
Rethinking PCA Through Duality

Jan Quan*

ESAT-STADIUS & Leuven.AI
KU Leuven, Belgium
jan.quan@kuleuven.be

Johan Suykens

ESAT-STADIUS & Leuven.AI
KU Leuven, Belgium
johan.suykens@kuleuven.be

Panagiotis Patrinos

ESAT-STADIUS & Leuven.AI
KU Leuven, Belgium
panos.patrinis@kuleuven.be

Abstract

Motivated by the recently shown connection between self-attention and (kernel) principal component analysis (PCA), we revisit the fundamentals of PCA. Using the difference-of-convex (DC) framework, we present several novel formulations and provide new theoretical insights. In particular, we show the kernelizability and out-of-sample applicability for a PCA-like family of problems. Moreover, we uncover that simultaneous iteration, which is connected to the classical QR algorithm, is an instance of the difference-of-convex algorithm (DCA), offering an optimization perspective on this longstanding method. Further, we describe new algorithms for PCA and empirically compare them with state-of-the-art methods. Lastly, we introduce a kernelizable dual formulation for a robust variant of PCA that minimizes the l_1 -deviation of the reconstruction errors.

1 Introduction

Principal component analysis (PCA) is one of the most fundamental dimensionality reduction techniques and has seen widespread use in various fields, including machine learning [20, 10, 38, 23], signal processing [4, 24], and computer vision [52, 78, 82], due to its ability to efficiently extract meaningful features from high-dimensional data. Its nonlinear extension, kernel PCA [51], extends this method through a feature mapping into a high-dimensional space (possibly infinite-dimensional), enabling the discovery of nonlinear patterns. Crucially, this is achieved without ever computing the transformed data explicitly, using the kernel trick [11].

In recent years, (kernel) PCA has regained interest due to its connection to the self-attention mechanism [79], which lies at the heart of the widely successful transformer architecture. Transformers have revolutionized a wide range of applications such as natural language processing [13, 6, 42], computer vision [25, 59, 49, 76], and speech processing [41, 50]. This success has led to many attempts to understand the underlying mechanisms from a more principled viewpoint.

In particular, [69] establishes a direct link between self-attention and kernel PCA by showing that the attention outputs are projections of the query vectors onto the principal components axes of the key matrix in a feature space. In a similar vein, [18] recovers self-attention from an asymmetric kernel singular value decomposition [67], which can be seen as a generalization of kernel PCA to multiple data sources. Moreover, it has also been shown that models based on low-rank matrix decompositions can achieve results on par with modern transformers when trained correctly [32].

*Corresponding author.

Motivated by these modern connections to PCA, we aim to revisit the fundamentals of PCA. Our main tool is the concept of difference-of-convex (DC) duality, which provides a general framework for analyzing nonconvex problems, as it is for example known that any C^2 function admits a difference-of-convex formulation [34]. One parameter-free algorithm that is suited for solving these types of problems is the difference-of-convex algorithm (DCA) [68], of which the convex-concave procedure [81] is a special case. This algorithm, which fits into the majorization-minimization framework [66], has found many successful applications in machine learning including expectation-maximization [21], successive linear approximation [12], and various clustering methods [3, 56].

Contributions Our contribution is threefold.

1. We derive three novel DC dual pairs for PCA. For these formulations, we compare some simple gradient methods with existing solvers. Moreover, we show that if one of the functions in the primal is unitarily invariant, its DC dual is kernelizable and out-of-sample applicable.
2. We show that simultaneous iteration [77, Algorithm 28.3] is an instance of DCA applied to the variance maximization objective, which in turn is related to the famous QR algorithm for computing eigenvalues of dense matrices. This result gives a new connection between numerical linear algebra and optimization, akin to the connection between (linear) conjugate gradients [35] and accelerated gradient descent.
3. Based on a least absolute deviation formulation for the robust subspace recovery problem, we provide a novel DC formulation that is kernelizable. Moreover, we show that DCA is related to approaches based on iteratively reweighted least squares for this problem.

Related work The idea of using DC duality on the variance maximization formulation of PCA is not new. In particular, it has been studied for one component in [9, 36, 5] while more recent work extended it to multiple components [74]. Nevertheless, we propose three DC pairs that are based on other formulations of PCA and have, to the best of our knowledge, not been considered before.

The QR algorithm [31] is a classical algorithm for computing the eigenvalues of dense matrices. It is known as one of the top ten algorithms of the 20th century [19] and still the de facto industry standard in high-performance computing libraries and software such as MATLAB. The connection between the power method without normalization and DCA is described in [71, Prop. 4]. Notably, this connection is only made for the leading principal component. In contrast, we show that DCA applied to the variance maximization objective of PCA yields a fundamental link with simultaneous iteration [77, Algorithm 28.3], which in turn is related to the QR algorithm. Some more advanced algorithms for solving the same formulation can be found in [2, 64, 1, 27].

Beyond standard PCA, there has been significant interest in incorporating additional structure such as sparsity, or enhancing robustness to noise and outliers [15, 53, 72, 40, 28, 29, 71, 48]. In the kernel setting, [74] extends DC duality to robust PCA by considering other variance-like objectives, though this complicates interpretation and the out-of-sample extension was not considered.

2 Preliminaries

2.1 Notation

We denote by $\langle \cdot, \cdot \rangle$ the Euclidean inner product for vectors and the Frobenius inner product for matrices. Let $T \in \mathbb{R}^{m \times n}$. We denote by $T_{i,\cdot}$ the i th row of T . The Schatten p -norm of T with $p \in [1, +\infty)$ is defined through $\|T\|_{S_p}^p := \sum_{i=1}^{\min(m,n)} (\sigma_i(T))^p$ where $\sigma(T) \in \mathbb{R}^{\min(m,n)}$ denotes the vector of singular values of T in nonincreasing order. S_∞ is defined as the spectral norm, i.e., the largest singular value of its argument. The Schatten 1-norm is also known as the nuclear norm or trace norm. The Schatten 2-norm is also known as the Frobenius norm or the Hilbert–Schmidt norm. A (full) SVD of T is denoted as $U \text{Diag}(\sigma(T)) V^\top$, where $\text{Diag}(\sigma(T)) \in \mathbb{R}^{m \times n}$ is a rectangular diagonal matrix and U, V are real orthogonal. If T is square, $\lambda(T)$ denotes the eigenvalues of T in any order. We use $\overline{\text{Diag}}(\lambda(T))$ to denote a square diagonal matrix with $\lambda(T)$ on the diagonal. The (closed) unit ball of a norm $\|\cdot\|$ on \mathcal{X} is denoted by $B_{\|\cdot\|} := \{x \in \mathcal{X} \mid \|x\| \leq 1\}$. The extended reals are denoted by $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$. Let $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$. The convex subdifferential of f is denoted by

∂f , whereas its convex conjugate is denoted by f^* . The function f is called absolutely symmetric if $f(\gamma) = f(\hat{\gamma})$ for all γ and where $\hat{\gamma}$ denotes the vector with components $|\gamma_i|$ in decreasing order. A function $F : \mathbb{R}^{m \times n} \rightarrow \bar{\mathbb{R}}$ is unitarily invariant if $F(VXU) = F(X)$ for all $X \in \mathbb{R}^{m \times n}$ and both $U \in \mathbb{R}^{n \times n}$, $V \in \mathbb{R}^{m \times m}$ are real orthogonal.

Remark 2.1. For clarity of exposition, we perform our complete discussion in Euclidean spaces. The extension to infinite-dimensional Hilbert spaces for kernel methods only needs some additional technical details. More concretely, let $\phi : \mathbb{R}^d \rightarrow \mathcal{H}$ be some feature mapping and $\{x_i\}_{i=1}^N \subset \mathbb{R}^d$ the given data. Any time a formulation contains the data matrix $X = (x_1^\top \cdots x_N^\top)^\top$, it can be formally replaced by $\Gamma : \mathcal{H} \rightarrow \mathbb{R}^N$, where $(\Gamma w)_i = \langle \phi(x_i), w \rangle_{\mathcal{H}}$ for all $w \in \mathcal{H}$, and $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the inner product of the Hilbert space. The kernel matrix is then $K := \Gamma \Gamma^* = [\langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}]_{i,j=1}^N$. We also employ the following informal definition to denote formulations which allow for practical implementations.

Definition 2.2 ((Informal) kernelizability). *Let $X \in \mathbb{R}^{N \times d}$ be the data matrix. A problem formulation is said to be **kernelizable** if it can be written solely in terms of the kernel matrix $K = XX^\top \in \mathbb{R}^{N \times N}$. If a problem is kernelizable, we moreover call it **out-of-sample applicable** if for a new data point $\tilde{x} \in \mathbb{R}^d$, the ‘output’ of the problem can be computed from only K and $X\tilde{x}$.*

2.2 Difference-of-convex duality

Our primary tool for deriving new formulations is difference-of-convex (DC) duality, also known as Toland duality, which yields a pair of primal-dual problems between which strong duality holds.

Proposition 2.3 (Toland duality [73]). *Let $G : \mathbb{R}^{d \times s} \rightarrow \bar{\mathbb{R}}$, $F : \mathbb{R}^{N \times s} \rightarrow \bar{\mathbb{R}}$, be two convex, closed and proper functions, and $X : \mathbb{R}^{d \times s} \rightarrow \mathbb{R}^{N \times s}$ a linear mapping. Then, the following pair of primal-dual problems*

$$\underset{W \in \mathbb{R}^{d \times s}}{\text{minimize}} \quad G(W) - F(XW) \quad (\text{P})$$

$$\underset{H \in \mathbb{R}^{N \times s}}{\text{minimize}} \quad F^*(H) - G^*(X^\top H) \quad (\text{D})$$

*have the same infimum, i.e., **strong duality** holds. If W^* is a solution of (P), then any $H^* \in \partial F(XW^*)$ is a solution of (D). If H^* is a solution of (D), then any $W^* \in \partial G^*(X^\top H^*)$ is a solution of (P).*

Since this DC duality is a special case of the duality described in [62, 11H], the way the primal problem is perturbed/modified can lead to vastly different dual problems, as will become clear from comparing DC dual pairs (l)-(m) and (n)-(o) in Figure 2.

2.3 Difference-of-convex algorithm

An effective and widely used algorithm to solve DC problems of the form (P) is the difference-of-convex algorithm (DCA) [68]. The iterates of DCA are given by

$$H^{(k)} \in \partial F(XW^{(k)}); \quad W^{(k+1)} \in \partial G^*(X^\top H^{(k)}) = \arg \min_{\bar{W} \in \mathbb{R}^{d \times s}} G(\bar{W}) - \langle \bar{W}, X^\top H^{(k)} \rangle. \quad (\text{DCA})$$

To make sure the iterates are well-defined, we also require the following constraint qualification.

Assumption 2.4. $X \text{ dom}(\partial G) \subseteq \text{dom}(\partial F)$ and $X^\top \text{ range}(\partial F) \subseteq \text{range}(\partial G)$.

We have the following relation between the iterates of applying (DCA) to (P) and (D) respectively.

Proposition 2.5. *Let $\{W^{(k)}\}_{k=0}^\infty$ be the sequence of iterates obtained from applying (DCA) to (P) starting from some $W^{(0)} \in \mathbb{R}^{d \times s}$. Then, there exists a sequence of iterates $\{\tilde{H}^{(k)}\}_{k=0}^\infty$ obtained from applying (DCA) to (D) starting from $\tilde{H}^{(0)} \in \partial F(XW^{(0)})$ such that $W^{(k+1)} \in \partial G^*(X^\top \tilde{H}^{(k)})$.*

The convergence properties for (DCA) are well-established in the literature [68, 63, 55]. We summarize the basic results for completeness. Consider the problem (P) and suppose that the iterates $\{W^{(k)}\}_{k=0}^\infty$ are generated by (DCA).

- Since (DCA) fits into the majorization-minimization framework [66], it is a descent method (without linesearch), i.e., $G(W^{(k+1)}) - F(XW^{(k+1)}) \leq G(W^{(k)}) - F(XW^{(k)})$.

- If $G(W^{(k+1)}) - F(XW^{(k+1)}) = G(W^{(k)}) - F(XW^{(k)})$, then (DCA) terminates at the k th iteration and $W^{(k)}$ is a critical point of (P). Here, a critical point \bar{W} of (P) is defined as a point satisfying $\partial G(\bar{W}) \cap X^\top \partial F(X\bar{W}) \neq \emptyset$, which is a necessary condition for optimality. If $G(W) - F(XW)$ is bounded below, then every limit point of the sequence $\{W^{(k)}\}_{k=0}^\infty$ is a critical point. We note that the optimal value being bounded below is an assumption that is trivially satisfied for all our upcoming PCA formulations.
- In general, under mild assumptions, the rate for (DCA) is sublinear. However, if certain growth conditions hold, then linear rates are possible, see for example [55, 43].

3 Principal Component Analysis

In this section, we briefly review the classical formulations of PCA before proposing novel DC dual pairs. In particular, we show that a PCA-like family with one of the functions in the primal being unitarily invariant, is kernelizable and out-of-sample applicable in the dual. Moreover, we look into (DCA) applied to these new formulations and provide a connection to simultaneous iteration.

3.1 Classical PCA

There are many formulations of PCA that are commonly used. Some of the most important ones are shown in Figure 1. Following the exposition of [54], we start from the fundamental low-rank approximation of the data matrix X shown in (a). By the classical Eckart–Young–Mirsky theorem [26], the optimal solution is obtained by the sum of scaled outer products of left and right singular vectors, corresponding to the top k singular values. This solution is unique if the spectrum of the covariance matrix is simple.

By parameterizing A in (a) as its (full) SVD $U_A \Sigma_A V_A^\top$ where U_A and V_A are orthogonal matrices, one obtains the formulations (b) and (e) by taking $W = V_A$ and $H = U_A$ respectively. It should be noted that these problems do not admit unique solutions. In fact, there are infinitely many non-isolated minimizers since the problem is invariant to orthogonal transformations. This formulation is closely related to the Burer–Monteiro factorization for semidefinite programming [14, 47], though one key difference is that we impose orthogonality constraints on one of the factors.

