Structure-Preserving Non-Linear PCA for Matrices

Joni Virta and Andreas Artemiou

***Abstract*—We propose a new dimension reduction method for matrix-valued data called Matrix Non-linear PCA (MNPCA), which is a non-linear generalization of (2D)^{2}PCA. MNPCA is based on optimizing over separate non-linear mappings on the left and right singular spaces of the observations, essentially amounting to the decoupling of the two sides of the matrices. We develop a comprehensive theoretical framework for MNPCA by viewing it as an eigenproblem in reproducing kernel Hilbert spaces. We study the resulting estimators on both population and sample levels, deriving their convergence rates and formulating a coordinate representation to allow the method to be used in practice. Simulations and a real data example demonstrate MNPCA's good performance over its competitors.**

***Index Terms*—(2D)^{2}PCA, dimension reduction, kernel methods, matrix data.**

I. Introduction

THE diverse types of data encountered in modern applications have caused a surge in the development of statistical methods specializing to datasets that do not exhibit the standard form of samples of points in $\mathbb{R}^p$. In this work, we focus on one of these special cases, matrix-valued data, where we observe n matrices, $X_1, \ldots, X_n$, each having the size $p_1 \times p_2$. In typical applications, such as imaging, the dimensions p_1, p_2 can be very large in size and the first step of the analysis is often dimension reduction.

One of the most well-known statistical dimension reduction techniques for matrix-valued data is a generalization of the classical PCA known as (2D)^{2}PCA [1], where the observed matrices are replaced with $(d_1 \times d_2)$-sized latent matrices $Z_i := A'(X_i - \bar{X})B$ where $\bar{X}$ is the sample mean matrix and A, B contain, respectively, any first d_1 and d_2 eigenvectors of the matrices

$$\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})' \quad \text{and} \quad \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})'(X_i - \bar{X}).$$

Manuscript received 10 October 2023; revised 22 March 2024; accepted 22 July 2024. Date of publication 1 August 2024; date of current version 16 August 2024. The work of Joni Virta was supported by the Academy of Finland under Grant 335077, Grant 347501, and Grant 353769. The associate editor coordinating the review of this article and approving it for publication was Prof. George Atia. *(Corresponding author: Joni Virta.)*

Joni Virta is with the Department of Mathematics and Statistics, University of Turku, 20014 Turku, Finland (e-mail: joni.virta@utu.fi).

Andreas Artemiou is with the Department of Information Technologies, University of Limassol, Nicosia 2151, Cyprus (e-mail: artemiou@uol.ac.cy).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2024.3437183, provided by the authors.

Digital Object Identifier 10.1109/TSP.2024.3437183

This reduction can be seen to be natural in the following two senses. (a) (2D)^{2}PCA takes as its input a sample of matrices and gives as its output a sample of matrices, essentially *preserving the type of the data*. This is not the case for many dimension reduction methods (such as the kernel methodology listed below) which instead produce samples of score vectors whose relation to the matrix structure of the original data is not clear. (b) The latent matrices Z_i exhibit a row-column dependency structure where two elements of Z_i that share a row (column) also share the same column of A (B). That is, denoting the columns of A and B by a_k and b_ℓ, respectively, we have $z_{i,k\ell} = a_k'(X_i - \bar{X})b_\ell$, showing that all latent variables on the kth row of Z_i use the column a_k of A in their computation, and similarly for the columns of Z_i. These properties let (2D)^{2}PCA properly leverage the structure of the observed matrices and see them as more than simply collections of elements. (2D)^{2}PCA has been used to great success in various applications, such as face recognition [1] and stock price prediction [2].

In this work we propose Matrix Non-linear PCA (MNPCA), a non-linear extension of (2D)^{2}PCA that retains both of the properties listed in the previous paragraph. That is, we construct a non-linear mapping $X_i \mapsto g(X_i) =: Z_i$ such that (a) $Z_1, \ldots, Z_n$ are $(d_1 \times d_2)$-sized matrices, (b) g imposes specific dependencies between the rows and between the columns of the images Z_i (i.e., two elements sharing a row are more similar than elements on different rows), and (c) when a linear kernel is used, the mapping g reduces to the usual (2D)^{2}PCA. Analogous to (2D)^{2}PCA, our non-linear mapping g can thus be seen to preserve the matrix structure of the original data. This behavior of g is in strict contrast to existing methods of non-linear dimension reduction for matrix-valued data which we review next. All of the methods listed below are based either on (2D)^{2}PCA [1] or its one-sided precursor 2DPCA [3] which applies only the transformation A or B but not both.

