shaojie Xu, Joel Vaughan, Jie Chen, A. Sudjianto, and Vijayan N. Nair. Supervised linear dimension-reduction methods: Review, extensions, and comparisons. *ArXiv*, 2021.

Ji-Gang Zhang and Hong-Wen Deng. Gene selection for classification of microarray data based on the bayes error. *BMC Bioinformatics*, 8(1):370, Oct 2007.

Hui Zou and Lingzhou Xue. A selective overview of sparse principal component analysis. *Proceedings of the IEEE*, 106(8):1311–1320, 2018.

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

A Appendix

Proof of Theorem 1

First, recall the definition of the proximal mapping associated to a convex function $f : \mathbb{R}^{n\times m} \to \mathbb{R} \cup \{+\infty\}$,

$$\text{prox}_f(Y) := \arg\min_X f(X) + \frac{1}{2}\|X - Y\|_F^2.$$

Let $h(X) = f(X) + \frac{1}{2}\|X - Y\|_F^2$. Since h is convex, as the sum of two convex functions, any minimizer is unique, which proves the uniqueness. There remains the proof of existence. Since f is convex, it can be lower bounded by an affine function, i.e. $f(X) \geq \langle X, A\rangle + b$, where $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}$ and $\langle X, A\rangle = \text{tr}(A^\top X)$ is the Frobenius inner product. Consequently,

$$h(X) \geq \langle X, A\rangle + b + \frac{1}{2}\|X - Y\|_F^2 \geq \min_{X\in\mathbb{R}^n}\{\langle X, A\rangle + b + \frac{1}{2}\|X - Y\|_F^2\} = c > -\infty,$$

making h bounded below. We can easily show that all sublevel sets of h are bounded since,

$$h(X) \leq \epsilon \implies \langle A, X\rangle + b + \frac{1}{2}\|X - Y\|_F^2 \leq \epsilon \Leftrightarrow \|X - (Y - A)\|_F^2 \leq \epsilon - b + \frac{1}{2}\|A\|_F^2 = C,$$

where $C > 0$ is a constant. Thus, the sublevel set $\{X : h(X) \leq \epsilon\}$ is contained in a Frobenius norm ball and is therefore bounded. Now, let (X_i) be a sequence such that $h(X_i) \downarrow \inf_X h(X)$. The sequence (X_i) lies in the sublevel set $\{X : h(X) \leq h(X_1)\}$, which is closed and bounded. By the Bolzano-Weirstrass theorem, there exists a convergent subsequence (X_{ik}) such that $X_{ik} \to X^*$ and by the lower semicontinuity of h,

$$h(X^*) \leq \lim \inf_{k \to \infty} h(X_{ik}) = \inf h.$$

Thus X^* is a minimizer of h, which is also unique, thus the proximal mapping is well-defined.

Proof of Theorem 2

Recall the formulation of the SCS-CPA optimization problem,

$$\min_{W \in \mathcal{S}(p,q)} F(W) := -\text{tr}(W^\top CW) + \eta \|W\|_1,$$

where $C = X^\top Y Y^\top X + \kappa X^\top X$ and $S(p,q)$ is the Stiefel manifold. The smooth part $f(W) = -\text{tr}(W^\top CW)$ is differentiable and its Euclidean gradient, $\nabla f(W) = -2CW$ is Lipschitz continuous with constant $L = 2\|C\|_{\text{op}}$, where $\|C\|_{\text{op}} < \infty$ is the operator norm corresponding to the largest singular value of the symmetric and positive semi-definite matrix C. The non-smooth part, $h(W) = \eta\|W\|_1$ is convex and Lipschitz continuous due to the L_1 norm. The Stiefel manifold is by definition compact and the retraction, Retr_W satisfies the following two properties:

1. $\|\text{Retr}_W - X\|_F \leq M_1 \|\xi\|_F, \ \ \forall X \in \mathcal{M}, \xi \in \mathcal{T}_X \mathcal{M}$
2. $\|\text{Retr}_W - (X + \xi)\|_F \leq M_2 \|\xi\|_F^2, \ \ \forall X \in \mathcal{M}, \xi \in \mathcal{T}_X \mathcal{M}.$

We will now employ Lemmas from Chen et al. (2020) to establish global convergence. By Lemma 5.1, the subproblem defined for SCS-CPA,

$$V_k := \arg \min_{V \in \mathcal{T}_{W_k}\mathcal{M}} \langle \text{grad} f(W_k), V \rangle + \frac{1}{2t}\|V\|_F^2 + \eta \|W_k + V\|_1,$$

ensures sufficient decrease in the objective

$$F(\text{Retr}_{W_k}(\alpha V_k)) \leq F(W_k) - \frac{\alpha}{2t}\|V_k\|_F^2.$$

If $V_k = 0$, then W_k satisfies the first-order optimality condition for (7)

$$0 \in \text{grad} f(W_k) + \text{Proj}_{\mathcal{T}_{W_k}\mathcal{S}(p,q)} \partial h(W_k),$$

where $\text{grad} f(W_k) = \text{Proj}_{\mathcal{T}_{W_k}\mathcal{S}(p,q)}(-2CW_k)$.

Since C is positive semi-definite, the sequence $\{F(W_k)\}$ is monotonically decreasing and bounded below. By Lemma 5.2, we obtain

$$\lim_{k \to \infty} \|V_k\|_F^2 = 0,$$

and by Lemma 5.3 every limit point W^* of $\{W_k\}$ is a stationary point. Finally, Theorem 5.5 by Chen et al. (2020) ensures that under the Assumptions discussed herein, every limit point of the sequence $\{W_k\}$ generated by Algorithm 1 is a stationary point of (7).