To obtain formulation (c) from (b), it suffices to remark that (b) is a linear least-squares problem in the unconstrained factor B and the unique solution is $B = XW$. The resulting formulation minimizes the reconstruction error of PCA as a linear autoencoder [58]. The formulation (f) is obtained from (e) in a completely analogous manner. By expanding the Frobenius norm in (c), the formulation (d) is obtained. This is the classical variance maximization objective of PCA, where it is common, though not necessary, to assume that each column has zero mean. The formulation (g) is similarly derived from (f) and now contains the kernel matrix XX^\top (also known as the Gram matrix of the data). The relation between (d) and (g) lies at the heart of the success of many kernel methods.

Formulation (h) is derived by considering the low-rank approximation of the covariance matrix $X^\top X$, which is closely related to the formulations from the first column in Figure 1. To obtain (i), we impose the additional constraint that A is positive semi-definite in (h) since it does not change the solution. A SVD of A can now be written as $(U_A \Sigma_A^{1/2})(U_A \Sigma_A^{1/2})^\top$ which indeed yields the desired formulation. The last two formulations (j) and (k) follow immediately by expanding the square and using the definition of the Schatten 4-norm. It should be noted that four more formulations can be derived in an analogous manner by starting from the low-rank approximation of the kernel matrix XX^\top instead.

3.2 DC dual formulations

This subsection is devoted to describing novel DC dual formulations of PCA. Our main result is the following.

Theorem 3.1 (Fundamental PCA DC pairs). *Let $X \in \mathbb{R}^{N \times d}$ be the data matrix and $s \leq \text{rank}(X)$. Then, the three DC dual pairs in Figure 2 hold. Moreover, (l) and (n) share the same minimizers, as do (o) and (q).*

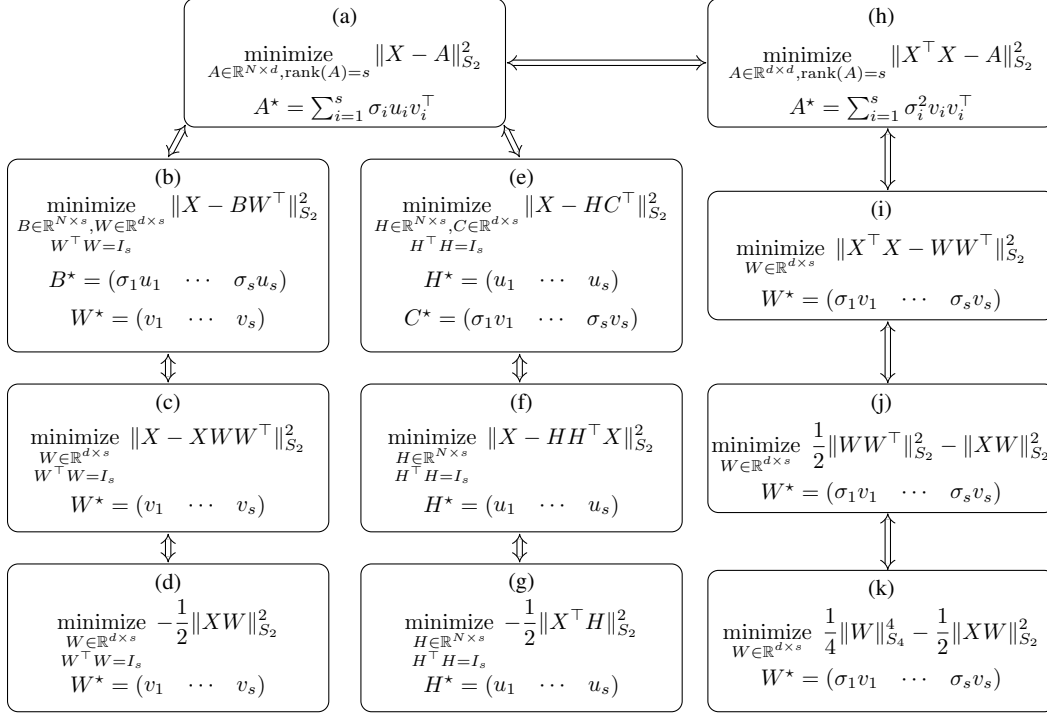


Figure 1: Classical PCA formulations for a data matrix $X \in \mathbb{R}^{N \times d}$ and $s \leq \text{rank}(X)$ principal components. The starred variables denote (not necessarily unique) global minimizers of the associated formulation. A (full) SVD of X is given by $U \text{Diag}(\sigma) V^T$. The arrows \Leftrightarrow denote that the two formulations are ‘equivalent’, in the sense that one can easily obtain the minimizers of one problem by solving the other.

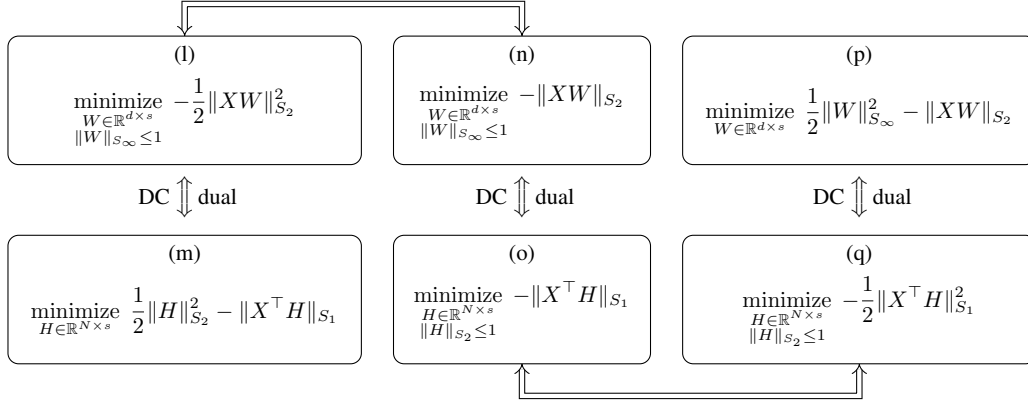


Figure 2: Three fundamental DC dual pairs for PCA.

To characterize the global minimizers of each problem from Figure 2, it suffices to combine the following known result with Proposition 2.3.

Proposition 3.2. *Let $X \in \mathbb{R}^{N \times d}$ be the data matrix and $s \leq \text{rank}(X)$. Then, the optimal value of (l) is $-\frac{1}{2} \sum_{i=1}^s \sigma_i(X)^2$. Consider any eigenvalue decomposition $X^T X = \tilde{W} \Lambda \tilde{W}^T$, where $\Lambda = \text{Diag}(\lambda_1, \lambda_2, \dots, \lambda_d)$ with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$ and \tilde{W} is a real orthogonal matrix. Then, the matrix $W \in \mathbb{R}^{d \times s}$ consisting of the first s columns of \tilde{W} achieves the optimum of (l).*

The DC dual pair (l)-(m) is well known for one component [9, 36, 5], and has recently been extended to multiple components in [74, Eq. (1) and (6)]. It should be noted that (l) is equivalent to (d) by

observing that the variance maximization formulation maximizes a (non-constant) convex function such that the constraint set can be relaxed to its convex hull, since the solutions necessarily lie on the boundary [61, Cor. 32.3.2]. Moreover, the constraint set is identified as the Stiefel manifold and its convex hull is the closed unit ball of the spectral norm [39, §3.4] which is exactly what appears in (l).

The two remaining DC dual pairs (n)-(o) and (p)-(q) are, to the best of our knowledge, novel. Their derivations rely on the simple fact that squaring/taking the square root and (positive) scaling does not change the solution sets of the minimization problems. Nevertheless, since the conjugates of the resulting transformed problems are quite different, so are their DC duals. In particular, the DC dual pair (n)-(o) is arguably more fundamental due to its symmetry. Each formulation captures a specific trade-off between Schatten p -norms.

Another interesting and novel dual problem follows from starting from (k) in Figure 1.

Proposition 3.3. *Let $X \in \mathbb{R}^{N \times d}$ be the data matrix and $s \leq \text{rank}(X)$. Then, the following DC dual pair holds.*

$$\underset{W \in \mathbb{R}^{d \times s}}{\text{minimize}} \frac{1}{4} \|W\|_{S_4}^4 - \frac{1}{2} \|XW\|_{S_2}^2 \quad \stackrel{\text{DC dual}}{\iff} \quad \underset{H \in \mathbb{R}^{N \times s}}{\text{minimize}} \frac{1}{2} \|H\|_{S_2}^2 - \frac{3}{4} \|X^\top H\|_{S_{4/3}}^{4/3}.$$

Remark 3.4. By replacing X with X^\top and vice versa (cf. (d) and (g)), four more analogous DC dual pairs can be formulated as the ones from Theorem 3.1 and Proposition 3.3.

In the Hilbert space setting, formulations involving the kernel matrix $K := XX^\top$ are needed, as they enable computations via the kernel trick. This is the case when G in the primal is unitarily invariant as we show in the following proposition.

Proposition 3.5 (Kernelizability for unitarily invariant G). *Let $X \in \mathbb{R}^{N \times d}$ be the data matrix and $s \leq \text{rank}(X)$. Consider the primal problem (P) and suppose $G = g \circ \sigma$ is unitarily invariant. Then, the corresponding DC dual (D) is kernelizable. In particular, (D) can be written as*

$$\underset{H \in \mathbb{R}^{N \times s}}{\text{minimize}} F^*(H) - g^*\left(\sqrt{\lambda(H^\top KH)}\right)$$

where $\sqrt{\cdot}$ is taken elementwise and $\lambda(\cdot)$ denotes the vector of eigenvalues of its argument (in any order).

Remark 3.6. The decomposition $G = g \circ \sigma$ always exists for unitarily invariant functions, where moreover g is absolutely symmetric, see Fact A.4. This fact also ensures that $g^*(\cdot)$ does not depend on the order of its arguments.

Remark 3.7. Note that while the objective function contains an eigenvalue decomposition, it is of an $s \times s$ matrix, which is small in practice, as typically only a few principal components are required. In particular, when $s = 1$, we have that $\lambda(H^\top KH) = H^\top KH \in \mathbb{R}$.

The previous result allows us to characterize a wide variety of formulations that are kernelizable in the DC dual, making the computations tractable. In particular, all the previously described formulations are kernelizable, since they are based on Schatten p -norms, which are unitarily invariant.

Another important requirement in practical applications is handling new, unseen data. The following result shows how one can perform feature extraction of a new sample by projecting onto the associated primal (lower-dimensional) subspace using only operations that are compatible with the kernel trick, thereby yielding out-of-sample applicability.

Theorem 3.8 (Out-of-sample applicability for unitarily invariant G). *Let $X \in \mathbb{R}^{N \times d}$ be the data matrix, $K = XX^\top \in \mathbb{R}^{N \times N}$ the kernel matrix, $s \leq \text{rank}(X)$ the number of principal components and $\tilde{x} \in \mathbb{R}^d$ a (new) data point. Consider the primal problem (P) and suppose $G = g \circ \sigma$ is unitarily invariant. Suppose moreover that $H^* \in \mathbb{R}^{N \times s}$ is a minimizer of the corresponding DC dual (D) and $H^{*\top}KH^*$ is nonsingular. Denote the eigenvalue decomposition of $H^{*\top}KH^* \in \mathbb{R}^{s \times s}$ as $V^\top \text{Diag}(\lambda)V$. Then, there exists a W^* which is a solution of (P) such that*

$$W^{*\top} \tilde{x} = V \text{Diag}(\mu \odot \lambda^{-1/2}) V^\top H^{*\top} X \tilde{x}$$

where $\mu \in \partial g^*(\sqrt{\lambda})$, \odot denotes the elementwise product, and both $\sqrt{\lambda}$ and $\lambda^{-1/2}$ are to be understood elementwise.

Remark 3.9. In a practical implementation, after finding an optimal solution H^* based on the training data, the matrix $V \text{Diag}(\mu \odot \lambda^{-1/2}) V^\top H^{*\top} \in \mathbb{R}^{s \times N}$ only needs to be calculated once.

3.3 DC algorithms

We now derive the algorithm (DCA) for multiple formulations from Subsection 3.2. To this end, a subgradient of a unitarily invariant function $G = g \circ \sigma$ is needed, which we can calculate from the following known result.

Proposition 3.10 ([46, Cor. 2.5]). *Let $X \in \mathbb{R}^{m \times n}$ and let $g : \mathbb{R}^{\min(m,n)} \rightarrow \overline{\mathbb{R}}$ be a proper and absolutely symmetric function. Suppose moreover that $\sigma(X) \subseteq \text{dom } g$. Then,*

$$\partial(g \circ \sigma)(X) = \{U \text{Diag}(\mu)V^\top \mid \mu \in \partial g(\sigma(X)), U \text{Diag}(\sigma(X))V^\top = \text{SVD}(X)\}.$$

By now applying (DCA) to the formulations (d) and (g) with the constraint set relaxed to its convex hull, one obtains Algorithms 1 and 2. Here $\overline{\text{SVD}}$ denotes a compact SVD, i.e., for $[U, \Sigma, V^\top] = \overline{\text{SVD}}(X)$ with $X \in \mathbb{R}^{m \times n}$, we have $U \in \mathbb{R}^{m \times r}$ and $V^\top \in \mathbb{R}^{r \times n}$ where r denotes the rank of X , and $U^\top U = I_r = V^\top V$. The derivation of these algorithms, as well as the reason why a compact SVD can be used here, is detailed in Appendix C.

Algorithm 1 (DCA) for (d)

Require: Matrix $X \in \mathbb{R}^{N \times d}$
1: Initialize $W^{(0)} \in \mathbb{R}^{d \times s}$ s.t. $\|W^{(0)}\|_{S_\infty} = 1$
2: Initialize $k \leftarrow 0$
3: **repeat**
4: $[U^{(k)}, \Sigma^{(k)}, V^{(k)\top}] \leftarrow \overline{\text{SVD}}(X^\top X W^{(k)})$
5: $W^{(k+1)} \leftarrow U^{(k)} V^{(k)\top}$
6: $k \leftarrow k + 1$
7: **until** convergence

Algorithm 2 (DCA) for (g)

Require: Matrix $K = X X^\top \in \mathbb{R}^{N \times N}$
1: Initialize $H^{(0)} \in \mathbb{R}^{N \times s}$ s.t. $\|H^{(0)}\|_{S_\infty} = 1$
2: Initialize $k \leftarrow 0$
3: **repeat**
4: $[U^{(k)}, \Sigma^{(k)}, V^{(k)\top}] \leftarrow \overline{\text{SVD}}(K H^{(k)})$
5: $H^{(k+1)} \leftarrow U^{(k)} V^{(k)\top}$
6: $k \leftarrow k + 1$
7: **until** convergence

Observe that when $s = 1$, the SVD is applied to a vector, resulting in its normalization, and the iterations correspond to the power method. This fact was also observed in [71, Prop. 4] after removing the sparsity constraint. In the next result, we show that these algorithms are connected to simultaneous iteration [77, Algorithm 28.3] which is strongly related to the QR algorithm [77, Theorem 28.3].

Theorem 3.11. *Algorithms 1 and 2 correspond to simultaneous iteration applied to $X^\top X$ and $X X^\top$ respectively, up to orthogonal transformation.*

Although the previous result states that (DCA) identifies the same subspace as the simultaneous iteration method, whose iterates span the leading eigenvector basis, (DCA) does not necessarily converge to the eigenvectors themselves, as the subspace admits infinitely many (orthonormal) bases. For the purpose of obtaining a lower-dimensional representation with meaningful features, the projection onto this subspace is often sufficient. If the principal vectors are specifically required, they can be recovered by performing an eigenvalue decomposition of a smaller $s \times s$ matrix.

An immediate corollary of this result and the known convergence result for simultaneous iteration [77, Theorem 28.4] is that Algorithms 1 and 2 inherit the linear rate as well as the convergence to global optimality with probability one, starting from random initialization. Similarly, the proximal gradient method that we use in Section 5 has a well-known interpretation of (DCA) applied to $(G + (1/2)\|\cdot\|^2) - (F + (1/2)\|\cdot\|^2)$, where G denotes the indicator of a convex set. Therefore, these can be connected to simultaneous iteration on an identity-shifted covariance/kernel matrix (i.e., it naturally incorporates regularization), such that again convergence to global optima is guaranteed.

Applying (DCA) to the problems (m) through (q) as well as their transposed variants (cf. Remark 3.4) is deferred to Appendix C. These algorithms are equivalent up to normalization and simple operations, as is to be expected due to the elementary problem transformation. A more interesting algorithm follows from considering (DCA) for the problems from Proposition 3.3. These are presented in Algorithms 3 and 4. The iteration is similar but now also uses information from the singular values through some nonlinear preconditioning. Here $\overline{\text{EIG}}$ denotes an eigenvalue decomposition of its argument but only keeping the columns corresponding to nonzero eigenvalues. Note that we are able to obtain a kernelizable formulation in the dual. The derivation is deferred to Appendix C.

Algorithm 3 (DCA) for Proposition 3.3 (primal)

Require: Matrix $X \in \mathbb{R}^{N \times d}$ 1: Initialize $W^{(0)} \in \mathbb{R}^{d \times s}$ 2: Initialize $k \leftarrow 0$ 3: **repeat**4: $[U^{(k)}, \Sigma^{(k)}, V^{(k)}] \leftarrow \overline{\text{SVD}}(X^\top X W^{(k)})$ 5: $W^{(k+1)} \leftarrow U^{(k)} (\Sigma^{(k)})^{1/3} V^{(k)\top}$ 6: $k \leftarrow k + 1$ 7: **until** convergence

Algorithm 4 (DCA) for Proposition 3.3 (dual)

Require: Matrix $K = X X^\top \in \mathbb{R}^{N \times N}$ 1: Initialize $H^{(0)} \in \mathbb{R}^{N \times s}$ 2: Initialize $k \leftarrow 0$ 3: **repeat**4: $[V^{(k)}, \Lambda^{(k)}] \leftarrow \overline{\text{EIG}}(H^{(k)\top} K H^{(k)})$ 5: $H^{(k+1)} \leftarrow K H^{(k)} V^{(k)} (\Lambda^{(k)})^{-1/3} V^{(k)\top}$ 6: $k \leftarrow k + 1$ 7: **until** convergence

4 Robust PCA

In this section, we start from formulation (c), the reconstruction error interpretation of PCA as a linear autoencoder, and note that it can be equivalently written as

$$\underset{\substack{W \in \mathbb{R}^{d \times s} \\ W^\top W = I_s}}{\text{minimize}} \sum_{i=1}^N \|X_{i,:} - W W^\top X_{i,:}\|^2$$

where $\|\cdot\|$ denotes the Euclidean norm. It is well known that measuring the error through the squared loss leads to non-robust models. A popular approach to obtain a robust version is to remove the square, thereby considering an l_1 penalty on the reconstruction errors instead. This is known in the robust subspace recovery literature as the least absolute deviation formulation [45]. We have the following DC dual pair, which was considered in [9] for only one component.

Proposition 4.1. *Let $X \in \mathbb{R}^{N \times d}$ be the data matrix and $s \leq \text{rank}(X)$. Then, the following DC dual pair holds:*

$$\underset{\substack{W \in \mathbb{R}^{d \times s} \\ \|W\|_{S_\infty} \leq 1}}{\text{minimize}} \sum_{i=1}^N \|X_{i,:} - W W^\top X_{i,:}\| \stackrel{\text{DC dual}}{\iff} \underset{H \in \mathbb{R}^{N \times s}}{\text{minimize}} \left(\sum_{i=1}^N \|X_{i,:}\| \sqrt{1 + \|H_{i,:}\|^2} \right) - \|X^\top H\|_{S_1}.$$

The dual formulation is kernelizable, as it can be written as

$$\underset{H \in \mathbb{R}^{N \times s}}{\text{minimize}} \left(\sum_{i=1}^N \sqrt{(X X^\top)_{i,i} (1 + \|H_{i,:}\|^2)} \right) - \text{tr} \left(\sqrt{H^\top X X^\top H} \right).$$

Moreover, since the associated G is the indicator function of the spectral unit norm ball, which is unitarily invariant, it satisfies the assumptions of Theorem 3.8 and therefore the problem is out-of-sample applicable. By applying (DCA) to both formulations from Proposition 4.1, we obtain Algorithms 5 and 6. In Appendix D, we show how the primal algorithm is related to an iteratively reweighted least squares scheme.

Algorithm 5 (DCA) for Proposition 4.1 (primal)

Require: Matrix $X \in \mathbb{R}^{N \times d}$ 1: Initialize $W^{(0)} \in \mathbb{R}^{d \times s}$ s.t. $\|W^{(0)}\|_{S_\infty} = 1$ 2: Initialize $k \leftarrow 0$ 3: **repeat**4: **for** $i = 1$ **to** N **do**5: $Y_{i,:}^{(k)} \leftarrow \frac{(X W^{(k)})_{i,:}}{\sqrt{\|X_{i,:}\|^2 - \|(X W^{(k)})_{i,:}\|^2}}$ 6: **end for**7: $[U^{(k)}, \Sigma^{(k)}, V^{(k)}] \leftarrow \overline{\text{SVD}}(X^\top Y^{(k)})$ 8: $W^{(k+1)} \leftarrow U^{(k)} V^{(k)\top}$ 9: $k \leftarrow k + 1$ 10: **until** convergence

Algorithm 6 (DCA) for Proposition 4.1 (dual)

Require: Matrix $K = X X^\top \in \mathbb{R}^{N \times N}$ 1: Initialize $H^{(0)} \in \mathbb{R}^{N \times s}$ 2: Initialize $k \leftarrow 0$ 3: **repeat**4: $[V^{(k)}, \Lambda^{(k)}] \leftarrow \overline{\text{EIG}}(H^{(k)\top} K H^{(k)})$ 5: $Y^{(k)} \leftarrow K H^{(k)} V^{(k)} (\Lambda^{(k)})^{-1/2} V^{(k)\top}$ 6: **for** $i = 1$ **to** N **do**7: $H_{i,:}^{(k)} \leftarrow \frac{(Y^{(k)})_{i,:}}{\sqrt{K_{i,i} - \|(Y^{(k)})_{i,:}\|^2}}$ 8: **end for**9: $k \leftarrow k + 1$ 10: **until** convergence

Table 1: Timing results for various methods applied to PCA formulations. The problem setting (N, d, s, ϵ) denotes a data matrix $X \in \mathbb{R}^{N \times d}$ with entries sampled from a standard normal distribution. s denotes the computed number of principal components and ϵ the stopping criterion tolerance. All timings are in milliseconds, and timings longer than 5 seconds are not displayed.

Method	(4000, 2000, 20, 10^{-3})	(2000, 4000, 20, 10^{-3})	(4500, 4500, 20, 10^{-3})
ZeroFPR (l)	2828.1 ± 376.5	1949.3 ± 97.9	8719.1
ZeroFPR (n)	210.4 ± 54.9	196.3 ± 49.1	441.0 ± 86.3
ZeroFPR (o)	392.6 ± 100.7	463.0 ± 121.5	1130.7 ± 143.3
ZeroFPR (q)	2341.7 ± 310.4	2021.0 ± 205.4	/
PG (l)	/	/	/
PG (n)	207.5 ± 14.0	139.1 ± 42.4	309.1 ± 26.1
PG (o)	203.6 ± 57.4	216.4 ± 29.1	689.7 ± 53.4
PG (q)	/	/	/
Algorithm 1	640.1 ± 154.5	2630.9 ± 33.0	3696.4 ± 115.2
Algorithm 2	2218.5 ± 35.2	635.3 ± 85.1	3294.2 ± 359.8
Algorithm 3	3848.8 ± 393.3	/	/
Algorithm 4	/	1868.2 ± 182.6	/
svds	808.0 ± 37.6	808.2 ± 29.6	2985.8 ± 128.3
KrylovKit	1130.4 ± 21.6	940.1 ± 13.6	3844.6 ± 253.2

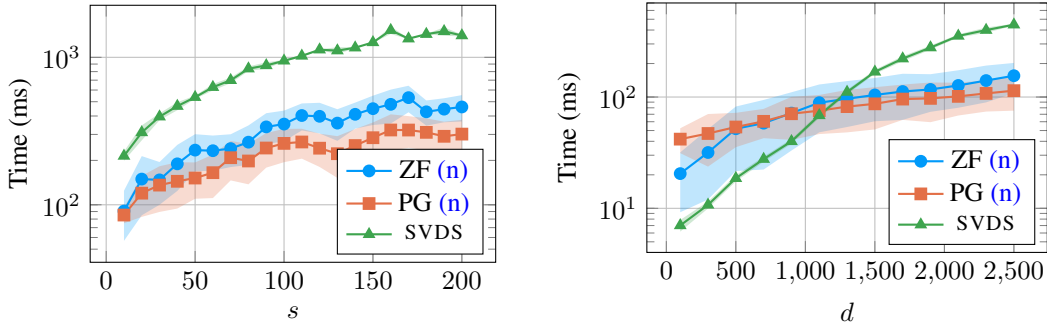


Figure 3: (left) Timings for the problem setting $(2000, 2000, s, 10^{-3})$ as defined in Table 1, with varying s and one standard deviation error bars. (right) Timings for the problem setting $(2000, d, 30, 10^{-3})$ as defined in Table 1, with varying d and one standard deviation error bars. In both figures, ZF is shorthand for ZeroFPR.

5 Experiments

We compare several simple gradient-based methods on various formulations from Theorem 3.1 and Proposition 3.3. All experiments are implemented in Julia 1.11.1 on a machine with an AMD Ryzen 7 Pro 5850U processor and 32 GB RAM. The timings are taken using BenchmarkTools.jl [17]. Our results are presented in Table 1. The code for reproducing all experiments is publicly available².

For the constrained problems, we apply both the proximal gradient (PG) method and ZeroFPR [70] using the implementation from [65]. Regarding the convergence properties of these algorithms, the standard assumptions to minimize objectives of the form $f(x) + g(x)$ using these methods require f to be L -smooth. In general, the (global) rates of PG and ZeroFPR to stationary points are both sublinear [70]. We note that, strictly speaking, for some formulations our f is nonsmooth. However, this does not pose a problem since the objectives are concave. Therefore, the Euclidean descent lemma [8, Lemma 5.7] holds with $L = 0$, and both PG and ZeroFPR enjoy the same theoretical guarantees as described above. In fact, we may choose arbitrarily large stepsizes, despite nonsmoothness. As a baseline, we compare our formulations against classical eigensolvers. Specifically, we used the classical ARPACK implementation of svds, which is based on the implicitly restarted Arnoldi

²<https://github.com/JanQ/pca-duality>

method [44]. Moreover, we also compare with KrylovKit.jl, a state-of-the-art package containing efficient and stable implementations of various Krylov subspace methods [33].

In the setting of this experiment, we conclude that the performance of PG and ZeroFPR varies depending on the formulation that is considered, thus motivating the theoretical framework presented in Section 3. In particular, state-of-the-art performance is achieved when using the newly proposed formulations (n)-(o). Moreover, we observe that depending on the shape of the data matrix, either the primal or the dual formulation is preferred. This is to be expected since the number of decision variables varies accordingly. We also remark that for classical variance-based formulations, these first-order methods do not perform well.

We also investigate the effect of the number of principal components and the sizes of the matrices involved for the best-performing methods in Figure 3. It can be seen that the first-order methods consistently outperform the classical methods for varying number of principal components, provided that the dimensions of the involved matrices are sufficiently large.

Additional experiments showcasing the performance of these first-order methods on various formulations as well as toy experiments for our robust kernel PCA can be found in Appendix F.

6 Limitations

We showed that the dual is kernelizable if one of the functions in the primal is unitarily invariant. While this may appear restrictive, many interesting functions such as the magnitude of the determinant, Ky Fan norms and Schatten p -norms all fall under this category, thereby leading to a whole new family of kernelizable formulations, derived in a principled way. Another issue that many kernel-based methods have is scalability, since the size of the kernel matrix scales as N^2 . Nevertheless, many mitigations have been proposed to deal with this problem, such as Nyström approximations [80, 30] or random Fourier features [60], which are also applicable to our framework.

Another limitation is that (DCA) does not yield the eigenvectors but only an (orthonormal) basis for the eigenspace. However, this is not really an issue in practice since only a small number of principal components are typically required and the eigenbasis can be recovered in a postprocessing step.

7 Conclusion and future outlook

In this paper we revisited PCA under the light of difference-of-convex duality. First, we proposed several novel DC pairs that yield more insight into the inner workings of PCA. Moreover, we showed that when one of the terms in the primal is unitarily invariant, the corresponding dual is not only kernelizable but also supports out-of-sample extensions which is essential for modern machine learning applications. Further, we showed that applying DCA to the variance maximization is related to simultaneous iteration, revealing a novel connection between optimization and numerical linear algebra. In addition, we derived a new kernelizable DC dual for robust kernel principal component analysis and an associated algorithm, which in turn is connected to iteratively reweighted least squares. Lastly, our experimental results showed that for the correct formulation, simple first-order optimization methods can outperform state-of-the-art solvers, depending on the required accuracy.

Several interesting research directions for future work remain open. One promising avenue is the exploration of other unitarily invariant objectives in the primal by incorporating domain knowledge or structural priors of the problem at hand. An alternative research direction is to consider deep variants by stacking (kernel) PCA layers [75] where each layer can be described with different formulations. In addition, a further line of research could involve developing specialized algorithms for our new DC dual pairs, as currently most of the algorithms are derived for (l) in Figure 2.

Acknowledgments and Disclosure of Funding

This work was supported by the Research Foundation Flanders (FWO) PhD grant 11A8T26N and research projects G081222N, G033822N, G0A0920N; Research Council KUL grant C14/24/103, iBOF/23/064; Flemish Government AI Research Program.

The authors thank Alexander Bodard for assistance with the experiments.

References

- [1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.
- [2] Foivos Alimisis, Yousef Saad, and Bart Vandereycken. Gradient-type subspace iteration methods for the symmetric eigenvalue problem. *SIAM Journal on Matrix Analysis and Applications*, 45(4):2360–2386, 2024.
- [3] Le Thi Hoai An, M Tayeb Belghiti, and Pham Dinh Tao. A new efficient algorithm based on DC programming and DCA for clustering. *Journal of Global Optimization*, 37(4):593–608, 2007.
- [4] Paolo Antonelli, HE Revercomb, LA Sromovsky, WL Smith, RO Knuteson, DC Tobin, RK Garcia, HB Howell, H-L Huang, and FA Best. A principal component noise filter for high spectral resolution infrared measurements. *Journal of Geophysical Research: Atmospheres*, 109(D23), 2004.
- [5] Giles Auchmuty. Dual variational-principles for eigenvalue problems. In *Symposia in Pure Mathematics*, volume 45, pages 55–71, 1986.
- [6] Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. *arXiv preprint arXiv:1809.10853*, 2018.
- [7] Heinz H Bauschke and Patrick L Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017.
- [8] Amir Beck. *First-order methods in optimization*. SIAM, 2017.
- [9] Amir Beck and Marc Teboulle. Dual randomized coordinate descent method for solving a class of nonconvex problems. *SIAM Journal on Optimization*, 31(3):1877–1896, 2021.
- [10] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- [11] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Fifth Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.
- [12] Paul S Bradley and Olvi L Mangasarian. Feature selection via concave minimization and support vector machines. In *International Conference on Machine Learning*, volume 98, pages 82–90, 1998.
- [13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.
- [14] Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.
- [15] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.
- [16] Riley Carlson, John Bauer, and Christopher D Manning. A new pair of gloves. *arXiv preprint arXiv:2507.18103*, 2025.
- [17] Jiahao Chen and Jarrett Revels. Robust benchmarking in noisy environments. *arXiv preprint arXiv:1608.04295*, 2016.
- [18] Yingyi Chen, Qinghua Tao, Francesco Tonin, and Johan Suykens. Primal-attention: Self-attention through asymmetric kernel SVD in primal representation. *Advances in Neural Information Processing Systems*, 36:65088–65101, 2023.
- [19] Barry A Cipra. The best of the 20th century: Editors name top 10 algorithms. *SIAM news*, 33(4):1–2, 2000.

- [20] John P Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *The Journal of Machine Learning Research*, 16(1):2859–2900, 2015.
- [21] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–22, 1977.
- [22] Li Deng. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- [23] Konstantinos I Diamantaras and Sun Yuan Kung. *Principal component neural networks: theory and applications*. John Wiley & Sons, Inc., 1996.
- [24] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *International Conference on Machine Learning*, 2004.
- [25] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [26] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.
- [27] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.
- [28] N Benjamin Erichson, Peng Zheng, Krithika Manohar, Steven L Brunton, J Nathan Kutz, and Aleksandr Y Aravkin. Sparse principal component analysis via variable projection. *SIAM Journal on Applied Mathematics*, 80(2):977–1002, 2020.
- [29] Jicong Fan and Tommy WS Chow. Exactly robust kernel principal component analysis. *IEEE transactions on neural networks and learning systems*, 31(3):749–761, 2019.
- [30] Shai Fine and Katya Scheinberg. Efficient SVM training using low-rank kernel representations. *The Journal of Machine Learning Research*, 2:243–264, 2001.
- [31] John GF Francis. The QR transformation a unitary analogue to the LR transformation—part 1. *The Computer Journal*, 4(3):265–271, 1961.
- [32] Zhengyang Geng, Meng-Hao Guo, Hongxu Chen, Xia Li, Ke Wei, and Zhouchen Lin. Is attention better than matrix decomposition? *arXiv preprint arXiv:2109.04553*, 2021.
- [33] Jutho Haegeman. KrylovKit, March 2024.
- [34] Philip Hartman. On functions representable as a difference of convex functions. 1959.
- [35] Magnus R Hestenes, Eduard Stiefel, et al. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.
- [36] J-B Hiriart-Urruty. Generalized differentiability/duality and optimization for problems dealing with differences of convex functions. In *Symposium on Convexity and Duality in Optimization*, pages 37–70. Springer, 1985.
- [37] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.
- [38] Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.
- [39] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *The Journal of Machine Learning Research*, 11(2), 2010.
- [40] Cheolmin Kim and Diego Klabjan. A simple and fast algorithm for L1-norm kernel PCA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):1842–1855, 2019.

- [41] Sehoon Kim, Amir Gholami, Albert Shaw, Nicholas Lee, Karttikeya Mangalam, Jitendra Malik, Michael W Mahoney, and Kurt Keutzer. Squeezeformer: An efficient transformer for automatic speech recognition. *Advances in Neural Information Processing Systems*, 35:9361–9373, 2022.
- [42] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213, 2022.
- [43] Hoai An Le Thi, Van Ngai Huynh, and Tao Pham Dinh. Convergence analysis of difference-of-convex algorithm with subanalytic data. *Journal of Optimization Theory and Applications*, 179(1):103–126, 2018.
- [44] Richard B Lehoucq, Danny C Sorensen, and Chao Yang. *ARPACK users’ guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, 1998.
- [45] Gilad Lerman and Tyler Maunu. An overview of robust subspace recovery. *Proceedings of the IEEE*, 106(8):1380–1410, 2018.
- [46] Adrian S Lewis. The convex analysis of unitarily invariant matrix functions. *Journal of Convex Analysis*, 2(1):173–183, 1995.
- [47] Xingguo Li, Junwei Lu, Raman Arora, Jarvis Haupt, Han Liu, Zhaoran Wang, and Tuo Zhao. Symmetry, saddle points, and global optimization landscape of nonconvex matrix factorization. *IEEE Transactions on Information Theory*, 65(6):3489–3514, 2019.
- [48] Jie Lin, Ting-Zhu Huang, Xi-Le Zhao, Teng-Yu Ji, and Qibin Zhao. Tensor robust kernel PCA for multidimensional data. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2662–2674, 2024.
- [49] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *International Conference on Computer Vision*, pages 10012–10022. IEEE, 2021.
- [50] Pingchuan Ma, Stavros Petridis, and Maja Pantic. End-to-end audio-visual speech recognition with conformers. In *International Conference on Acoustics, Speech and Signal Processing*, pages 7613–7617. IEEE, 2021.
- [51] Sebastian Mika, Bernhard Schölkopf, Alex Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel PCA and de-noising in feature spaces. *Advances in Neural Information Processing Systems*, 11, 1998.
- [52] Nikhil Naikal, Allen Y Yang, and S Shankar Sastry. Informative feature selection for object recognition via sparse PCA. In *International Conference on Computer Vision*, pages 818–825. IEEE, 2011.
- [53] Minh Nguyen and Fernando Torre. Robust kernel principal component analysis. *Advances in Neural Information Processing Systems*, 21, 2008.
- [54] Feiping Nie, Jianjun Yuan, and Heng Huang. Optimal mean robust principal component analysis. In *International Conference on Machine Learning*, pages 1062–1070. PMLR, 2014.
- [55] Konstantinos Oikonomidis, Emanuel Laude, and Panagiotis Patrinos. Forward-backward splitting under the light of generalized convexity. *arXiv preprint arXiv:2503.18098*, 2025.
- [56] Wei Pan, Xiaotong Shen, and Binghui Liu. Cluster analysis: unsupervised learning via supervised learning with a non-convex penalty. *The Journal of Machine Learning Research*, 14(1):1865–1889, 2013.
- [57] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.
- [58] Elad Plaut. From principal subspaces to principal components with linear autoencoders. *arXiv preprint arXiv:1804.10253*, 2018.

- [59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PmLR, 2021.
- [60] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 2007.
- [61] R Tyrrell Rockafellar. *Convex analysis*, volume 28. Princeton university press, 1997.
- [62] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.
- [63] Teodor Rotaru, Panagiotis Patrinos, and François Glineur. Tight analysis of difference-of-convex algorithm (DCA) improves convergence rates for proximal gradient descent. *arXiv preprint arXiv:2503.04486*, 2025.
- [64] Ahmed Sameh and Zhanye Tong. The trace minimization method for the symmetric generalized eigenvalue problem. *Journal of Computational and Applied Mathematics*, 123(1-2):155–175, 2000.
- [65] Lorenzo Stella. ProximalAlgorithms.jl: Proximal algorithms for nonsmooth optimization in Julia.
- [66] Ying Sun, Prabhu Babu, and Daniel P Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65(3):794–816, 2016.
- [67] Johan Suykens. SVD revisited: A new variational principle, compatible feature maps and nonlinear extensions. *Applied and Computational Harmonic Analysis*, 40(3):600–609, 2016.
- [68] Pham Dinh Tao and LT Hoai An. Convex analysis approach to DC programming: theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355, 1997.
- [69] Rachel SY Teo and Tan Nguyen. Unveiling the hidden structure of self-attention via kernel principal component analysis. *Advances in Neural Information Processing Systems*, 37:101393–101427, 2024.
- [70] Andreas Themelis, Lorenzo Stella, and Panagiotis Patrinos. Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone linesearch algorithms. *SIAM Journal on Optimization*, 28(3):2274–2303, 2018.
- [71] Mamadou Thiao, Pham D Tao, and Le An. A DC programming approach for sparse eigenvalue problem. In *International Conference on Machine Learning*, pages 1063–1070, 2010.
- [72] Michael Tipping. Sparse kernel principal component analysis. *Advances in Neural Information Processing Systems*, 13, 2000.
- [73] John F Toland. A duality principle for non-convex optimisation and the calculus of variations. *Archive for Rational Mechanics and Analysis*, 71:41–61, 1979.
- [74] Francesco Tonin, Alex Lambert, Panagiotis Patrinos, and Johan Suykens. Extending kernel PCA through dualization: sparsity, robustness and fast algorithms. In *International Conference on Machine Learning*, pages 34379–34393. PMLR, 2023.
- [75] Francesco Tonin, Qinghua Tao, Panagiotis Patrinos, and Johan Suykens. Deep kernel principal component analysis for multi-level feature learning. *Neural Networks*, 170:578–595, 2024.
- [76] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.
- [77] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. SIAM, 1997.

- [78] Matthew A Turk, Alex Pentland, et al. Face recognition using eigenfaces. In *Conference on Computer Vision and Pattern Recognition*, volume 91, pages 586–591, 1991.
- [79] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.
- [80] Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems*, 13, 2000.
- [81] Alan L Yuille and Anand Rangarajan. The concave-convex procedure (CCCP). *Advances in Neural Information Processing Systems*, 14, 2001.
- [82] Jun Zhang, Yong Yan, and Martin Lades. Face recognition: eigenface, elastic matching, and neural nets. *Proceedings of the IEEE*, 85(9):1423–1435, 1997.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract contains the claims made in the paper and the contributions are highlighted in a separate section in the introduction.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The paper has a separate section discussing the limitations.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: The paper states all the made assumptions and the proofs are complete and correct.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: The paper discloses all the information to reproduce the main experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code for the experiments is provided as supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper discloses the complete experimental setting.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper provides the standard deviation of benchmark timings.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The paper provides sufficient information on the computer resources.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper conforms to the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The assets used in the paper are properly credited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Facts from convex analysis

We include a summary of relevant facts from convex analysis in this Appendix for completeness.

Fact A.1 (Equality case of Fenchel–Young inequality, [7, Theorem 16.29]). *Let \mathcal{H} be a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Let $f : \mathcal{H} \rightarrow \overline{\mathbb{R}}$ be a convex, closed and proper function. Let $x, u \in \mathcal{H}$, then the following are equivalent:*

$$u \in \partial f(x) \iff f(x) + f^*(u) = \langle x, u \rangle_{\mathcal{H}} \iff x \in \partial f^*(u).$$

Fact A.2 (Conjugate separable sum, [8, Theorem 4.12]). *Let $f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \dots \times \mathbb{R}^{n_N} \rightarrow \overline{\mathbb{R}}$ be given by $f(x_1, x_2, \dots, x_N) = \sum_{i=1}^N f_i(x_i)$ where each $f_i : \mathbb{R}^{n_i} \rightarrow \overline{\mathbb{R}}$ is proper. Then $f^*(y_1, y_2, \dots, y_N) = \sum_{i=1}^N f_i^*(y_i)$.*

Fact A.3 (Conjugates). *The conjugate pairs in Table 2 hold. Recall that the indicator function δ_C of a set C is given by*

$$\delta_C(x) = \begin{cases} 0 & x \in C, \\ +\infty & \text{otherwise.} \end{cases}$$

Table 2: Conjugate functions [8, Appendix B]. $\|\cdot\|_*$ denotes the dual norm of $\|\cdot\|$ and $\|\cdot\|_p$ denotes the l_p -norm.

f	$\text{dom}(f)$	f^*	Assumption	
$\frac{1}{2}\ x\ ^2$	\mathbb{R}^n	$\frac{1}{2}\ y\ _*^2$	-	(1)
$\delta_{B_{\ \cdot\ }}(x)$	$B_{\ \cdot\ }$	$\ y\ _*$	-	(2)
$\frac{1}{p}\ x\ _p^p$	\mathbb{R}^n	$\frac{1}{q}\ x\ _q^q$	$p > 1, \frac{1}{p} + \frac{1}{q} = 1$	(3)
$-\sqrt{\alpha^2 - \ x\ ^2}$	$B_{\alpha^{-1}\ \cdot\ }$	$\alpha\sqrt{\ y\ _*^2 + 1}$	$\alpha > 0$	(4)

Fact A.4 (Equivalence unitarily invariant and absolutely symmetric, [46, Prop. 2.2]). *A function $F : \mathbb{R}^{m \times n} \rightarrow \overline{\mathbb{R}}$ is unitarily invariant if and only if it can be written as $f \circ \sigma$ where $f : \mathbb{R}^q \rightarrow \overline{\mathbb{R}}$ is absolutely symmetric and $\sigma : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^q$ denotes the singular value map with $q = \min(m, n)$, i.e., $\sigma(X)$ is the vector with components $\sigma_1(X) \geq \sigma_2(X) \geq \dots \geq \sigma_q(X) \geq 0$, the singular values of X .*

Fact A.5 (Spectral conjugate, [46, Theorem 2.4]). *Let $g : \mathbb{R}^q \rightarrow \overline{\mathbb{R}}$ be absolutely symmetric. Then*

$$(g \circ \sigma)^* = g^* \circ \sigma$$

where σ is defined as in Fact A.4.

Corollary A.6 (Conjugates spectral functions). *The conjugate pairs in Table 3 hold.*

Table 3: Conjugates of spectral functions.

f	$\text{dom}(f)$	f^*	Assumption	
$\delta_{B_{\ \cdot\ _{S_\infty}}}(X)$	$B_{\ \cdot\ _{S_\infty}}$	$\ Y\ _{S_1}$	-	(5)
$\ X\ _{S_p}$	$\mathbb{R}^{m \times n}$	$\delta_{B_{\ \cdot\ _{S_q}}}(Y)$	$p > 1, \frac{1}{p} + \frac{1}{q} = 1$	(6)
$\frac{1}{2}\ X\ _{S_\infty}^2$	$\mathbb{R}^{m \times n}$	$\frac{1}{2}\ Y\ _{S_1}^2$	-	(7)
$\frac{1}{p}\ X\ _{S_p}^p$	$\mathbb{R}^{m \times n}$	$\frac{1}{q}\ Y\ _{S_q}^q$	$p > 1, \frac{1}{p} + \frac{1}{q} = 1$	(8)

Fact A.7 (Subdifferential composition with norm, [7, Example 16.73]). *Let $h : \mathbb{R} \rightarrow \mathbb{R}$ be convex and even, and let $f = h \circ \|\cdot\|$. Then, $\partial h(0) = [-\rho, \rho]$ for some $\rho \in \mathbb{R}_+$ and*

$$\partial f(x) = \begin{cases} \left\{ \frac{\alpha}{\|x\|} x \mid \alpha \in \partial h(\|x\|) \right\} & \text{if } x \neq 0, \\ B_{\rho^{-1}\|\cdot\|} & \text{if } x = 0. \end{cases}$$

B Missing proofs

B.1 Proof of Proposition 2.3

We recall the proof from [73, Sec. 3.1] for completeness.

Proof. Let $\mathcal{H}_1 = \mathbb{R}^{d \times s}$, $\mathcal{H}_2 = \mathbb{R}^{N \times s}$ and denote $\langle \cdot, \cdot \rangle_{\mathcal{H}_1}$, $\langle \cdot, \cdot \rangle_{\mathcal{H}_2}$ for the corresponding inner products. Since F is convex, closed and proper, we have that $F = F^{**}$ [7, Cor. 13.38]. Combining this fact with the definition of the convex conjugate yields

$$\begin{aligned} \inf_{W \in \mathcal{H}_1} G(W) - F(XW) &= \inf_{W \in \mathcal{H}_1} G(W) - F^{**}(XW) \\ &= \inf_{W \in \mathcal{H}_1} G(W) - \sup_{H \in \mathcal{H}_2} \{\langle H, XW \rangle_{\mathcal{H}_2} - F^*(H)\} \\ &= \inf_{W \in \mathcal{H}_1} \inf_{H \in \mathcal{H}_2} G(W) + F^*(H) - \langle H, XW \rangle_{\mathcal{H}_2} \\ &= \inf_{H \in \mathcal{H}_2} F^*(H) - \sup_{W \in \mathcal{H}_1} \{\langle X^\top H, W \rangle_{\mathcal{H}_1} - G(W)\} \\ &= \inf_{H \in \mathcal{H}_2} F^*(H) - G^*(X^\top H) \end{aligned}$$

which shows the strong duality of the two problems. Moreover, let W^* be a solution of (P), i.e.,

$$G(W^*) - F(XW^*) = \min_{W \in \mathcal{H}_1} G(W) - F(XW) = \inf_{H \in \mathcal{H}_2} F^*(H) - G^*(X^\top H).$$

Thus we have

$$G(W^*) - F(XW^*) \leq F^*(\tilde{H}) - G^*(X^\top \tilde{H}) \quad (9)$$

for all $\tilde{H} \in \mathcal{H}_2$. Let $H^* \in \partial F(XW^*)$, then by Fact A.1

$$F(XW^*) + F^*(H^*) = \langle XW^*, H^* \rangle_{\mathcal{H}_2} = \langle W^*, X^\top H^* \rangle_{\mathcal{H}_1}$$

which we substitute in (9) with $\tilde{H} = H^*$ to obtain

$$G(W^*) - \langle W^*, X^\top H^* \rangle_{\mathcal{H}_1} \leq -G^*(X^\top H^*) \iff G(W^*) + G^*(X^\top H^*) \leq \langle W^*, X^\top H^* \rangle_{\mathcal{H}_1}.$$

The Fenchel–Young inequality [7, Prop. 13.15] yields

$$G(W^*) + G^*(X^\top H^*) \geq \langle W^*, X^\top H^* \rangle_{\mathcal{H}_1}$$

such that we must have equality throughout and therefore

$$G(W^*) - F(XW^*) = F^*(H^*) - G^*(X^\top H^*)$$

which shows that H^* solves (D) since strong duality holds. The proof that any $W^* \in \partial G^*(X^\top H^*)$ is a solution of (P) whenever H^* is a solution of (D) is completely analogous. \square

B.2 Proof of Proposition 2.5

Proof. The iterates for (DCA) applied to (P) and (D) are respectively

$$\begin{aligned} H^{(k)} &\in \partial F(XW^{(k)}); & W^{(k+1)} &\in \partial G^*(X^\top H^{(k)}), \\ \tilde{W}^{(k)} &\in \partial G^*(X^\top \tilde{H}^{(k)}); & \tilde{H}^{(k+1)} &\in \partial F^{**}(X\tilde{W}^{(k)}) = \partial F(X\tilde{W}^{(k)}) \end{aligned}$$

where the latter equality follows from the fact that F is convex, closed and proper and therefore $F = F^{**}$ [7, Cor. 13.38]. We now proceed by induction on k . The base case follows immediately by our choice of $\tilde{H}^{(0)} \in \partial F(XW^{(0)})$. Suppose that the statement holds for $k \geq 0$, we show that it also holds for $k+1$. Indeed, since $W^{(k+1)} \in \partial G^*(X^\top \tilde{H}^{(k)})$ by the induction hypothesis, we may take $\tilde{W}^{(k)} = W^{(k+1)}$ and

$$\tilde{H}^{(k+1)} = H^{(k+1)} \in \partial F(XW^{(k+1)})$$

which implies

$$W^{(k+2)} \in \partial G^*(X^\top H^{(k+1)}) = \partial G^*(X^\top \tilde{H}^{(k+1)}),$$

as desired. \square

B.3 Proof of Theorem 3.1

Proof. The problem (l) can be written in the form (P) via $G = \delta_{B_{\|\cdot\|_{S_\infty}}}$ and $F = \frac{1}{2}\|\cdot\|_{S_2}^2$. The DC dual follows from Proposition 2.3, (5) and (8).

The problem (n) is equivalent to (l) since multiplying by two and taking the square root does not change the set of minimizers. The problem (n) can be written in the form (P) via $G = \delta_{B_{\|\cdot\|_{S_\infty}}}$ and $F = \|\cdot\|_{S_2}$. The DC dual follows from Proposition 2.3, (5) and (6).

The problem (o) is equivalent to (q) since squaring the norm and dividing by two does not change the set of minimizers. The problem (p) can be written in the form (P) via $G = \frac{1}{2}\|\cdot\|_{S_\infty}^2$ and $F = \|\cdot\|_{S_2}$. The DC dual follows from Proposition 2.3, (7) and (6). \square

B.4 Proof of Proposition 3.2

Proof. Since the solution of (l) lies at the boundary by the discussion after Proposition 3.2, we may consider the equivalent problem

$$\underset{\substack{W \in \mathbb{R}^{d \times s} \\ W^\top W = I}}{\text{minimize}} F(W) := -\frac{1}{2}\|XW\|_{S_2}^2 = -\frac{1}{2}\text{tr}(W^\top X^\top XW). \quad (10)$$

Writing an eigenvalue decomposition $X^\top X = \tilde{W}\overline{\text{Diag}}(\Lambda)\tilde{W}^\top$ with \tilde{W} real orthogonal, and assuming that the (necessarily nonnegative real) eigenvalues are in descending order, we find that $\overline{\text{Diag}}(\Lambda)$ contains the entries $\sigma_1(X)^2 \geq \dots \geq \sigma_d(X)^2 \geq 0$ on the diagonal by [37, Theorem 2.6.3(b)]. Letting W^* be the matrix consisting of the first s columns of \tilde{W} , and using the orthonormality of the columns of \tilde{W} , we obtain that

$$F(W^*) = -\frac{1}{2}\text{tr}(W^{*\top} \tilde{W}\overline{\text{Diag}}(\sigma_1(X)^2, \dots, \sigma_d(X)^2)\tilde{W}^\top W^*) = -\frac{1}{2}\sum_{i=1}^s \sigma_i(X)^2$$

so the minimal value of (10) is less than or equal to $F(W^*)$. On the other hand, from [37, Theorem 4.3.45], we find that $F(W) \geq -\frac{1}{2}\sum_{i=1}^s \sigma_i(X)^2$ such that W^* is a solution of (10). \square

B.5 Proof of Proposition 3.3

Proof. The problem (k) can be written in the form (P) via $G = \frac{1}{4}\|\cdot\|_{S_4}^4$ and $F = \frac{1}{2}\|\cdot\|_{S_2}^2$. The DC dual follows from Proposition 2.3 and (8). \square

B.6 Proof of Proposition 3.5

Proof. Since G is absolutely invariant, it can be written as $g \circ \sigma$ by Fact A.4. Then, by using Proposition 2.3 and Fact A.5, we find that (D) becomes

$$\underset{H \in \mathbb{R}^{N \times s}}{\text{minimize}} F^*(H) - (g^* \circ \sigma)(X^\top H).$$

But the singular value vector $\sigma(X^\top H)$ is equal to the elementwise square root of the nonincreasing eigenvalue vector $\lambda(H^\top KH)$, i.e., $\sigma(X^\top H) = \sqrt{\lambda(H^\top KH)}$. Therefore, the DC dual (D) takes the form

$$\underset{H \in \mathbb{R}^{N \times s}}{\text{minimize}} F^*(H) - g^*\left(\sqrt{\lambda(H^\top KH)}\right)$$

where $\lambda(X)$ denotes the vector with the eigenvalues of X (necessarily square) in any order and $\sqrt{\cdot}$ is taken elementwise. \square

B.7 Proof of Theorem 3.8

Proof. Let a full SVD of $X^\top H^*$ be given by $U \text{Diag}(\sigma(X^\top H^*))V^\top$ where $U \in \mathbb{R}^{d \times d}$, $V \in \mathbb{R}^{s \times s}$ are both orthogonal and $\sigma(X^\top H^*) \in \mathbb{R}^{\min(d,s)} = \mathbb{R}^s$, since we assumed $s \leq \text{rank}(X) \leq \min(N, d) \leq d$. Let \tilde{U} be U after dropping the last $d - s$ columns, we can then equivalently write

$$X^\top H^* = \tilde{U}\overline{\text{Diag}}(\sigma(X^\top H^*))V^\top \quad (11)$$

where $\overline{\text{Diag}}(\sigma(X^\top H^\star)) \in \mathbb{R}^{s \times s}$ is now a square diagonal matrix with $\sigma(X^\top H^\star)$ on the diagonal.

Since H^\star is a solution of (D) by assumption, we know that there exists a $W^\star \in \partial G^*(X^\top H^\star)$ which is a solution of (P) by Proposition 2.3 and Assumption 2.4. By Proposition 3.10, this solution has the form $\tilde{U} \overline{\text{Diag}}(\mu) V^\top$ where $\mu \in \partial g^*(\sigma(X^\top H^\star))$ where we again omitted the columns of U that are multiplied with zero.

Therefore, it is required to calculate

$$W^{\star\top} \tilde{x} = V \overline{\text{Diag}}(\mu) \tilde{U}^\top \tilde{x}. \quad (12)$$

This is however not yet kernelizable. To this end, we first note that (11) is equivalent to

$$\tilde{U} = X^\top H^\star V (\overline{\text{Diag}}(\sigma(X^\top H^\star)))^{-1}$$

where we used the assumption that $H^{\star\top} K H^\star$ is nonsingular so that the singular values of $X^\top H^\star$ are nonzero. Secondly, we have an eigendecomposition

$$H^{\star\top} K H^\star = V \overline{\text{Diag}}(\lambda) V^\top = V (\overline{\text{Diag}}(\sigma(X^\top H^\star)))^2 V^\top$$

such that $\overline{\text{Diag}}(\sigma(X^\top H^\star)) = \overline{\text{Diag}}(\lambda)^{1/2}$ where λ is a shorthand for $\lambda(H^{\star\top} K H^\star)$.

Using these two insights and (12), we find the desired result

$$W^{\star\top} \tilde{x} = V \overline{\text{Diag}}(\mu \odot \lambda^{-1/2}) V^\top H^{\star\top} X \tilde{x}$$

where $\mu \in \partial g^*(\sqrt{\lambda})$ and both $\sqrt{\lambda}$ and $\lambda^{-1/2}$ are elementwise. \square

B.8 Proof of Theorem 3.11

Proof. Without loss of generality, we consider the statement for $X^\top X$. (DCA) for (d) and simultaneous iteration are shown in Algorithms 7 and 8 respectively. Here, **QR** denotes a compact QR decomposition. The result then follows from noting that **QR** and **SVD** compute orthonormal bases for the column spaces of their arguments in U and Q respectively. Moreover, if $U^{(k)}$ and $Q^{(k)}$ span the same space and $V^{(k)} \in \mathbb{R}^{s \times r}$ has full rank, then we also have $\text{col}(X^\top X U^{(k)} V^{(k)\top}) = \text{col}(X^\top X Q^{(k)})$ where $\text{col}(A)$ denotes the column space of A . Therefore, $U^{(k)}$ and $Q^{(k)}$ form orthonormal bases for the same subspaces at every iteration and are related through orthogonal transformation by [37, Theorem 2.1.18]. \square

Algorithm 7 (DCA) for (d)

Require: Matrix $X \in \mathbb{R}^{N \times d}$

- 1: Initialize $W^{(0)} \in \mathbb{R}^{d \times s}$
 - 2: Initialize $k \leftarrow 0$
 - 3: **repeat**
 - 4: $[U^{(k)}, \Sigma^{(k)}, V^{(k)}] \leftarrow \overline{\text{SVD}}(X^\top X W^{(k)})$
 - 5: $W^{(k+1)} \leftarrow U^{(k)} V^{(k)\top}$
 - 6: $k \leftarrow k + 1$
 - 7: **until** convergence
-

Algorithm 8 Simultaneous iteration for $X^\top X$

Require: Matrix $X \in \mathbb{R}^{N \times d}$

- 1: Initialize $W^{(0)} \in \mathbb{R}^{d \times s}$
 - 2: Initialize $k \leftarrow 0$
 - 3: **repeat**
 - 4: $[Q^{(k)}, R^{(k)}] \leftarrow \overline{\text{QR}}(X^\top X W^{(k)})$
 - 5: $W^{(k+1)} \leftarrow Q^{(k)}$
 - 6: $k \leftarrow k + 1$
 - 7: **until** convergence
-

B.9 Proof of Proposition 4.1

Proof. First, the objective from (c) without the square can be written as

$$\underset{\substack{W \in \mathbb{R}^{d \times s} \\ W^\top W = I_s}}{\text{minimize}} \sum_{i=1}^N \sqrt{(X_{i,:} - W W^\top X_{i,:})^\top (X_{i,:} - W W^\top X_{i,:})}$$

which is equivalent to

$$\underset{\substack{W \in \mathbb{R}^{d \times s} \\ W^\top W = I_s}}{\text{minimize}} \sum_{i=1}^N \sqrt{\|X_{i,:}\|^2 - \|W^\top X_{i,:}\|^2}$$

where we expanded and used the constraint. This minimization problem can be written as

$$\begin{aligned} & \underset{\substack{W \in \mathbb{R}^{d \times s} \\ W^\top W = I_s}}{\text{minimize}} \quad - \sum_{i=1}^N f_i((XW)_{i,:}) \end{aligned}$$

where

$$f_i : \mathbb{R}^s \rightarrow \overline{\mathbb{R}} : w \mapsto \begin{cases} -\sqrt{\|X_{i,:}\|^2 - \|w\|^2} & \text{if } \|w\| \leq \|X_{i,:}\|, \\ +\infty & \text{otherwise,} \end{cases}$$

which is a convex function (see [61, p. 106]). The separable sum of convex functions is again convex and the objective is therefore to maximize a convex function. By the same reasoning as in Section 3.2, the constraint set can again be relaxed to its convex hull, being the spectral norm unit ball. The result now follows from taking $G = \delta_{B_{\|\cdot\|_{S_\infty}}}$ and $F = \sum_{i=1}^N f_i((\cdot)_{i,:})$, where the conjugates follow by (5), and combining (4) with Fact A.2. \square

C DC algorithms

This Appendix is devoted to deriving the many algorithms obtained from (DCA). To this end, we first recall some known subdifferential results.

Proposition C.1. *Let $F = \|\cdot\|_{S_1}$, then*

$$\tilde{U}\tilde{V}^\top \in \partial F(X) = \{U \text{Diag}(\mu)V^\top \mid \mu \in \partial\|\sigma(X)\|_1, U \text{Diag}(\sigma(X))V^\top = \mathbf{SVD}(X)\}$$

where $\|\cdot\|_1$ denotes the l_1 -norm and $\tilde{U}\tilde{\Sigma}\tilde{V}^\top$ is a compact SVD of X .

Proof. The expression of the subdifferential follows immediately from Proposition 3.10 and the definition of the Schatten 1-norm. To show that $\tilde{U}\tilde{V}^\top$ is a subgradient at X , from [61, Theorem 23.8] we have that

$$\partial\|\sigma(X)\|_1 = \partial|\sigma(X)_1| \times \partial|\sigma(X)_2| \times \cdots \times \partial|\sigma(X)_{\min(N,d)}|$$

where we assume that $X \in \mathbb{R}^{N \times d}$. Moreover, recall that

$$\partial|x| = \begin{cases} \{1\} & \text{if } x > 0, \\ [-1, 1] & \text{if } x = 0, \\ \{-1\} & \text{if } x < 0. \end{cases}$$

Therefore $\mu = \text{sign}(\sigma(X)) \in \partial\|\sigma(X)\|_1$ where $\text{sign}(x)$ is the elementwise sign function with the convention $\text{sign}(0) = 0$. For this choice of μ , we obtain

$$\tilde{U}\tilde{V}^\top = U \text{Diag}(\mu)V^\top \in \partial F(X)$$

since the singular values are always nonnegative and the columns/rows of U, V corresponding to zero singular values are multiplied with zero in the sum. \square

Remark C.2. It should be noted that when one of the singular values is zero, it is possible to choose any value in the interval $[-1, 1]$ for the corresponding entry in μ . While this is generally not encountered in practice, it could lead to some interesting algorithms to deal with degenerate cases.

Proposition C.3. *Let $F = \frac{1}{2}\|\cdot\|_{S_2}^2$, then $\partial F(X) = \{X\}$.*

Proof. The statement follows from the fact that F is differentiable and [8, Theorem 3.33]. \square

Proposition C.4. *Let $F = \|\cdot\|_{S_2}$, then $\frac{X}{\|X\|_{S_2}} \in \partial F(X)$ for $X \neq 0$.*

Proof. The result follows immediately from Proposition 3.10 and the fact that $\partial\|x\|_2 = \{\frac{x}{\|x\|_2}\}$ for $x \neq 0$ since it is differentiable in that case. \square

Remark C.5. The case $X = 0$ in the preceding Proposition can be handled by using the fact that $\partial\|0\|_2 = B_{\|\cdot\|_2}$.

Proposition C.6. Let $F = \frac{1}{2} \|\cdot\|_{S_1}^2$, then $\text{tr}(\tilde{\Sigma})\tilde{U}\tilde{V}^\top = \|X\|_{S_1}\tilde{U}\tilde{V}^\top \in \partial F(X)$, where $\tilde{U}\tilde{\Sigma}\tilde{V}^\top$ is a compact SVD of X .

Proof. Note that $F = g \circ H$ where $g(t) = \frac{1}{2} \max(0, t)^2$ is a nondecreasing, differentiable convex function and $H = \|\cdot\|_{S_1}$ is convex. By the chain rule of subdifferential calculus [8, Theorem 3.47], we have that $\partial F(X) = g'(H(X))\partial H(X)$. The desired subgradient then follows after using Proposition C.1. \square

Proposition C.7. Let $F = \frac{3}{4} \|\cdot\|_{S_{4/3}}^{4/3}$, then $\partial F(X) = \{\tilde{U}\tilde{\Sigma}^{1/3}\tilde{V}^\top\}$, where $\tilde{U}\tilde{\Sigma}\tilde{V}^\top$ is a compact SVD of X .

Proof. Note that $F = \frac{3}{4} \|\cdot\|_{4/3}^{4/3} \circ \sigma$ where $\|\cdot\|_{4/3}$ denotes the $l_{4/3}$ -norm

$$\|\cdot\|_{4/3} : \mathbb{R}^q \rightarrow \mathbb{R} : x \mapsto (|x_1|^{4/3} + |x_2|^{4/3} + \dots + |x_q|^{4/3})^{3/4}.$$

Moreover, we have that the gradient of $\frac{3}{4} \|\cdot\|_{4/3}^{4/3}$ is given by the elementwise cube root of its argument. The desired result then follows from Proposition 3.10 and the same arguments from the proof of Proposition C.1 to use a compact SVD instead of a full SVD. \square

C.1 Derivation of Algorithms 1 and 2

Proof. The formulation (I) (which is equivalent to (d) after relaxing the constraint), can be written as (P) with $G = \delta_{B\|\cdot\|_{S_\infty}}$ and $F = \frac{1}{2} \|\cdot\|_{S_2}^2$. Algorithm 1 then follows from (DCA) by noting that $\partial F(X) = \{X\}$ (Proposition C.3) and $\tilde{U}\tilde{V}^\top \in \partial G^*(Y)$ where $\tilde{U}\tilde{\Sigma}\tilde{V}^\top = \overline{\text{SVD}}(Y)$ by (5) and Proposition C.1. The derivation of Algorithm 2 follows analogously. \square

C.2 Derivation of Algorithm 3

Proof. The primal formulation of Proposition 3.3 can be written as (P) with $G = \frac{1}{4} \|\cdot\|_{S_4}^4$ and $F = \frac{1}{2} \|\cdot\|_{S_2}^2$. Algorithm 3 then follows from (DCA) by using $\partial F(X) = \{X\}$ (Proposition C.3) and $\partial G^*(Y) = \{\tilde{U}\tilde{\Sigma}^{1/3}\tilde{V}^\top\}$ where $\tilde{U}\tilde{\Sigma}\tilde{V}^\top = \overline{\text{SVD}}(Y)$ by (8) and Proposition C.7. \square

C.3 Derivation of Algorithm 4

Proof. The dual formulation of Proposition 3.3 can be written as (P) with $F = \frac{3}{4} \|\cdot\|_{S_{4/3}}^{4/3}$, $G = \frac{1}{2} \|\cdot\|_{S_2}^2$ where X^\top takes the role of X . Using (DCA), $\partial G^*(X) = \{X\}$ (Proposition C.3) and Proposition C.7 yields the updates

$$[U^{(k)}, \Sigma^{(k)}, V^{(k)\top}] \leftarrow \overline{\text{SVD}}(X^\top H^{(k)}), \quad H^{(k+1)} \leftarrow XU^{(k)}(\Sigma^{(k)})^{1/3}V^{(k)\top}.$$

Now note that

$$U^{(k)} = X^\top H^{(k)} V^{(k)} \Sigma^{(k)-1}$$

where $\Sigma^{(k)}$ is invertible since a compact SVD is taken. Moreover, a compact eigendecomposition of $H^{(k)\top}KH^{(k)}$ yields $V^{(k)}\Lambda^{(k)}V^{(k)\top} = H^{(k)\top}KH^{(k)}$ where $\Lambda^{(k)} = \Sigma^{(k)^2}$. The update is therefore equivalent to

$$[V^{(k)}, \Lambda^{(k)}] \leftarrow \overline{\text{EIG}}(H^{(k)\top}KH^{(k)}), \quad H^{(k+1)} \leftarrow KH^{(k)}V^{(k)}\Sigma^{(k)}\Lambda^{(k)-1/3}V^{(k)\top}$$

which is exactly Algorithm 4. \square

C.4 Derivation of other DC algorithms

In this subsection, we consider kernelizable dual updates of (DCA) applied to the problems (I) through (q). First, we have the following general result when G is unitarily invariant.

Proposition C.8. Let $X \in \mathbb{R}^{N \times d}$ be the data matrix, $s \leq \text{rank}(X)$ and define the kernel matrix $K = XX^\top$. Consider the primal problem (P) and suppose $G = g \circ \sigma$ is unitarily invariant. Assume $H^{(k)\top}KH^{(k)}$ is invertible for each k , then the iterations of (DCA) applied to the dual are

$$\begin{aligned} [V^{(k)}, \overline{\text{Diag}}(\lambda^{(k)})] &\leftarrow \overline{\text{EIG}}(H^{(k)\top}KH^{(k)}) \\ H^{(k+1)} &\in \partial F \left(KH^{(k)}V^{(k)}\overline{\text{Diag}}(\mu^{(k)} \oslash \sqrt{\lambda^{(k)}})V^{(k)\top} \right) \end{aligned}$$

where $\mu^{(k)} \in \partial g^*(\sqrt{\lambda^{(k)}})$. We use the notation \oslash and $\sqrt{\cdot}$ to denote elementwise division and square root respectively.

Proof. The dual problem (D) is

$$\underset{H \in \mathbb{R}^{N \times s}}{\text{minimize}} \quad F^*(H) - (g^* \circ \sigma)(X^\top H)$$

from Proposition 3.5. By using Proposition 3.10 (note that it contains a full SVD and not a compact SVD), the iterations of (DCA) applied to the dual problem are

$$[U^{(k)}, \text{Diag}(\sigma^{(k)}), V^{(k)\top}] \leftarrow \text{SVD}(X^\top H^{(k)}), \quad H^{(k+1)} \leftarrow \partial F^{**}(XU^{(k)}\text{Diag}(\mu^{(k)})V^{(k)\top})$$

where $\mu^{(k)} \in \partial g^*(\sigma^{(k)})$. Since $H^{(k)\top}KH^{(k)}$ is nonsingular, each entry of $\sigma^{(k)}$ is strictly greater than 0 and we may use a compact SVD and compact eigenvalue decomposition. The result then follows from the same arguments as in Subsection C.3 and the fact that $F^{**} = F^*$ [7, Cor. 13.38]. \square

Remark C.9. Instead of the assumption that $H^{(k)\top}KH^{(k)}$ is invertible for each k , the result also holds whenever $y_i = 0$ implies that $0 \in \partial g^*(y)_i$ for each component and $0/0$ is taken to be 0. This is the case for all the functions encountered previously (Propositions C.1 through C.7).

Applying this result to the formulations (l) through (q) (or transposed variations) yields the updates in Table 4. It is clear that many of these algorithms are tightly related. For example, (14) is equivalent to Algorithm 2 as well as (16) since the scaling only affects $\Sigma^{(k)}$ which is discarded. Similarly, (18) is the same as (14) up to scaling. Moreover, (15) is (13) with additional normalization while (15) is also equivalent to (17) since the multiplication with the trace in (17) is void due to the normalization afterwards. In some sense, (DCA) is blind to simple problem transformations. Nevertheless, while they may be mathematically tightly related, the practical performance may differ significantly.

C.5 Derivation of Algorithms 5 and 6

Proof. Recall from the proof of Proposition 4.1 that the primal problem can be written as (P) with $G = \delta_{B_{\|\cdot\|_{S_\infty}}}$ and $F = \sum_{i=1}^N f_i(\cdot)_{i,:}$. The expression for ∂G^* follows immediately from (5) and Proposition C.1, while for ∂F we use [61, Theorem 23.8] to find that the subdifferential of a separable sum is the Cartesian product of the subdifferentials. Moreover, we note that each f_i can be written as $h_i \circ \|\cdot\|$ where

$$h_i : \mathbb{R} \rightarrow \overline{\mathbb{R}} : a \mapsto \begin{cases} -\sqrt{\|X_{i,:}\|^2 - a^2} & \text{if } |a| \leq \|X_{i,:}\|, \\ +\infty & \text{otherwise} \end{cases}$$

such that we can use Fact A.7. More concretely, we have that

$$\partial h_i(a) = \left\{ \frac{a}{\sqrt{\|X_{i,:}\|^2 - a^2}} \right\}$$

for $|a| < \|X_{i,:}\|$ and the empty set otherwise. Combining these subdifferentials with (DCA) yields Algorithm 5. Note that due to the initialization as well as the update rule for $W^{(k)}$, the spectral norm of $W^{(k)}$ is always less than 1 such that the argument of ∂h_i has absolute value $|a| \leq \|X_{i,:}\|$ and the square root is always well-defined. To derive Algorithm 6, the same subdifferentials can be used in combination with the same arguments as in the derivation of Algorithm 4 to make it kernelizable. \square

Remark C.10. Whenever $(XW^{(k)})_{i,:} = 0$ in Algorithm 5, $Y_{i,:}^{(k)}$ can be set to an arbitrary vector in some ball of some radius, in accordance with the second case in Fact A.7. Another edge case occurs whenever $\|X_{i,:}\|^2 = \|(XW^{(k)})_{i,:}\|^2$. In practice, this is alleviated by adding some regularization constant ε in the denominator. Formally, this is accomplished by modifying the first case in the definition of h_i to be $-\sqrt{\|X_{i,:}\|^2 - a^2 + \varepsilon^2}$ for $|a| \leq \sqrt{\|X_{i,:}\|^2 + \varepsilon^2}$ as in [9].

Table 4: DC algorithms for formulations (l) through (q) where the data matrix is $X \in \mathbb{R}^{N \times d}$. Only the kernelizable versions involving $K := XX^\top$ are provided.

Problem	(DCA) dual updates (kernelizable versions)	Derived through
(l)	$[V^{(k)}, \Lambda^{(k)}] \leftarrow \overline{\text{EIG}}(H^{(k)\top}KH^{(k)})$ $H^{(k+1)} \leftarrow KH^{(k)}V^{(k)}\Lambda^{(k)-1/2}V^{(k)\top}$	Proposition C.1 Proposition C.3
(m)	$[U^{(k)}, \Sigma^{(k)}, V^{(k)\top}] \leftarrow \overline{\text{SVD}}(KH^{(k)})$ $H^{(k+1)} \leftarrow U^{(k)}V^{(k)\top}$	Proposition C.3 Proposition C.1
(n)	$[V^{(k)}, \Lambda^{(k)}] \leftarrow \overline{\text{EIG}}(H^{(k)\top}KH^{(k)})$ $\tilde{H}^{(k+1)} \leftarrow KH^{(k)}V^{(k)}\Lambda^{(k)-1/2}V^{(k)\top}$ $H^{(k+1)} \leftarrow \frac{\tilde{H}^{(k+1)}}{\ \tilde{H}^{(k+1)}\ _{S_2}}$	Proposition C.1 Proposition C.4
(o)	$[U^{(k)}, \Sigma^{(k)}, V^{(k)\top}] \leftarrow \overline{\text{SVD}}(\text{tr}(H^{(k)\top}KH^{(k)})^{-1/2}KH^{(k)})$ $H^{(k+1)} \leftarrow U^{(k)}V^{(k)\top}$	Proposition C.4 Proposition C.1
(p)	$[V^{(k)}, \Lambda^{(k)}] \leftarrow \overline{\text{EIG}}(H^{(k)\top}KH^{(k)})$ $\tilde{H}^{(k+1)} \leftarrow \text{tr}(\Lambda^{(k)1/2})KH^{(k)}V^{(k)}\Lambda^{(k)-1/2}V^{(k)\top}$ $H^{(k+1)} \leftarrow \frac{\tilde{H}^{(k+1)}}{\ \tilde{H}^{(k+1)}\ _{S_2}}$	Proposition C.6 Proposition C.4
(q)	$[U^{(k)}, \Sigma^{(k)}, V^{(k)\top}] \leftarrow \overline{\text{SVD}}(KH^{(k)})$ $H^{(k+1)} \leftarrow \text{tr}(H^{(k)\top}KH^{(k)})^{-1/2} \text{tr}(\Sigma^{(k)})U^{(k)}V^{(k)\top}$	Proposition C.4 Proposition C.6

D IRLS and DCA

In this Appendix, we derive an iteratively reweighted least squares (IRLS) algorithm for the robust PCA formulation from Section 4 and compare with Algorithm 5. As is common in the literature (see [45] and references therein), the IRLS update for the robust subspace recovery problem is of the form

$$W^{(k+1)} \in \arg \min_{\substack{W \in \mathbb{R}^{d \times s} \\ \|W\|_{S_\infty} \leq 1}} \sum_{i=1}^N \frac{1}{\beta_i(W^{(k)})} \|X_{i,:} - WW^\top X_{i,:}\|^2$$

where

$$\beta_i(W^{(k)}) = \|X_{i,:} - W^{(k)}W^{(k)\top}X_{i,:}\|.$$

This minimization problem is equivalent to

$$W^{(k+1)} \in \arg \min_{\substack{W \in \mathbb{R}^{d \times s} \\ \|W\|_{S_\infty} \leq 1}} \|BX - WW^\top BX\|_{S_2}^2 = \arg \min_{\substack{W \in \mathbb{R}^{d \times s} \\ \|W\|_{S_\infty} \leq 1}} -\frac{1}{2} \|BXW\|_{S_2}^2$$

where B is a square diagonal matrix with $(\beta_i(W^{(k)}))^{-1/2}$ on the diagonal. The latter is recognized to be (l) such that we may use Proposition 3.2 to obtain a solution of the inner problem. This leads to Algorithm 10, where we also used the fact that if $A = U\Sigma V^\top$ is a SVD of A , then $A^\top = V\Sigma^\top U^\top$ is a SVD of A^\top . Note that for each $k \geq 1$, $W^{(k)\top}W^{(k)} = I_s$ such that $\beta_i^{(k)}$ could equivalently be written as $\sqrt{\|X_{i,:}\|^2 - \|(XW^{(k)})_{i,:}\|^2}$, which is exactly the denominator on line 5 of Algorithm 9.

The main difference between the two algorithms is that an additional square root is taken of $\beta_i^{(k)}$ in Algorithm 10, while in Algorithm 9 a multiplication with $(XW^{(k)})_{i,:}$ is performed instead. Another difference is that an orthogonal factor remains in Algorithm 9, which is reminiscent of the connection between Algorithm 7 and simultaneous iteration, see Theorem 3.11.

Algorithm 9 (DCA) for Proposition 4.1 (primal)

Require: Matrix $X \in \mathbb{R}^{N \times d}$

- 1: Initialize $W^{(0)} \in \mathbb{R}^{d \times s}$ s.t. $\|W^{(0)}\|_{S_\infty} = 1$
- 2: Initialize $k \leftarrow 0$
- 3: **repeat**
- 4: **for** $i = 1$ **to** N **do**
- 5: $Y_{i,:}^{(k)} \leftarrow \frac{(XW^{(k)})_{i,:}}{\sqrt{\|X_{i,:}\|^2 - \|(XW^{(k)})_{i,:}\|^2}}$
- 6: **end for**
- 7: $[U^{(k)}, \Sigma^{(k)}, V^{(k)\top}] \leftarrow \text{SVD}(X^\top Y^{(k)})$
- 8: $W^{(k+1)} \leftarrow U^{(k)} V^{(k)\top}$
- 9: $k \leftarrow k + 1$
- 10: **until** convergence

Algorithm 10 IRLS for robust subspace recovery

Require: Matrix $X \in \mathbb{R}^{N \times d}$

- 1: Initialize $W^{(0)} \in \mathbb{R}^{d \times s}$ s.t. $\|W^{(0)}\|_{S_\infty} = 1$
- 2: Initialize $k \leftarrow 0$
- 3: **repeat**
- 4: **for** $i = 1$ **to** N **do**
- 5: $\beta_i^{(k)} \leftarrow \|X_{i,:} - W^{(k)} W^{(k)\top} X_{i,:}\|$
- 6: **end for**
- 7: $B^{(k)} \leftarrow \text{Diag}(\beta^{-1/2})$
- 8: $[U^{(k)}, \Sigma^{(k)}, V^{(k)\top}] \leftarrow \text{SVD}(X^\top B^{(k)})$
- 9: $W^{(k+1)} \leftarrow U_{:,1:s}^{(k)}$
- 10: $k \leftarrow k + 1$
- 11: **until** convergence

E DC formulations for sparse PCA

The DC duality framework is very broad. We now discuss two promising avenues for sparse PCA. First, similar to formulation (q), we can consider the DC formulation

$$\underset{W \in \mathbb{R}^{d \times s}}{\text{minimize}} \quad \frac{1}{2} \|W\|_S^2 - \|XW\|_{S_2},$$

where $\|\cdot\|_S$ is a sparsity inducing norm such as taking the sum of the absolute value of its elements (l_1 -norm on its vectorization). Note that this is not a classical sparse PCA formulation but achieves a similar purpose. The convex conjugate of this function follows from (1).

Second, if we consider only one principal component, then a DC formulation for sparse PCA is described in [9]. This formulation takes the form

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \quad \delta_C(w) - \frac{1}{2} \|Xw\|_{S_2}^2,$$

where C is the convex hull of the intersection of the unit norm ball and $\{w \in \mathbb{R}^d \mid \|w\|_0 \leq k\}$. The conjugate of this indicator function can be computed in closed form. Indeed, $\delta_C^*(z) = \max_{w \in C} w^\top z$ by definition. Then, since w has at most k nonzeros, we see that this maximization problem is solved when the support of w corresponds to the k largest entries of z in absolute value. Taking into account that w is also a unit vector, it follows that $\delta_C^*(z)$ is exactly the Euclidean norm of the vector containing the k largest entries of z in absolute value.

Both of these formulations are not unitarily invariant and therefore not easily kernelizable through Proposition 3.5 (a sufficient condition for kernelizability). Though, to the best of our knowledge, sparse PCA is generally not used in the kernel setting since the main motivation for sparse PCA is interpretability of the data and principal components. More concretely, by restricting the cardinality of the principal components, this implicitly assumes that they are linear combinations of just a few input variables, which does not mesh well with kernel methods that are inherently nonlinear.

F Additional experiments

In this Appendix, we provide further experimental results.

Higher accuracy Timing results for higher target accuracies than those in Section 5 are presented in Table 5. The problem instances are generated in a similar manner. We observe that the Krylov-type methods are more suited than the first-order methods in this setting. Nevertheless, ZeroFPR applied to (n) still outperforms the Krylov methods for larger matrices. We observe that the (DCA) algorithms are quite slow, which is mostly due to the relatively expensive check of the stopping criterion at each iteration.

Table 5: Timing results for various methods applied to PCA formulations with higher accuracies. The problem setting (N, d, s, ε) is the same as in Table 1. All timings are in milliseconds, and timings longer than 5 seconds are not displayed.

Method	(3000, 2000, 20, 10^{-5})	(2000, 3000, 20, 10^{-5})	(3500, 3500, 20, 10^{-5})
ZeroFPR (l)	2009.0 \pm 122.5	2202.6 \pm 296.0	4269.3 \pm 57.6
ZeroFPR (n)	735.0 \pm 295.7	876.1 \pm 194.8	1269.1 \pm 357.0
ZeroFPR (o)	1092.5 \pm 285.1	1532.8 \pm 213.1	2577.7 \pm 274.3
ZeroFPR (q)	2018.4 \pm 141.1	2148.0 \pm 319.6	4245.3 \pm 715.5
PG (n)	/	3569.9 \pm 410.1	/
PG (o)	4011.7 \pm 969.8	/	/
Algorithm 1	4709.5 \pm 3963.0	/	/
Algorithm 2	/	2559.5 \pm 832.2	/
Algorithm 3	/	/	/
Algorithm 4	/	/	/
svDS	625.3 \pm 10.6	556.9 \pm 18.7	1564.8 \pm 31.1
KrylovKit	773.8 \pm 29.7	696.0 \pm 17.5	1858.7 \pm 24.3

Different spectra The performance of Krylov methods for computing the top s singular values is known to be highly sensitive to the spectrum, and in particular the spectral gaps σ_i/σ_{i+1} . Table 6 reports timing results for various methods applied to matrices with different fixed spectra. In the first column, where the singular values decay exponentially, all methods perform equally well. In the second column, the singular values decrease linearly from 500 to 100, resulting in small spectral gaps. Here, the Krylov methods are roughly 10 times slower compared to the exponential decay case. Notably, ZeroFPR and PG on the formulations (n) and (o) significantly outperform the Krylov solvers. This trend continues in the third column, where the singular values decrease linearly from 500 to 10^{-8} .

Table 6: Timing results for various methods applied to PCA formulations for matrices with specified spectrum. All matrices are square with $N = 3500$. A stopping criterion tolerance of 10^{-3} was used to find $s = 20$ principal components. All timings are in milliseconds, and timings longer than 5 seconds are not displayed.

Method	$\sigma_i = 100 \cdot 0.9^i$	$\sigma_i = 500 - (i - 1) \frac{400}{N-1}$	$\sigma_i = 500 - (i - 1) \frac{500-10^{-8}}{N-1}$
ZeroFPR (l)	490.7 \pm 123.5	/	/
ZeroFPR (n)	171.9 \pm 64.8	456.2 \pm 90.1	455.7 \pm 80.0
ZeroFPR (o)	260.1 \pm 68.1	1043.3 \pm 164.0	1137.7 \pm 165.6
ZeroFPR (q)	298.5 \pm 97.2	/	/
PG (n)	115.8 \pm 22.1	730.2 \pm 55.0	729.5 \pm 71.1
PG (o)	262.1 \pm 40.3	1068.8 \pm 34.3	1173.2 \pm 88.5
Algorithm 1	300.2 \pm 13.9	3633.2 \pm 184.8	3581.2 \pm 554.2
Algorithm 2	295.7 \pm 14.5	3594.1 \pm 36.0	3866.0 \pm 59.4
Algorithm 3	429.7 \pm 19.2	/	/
Algorithm 4	387.1 \pm 16.2	/	/
svDS	288.9 \pm 2.1	3113.9 \pm 133.4	2588.5 \pm 178.1
KrylovKit	242.8 \pm 3.1	4250.3 \pm 392.9	3653.5 \pm 309.8

Real-world datasets Table 7 shows the timing results of the best performing methods on the MNIST dataset [22] ($N = 60000, d = 784$) with a tolerance of $\varepsilon = 10^{-3}$. We observe the same behavior as in the synthetic experiments: the generic first-order methods are faster than the classical eigensolvers in this setting.

Table 8 shows the timing results of the best performing methods on the 100k top words from the 2024 Wikipedia + Gigaword 5, 50d GloVe word embedding dataset [57, 16] ($N = 100000, d = 50$) with a tolerance of $\varepsilon = 10^{-3}$. While the first-order methods are still faster than the classical eigensolvers,

this difference is less pronounced. We attribute this result to the fact that these word embeddings already form a compressed version of the word vector space, and therefore have favorable spectral characteristics, similar to the results encountered in the first column of Table 6.

Table 7: Timing results for different methods of applying PCA to the MNIST dataset with a tolerance of $\varepsilon = 10^{-3}$. The column headers denote the number of principal components. All timings are in milliseconds, and timings longer than 5 seconds are not displayed.

Method	30	50	100	150
ZeroFPR (n)	286.7 \pm 92.2	444.7 \pm 86.1	1734.4 \pm 247.0	2363.3 \pm 135.2
PG (n)	215.5 \pm 35.7	351.9 \pm 62.4	1254.8 \pm 103.1	1772.2 \pm 150.5
SVDS	1685.1 \pm 226.2	2576.5 \pm 4.5	/	/

Table 8: Timing results for different methods of applying PCA to the GloVe word embedding dataset with a tolerance of $\varepsilon = 10^{-3}$. The column headers denote the number of principal components. All timings are in milliseconds.

Method	10	15	20	30
ZeroFPR (n)	131.1 \pm 114.5	208.2 \pm 150.4	271.8 \pm 147.2	367.1 \pm 198.2
PG (n)	76.0 \pm 71.4	120.0 \pm 100.4	165.3 \pm 111.9	250.4 \pm 113.2
SVDS	200.6 \pm 14.6	281.0 \pm 31.4	319.7 \pm 11.5	329.0 \pm 34.5

Additional formulations In Table 9, we present some timings for the algorithms described in Table 4, where we used the relative error with respect to the optimal value for the corresponding formulation as a stopping criterion. Wherever possible, we reuse information from the SVDs/EIGs to cheaply evaluate the objective value. We observe that in general, the algorithms using eigenvalue decompositions are faster than those involving singular value decompositions. Moreover, the algorithms are on par with the Krylov solvers whenever the number of features is greater than or equal to the number of datapoints. This is to be expected since the algorithms use the kernel matrix. In the opposite setting, one should use algorithms using the (scaled) covariance matrix instead.

Table 9: Timing results for methods from Table 4 applied to PCA formulations. The problem setting (N, d, s, ε) is the same as in Table 1. All timings are in milliseconds.

Method	(3000, 2000, 20, 10^{-3})	(2000, 3000, 20, 10^{-3})	(3000, 3000, 20, 10^{-3})
Iteration (13)	1135.6 \pm 72.7	520.4 \pm 58.3	1198.9 \pm 153.3
Iteration (14)	1327.5 \pm 331.8	567.6 \pm 78.6	1179.1 \pm 272.4
Iteration (15)	792.4 \pm 88.1	338.1 \pm 42.3	800.7 \pm 79.9
Iteration (16)	1059.4 \pm 65.5	556.6 \pm 58.7	1182.2 \pm 94.0
Iteration (17)	1742.6 \pm 206.4	774.6 \pm 84.4	1815.1 \pm 71.7
Iteration (18)	5926.6 \pm 149.6	1085.7 \pm 76.0	4895.1 \pm 94.9
SVDS	638.1 \pm 41.7	522.3 \pm 44.4	1164.0 \pm 57.5
KrylovKit	755.5 \pm 26.3	685.4 \pm 44.3	1215.7 \pm 43.5

A note on tolerance Previous comparisons used a fixed tolerance ε across methods, but this is not ideal since (1) the problem formulations differ, and (2) algorithms interpret ε differently in their stopping criteria. To see this impact, we constructed 10 different 50×50 matrices with entries sampled from the standard normal distribution. Then, for each tolerance from 10^{-1} down to 10^{-9} on a log-spaced grid, we computed $s = 10$ components, and evaluated the relative error with respect to the optimal solution on the formulations (n) or (o). We averaged these errors over the 10 different matrices and present the results in Figure 4. We observe that KrylovKit.jl reaches machine precision for the function value with $\varepsilon = 10^{-3}$ while ZeroFPR and PG perform the worst on (n) and (o).

Robust (kernel) PCA Consider the MNIST dataset [22] with a train-test split of 80-20. To verify the robustness properties of the robust PCA formulation in Section 4, we contaminate 15% of the

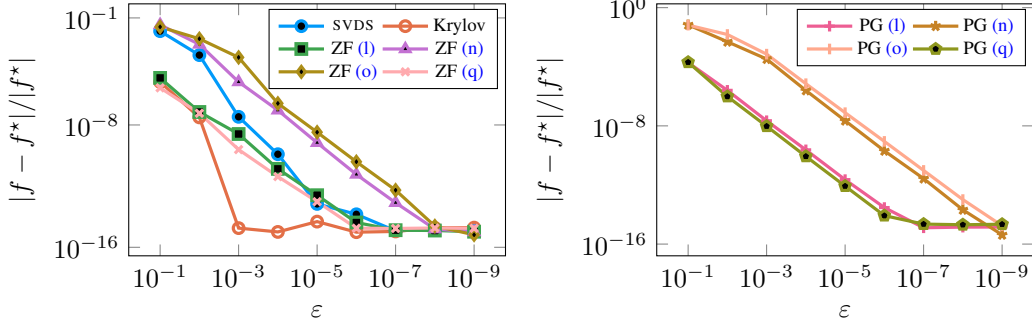


Figure 4: Relative error of the objective value (n) or (o) for multiple methods with varying tolerance for the stopping error criterion. f^* denotes the optimal value while ZF is shorthand for ZeroFPR.

training data with heavy Gaussian noise ($\sigma = 15$) and leave the test set untouched. We consider four different settings:

1. Linear PCA on the noncontaminated data (baseline)
2. Linear PCA on the contaminated data
3. Robust PCA on the contaminated data using Algorithm 5
4. Robust PCA on the contaminated data using Algorithm 6 with a linear kernel (note that the kernel matrix does not fit in memory but we can use its factored representation since the kernel is linear)

For each of these settings, we evaluate the reconstruction error on the noncontaminated test set. These errors are summarized in Table 10. The robust methods outperform standard linear PCA in the contaminated setting. Further, the similar performance of the two robust methods is to be expected, due to strong duality. Lastly, the top components are less affected by the outliers than the remaining components. This is logical since these latter components explain less ‘variance’ and it becomes more difficult to distinguish noise/outliers from data.

Table 10: Reconstruction errors on the test set of various PCA schemes. The column headers denote the number of principal components. The robust PCA formulations outperform standard linear PCA on the contaminated dataset.

Method	50	100	150
1. Linear PCA (noncontaminated)	0.011	0.0057	0.0034
2. Linear PCA (contaminated)	0.062	0.0569	0.0520
3. Robust PCA (contaminated) with Algorithm 5	0.014	0.0107	0.0090
4. Robust PCA (contaminated) linear kernel with Algorithm 6	0.014	0.0107	0.0090

To further illustrate the strength and usecase of this formulation, we can now extract robust features in light of Theorem 3.8. To this end, for each of the settings, we train a small multilayer perceptron classifier (1 hidden layer with 20 neurons) on these extracted features. Note that we choose a very simple classifier so the quality of the features becomes more apparent. Additionally, we also consider the following two settings:

5. Robust PCA on the noncontaminated data using Algorithm 5
6. Robust kernel PCA on the contaminated data using Algorithm 6 with a RBF kernel (length scale paramter $\gamma = 0.01$) approximated using a Nyström approximation with 500 pivots

The test accuracies of each classifier are displayed in Table 11. We observe that the robust formulations always perform better than the non-robust formulations for the contaminated data. The robust formulation on the noncontaminated data (setting 5) also does not degrade in performance with respect to the baseline (setting 1). Lastly, the RBF kernel trained on the contaminated data performs on par with models trained on uncontaminated data.

Table 11: Test accuracies of a small multilayer perceptron after feature extraction.

Method	50	100	150
1. Linear PCA (noncontaminated)	95.87	95.73	94.84
2. Linear PCA (contaminated)	76.57	79.93	81.09
3. Robust PCA (contaminated) with Algorithm 5	86.89	87.32	85.53
4. Robust PCA (contaminated) linear kernel with Algorithm 6	87.63	84.44	86.92
5. Robust PCA (noncontaminated) with Algorithm 5	95.73	96.06	95.43
6. Robust PCA (contaminated) RBF kernel with Algorithm 6	95.21	95.43	94.34

As a last experiment, we compare the robust PCA formulation with the classical robust PCA [15]. Note that the two approaches are of a different nature. The classical method decomposes a matrix into a low-rank component and a sparse component, which is well-suited for motion segmentation. In contrast, the formulation in Section 4 is based on the autoencoder perspective and aims to enforce sparsity on the reconstruction errors, which is better aligned with outlier detection. Another major difference is that the classical method is not out-of-sample applicable while ours is through Theorem 3.8. These differences make fair comparisons difficult.

To further illustrate this fact, consider the synthetic experiment from [15, Section 4.1]. We first generate a low-rank component L^* according to the same settings and add a sparse matrix E where E has limited support, with the nonzero entries of E being independent Bernoulli ± 1 entries. We consider two settings:

1. The support of E is chosen uniformly distributed over all its entries
2. The support of E consists exclusively of full rows (i.e., an image where complete rows are perturbed)

We then compare the classical principal component pursuit (PCP) with alternating directions [15, Algorithm 1] with Algorithm 5. For our experiment, we choose $\text{rank}(L^*) = 25$ and $L^* \in \mathbb{R}^{500 \times 500}$. We assume the size of the support of E is $|E| = 12500$ and compute both the reconstruction error of the low-rank component and the primal cost of Proposition 4.1 (i.e., the l_1 -norm of the row-wise reconstruction errors) and summarize our results in Table 12 (all metrics were averaged over 20 different random problems). We observe that the classical PCP performs better in the setting where it was proposed while our algorithm is better in the outlier setting. Moreover, we see that the primal cost is lower for our algorithm in both cases, which is to be expected since the PCP algorithm is not designed to minimize this cost.

Table 12: Comparison principal component pursuit (PCP) and Algorithm 5 for decomposing a matrix into a low-rank component L and a sparse component E .

Setting	Method	$\frac{\ L-L^*\ _{S_2}}{\ L^*\ _{S_2}}$	Primal cost
Uniform perturbation	PCP	3×10^{-8}	2488
	Algorithm 5	0.013	2364
Full row perturbations	PCP	0.045	559
	Algorithm 5	0.009	545