The authors in [4] defined kernel 2DPCA (K2DPCA), which is essentially equivalent to applying the standard kernel PCA to the set of all np_1 rows in the matrix sample. Independently, [5] proposed a method equivalent to K2DPCA of [4]. In [6] K2DPCA was applied separately to the rows and columns of the input matrices and standard PCA was used on the resulting pairs of latent representations to obtain combined latent variables. Whereas, [7] first used K2DPCA to reduce the number of rows in the data and then applied regular 2DPCA to the obtained latent matrices to reduce also their column dimension. To summarize, the literature on non-linear extensions of two-dimensional PCA either focuses on one-sided methods (extensions of 2DPCA) or combines a pair of one-sided methods into

a two-sided method (i.e., into one that reduces both rows and columns simultaneously) in a theoretically cumbersome way. Both options can be seen as sub-optimal: If the number of columns in the data is even moderately large, reducing only the number of rows still leaves the data dimension high, making, e.g., the visualization of the resulting components impossible. Whereas, in chaining the row and column reductions, the outcome either depends non-trivially on the order in which the rows and columns are reduced [7] or is artificial and loses the structural connection to the original matrices [6].

A further problem underlying the methods listed above is their high computational complexity. Namely, as K2DPCA operates on the sample of all np_1 rows of the data, the size $np_1 \times np_1$ of the corresponding kernel matrix can be enormous even for combinations of a moderate sample size n and number of rows p_1. To combat this, [6] propose approximating the full kernel matrix with the kernel matrix of the within-observation row means of the data, but this leads to the loss of the row structure and it is not clear how it affects the accuracy of the method.

Our proposed method for constructing the non-linear mapping $X_i \mapsto g(X_i)$ avoids the previous pitfalls by working not with the observed matrices X_i themselves but with their singular value decompositions (SVD), $X_i = U_i D_i V_i'$. This simple change of perspective has two major implications:

1) The singular value decomposition essentially "decouples" the row and column information in the input matrices, allowing the independent and simultaneous reduction of the row and column spaces, possibly with different kernel functions. As a consequence, MNPCA is fully two-sided and the order in which the two sides are reduced does not affect the outcome.
2) As leading singular vectors capture the main directions of data variation, truncating the SVD allows us to leverage (almost) the full data information while simultaneously avoiding the inflation of the size of the kernel matrix.

The primary contributions of this work are: (i) We formulate the population-level version of MNPCA, our proposed SVD-based non-linear extension of (2D)^{2}PCA. As is typical in the literature on non-linear dimension reduction [8], this requires casting the problem into the framework of reproducing kernel Hilbert spaces (RKHS) and Hilbert-Schmidt operators. Special attention is paid to formulating the exact assumptions under which MNPCA is well-defined. (ii) We study the asymptotic properties of MNPCA and derive the convergence rate of the corresponding estimator. (iii) We derive a coordinate representation for MNPCA, allowing its sample-level implementation, and discuss the selection of its tuning parameters. (iv) We compare MNPCA to several of its competitors using both simulations and a real data example. Note that earlier work on the non-linear extensions of (2D)^{2}PCA (see the list of references earlier) has focused solely on points (iii) and (iv), ignoring the finer theoretical aspects of the corresponding methods.

We briefly note that the structure-acknowledging non-linear dimension reduction of matrices can be approached also from another viewpoint besides (2D)^{2}PCA. Namely, [9] apply standard kernel PCA to the sample $X_1, \ldots, X_n$ but with a very specific choice of kernel that recognizes the matrix structure of the data. While interesting, this approach goes somewhat against the spirit of kernel methodology where the kernel function is typically seen as a tuning parameter and its choice is equivalent to determining what kind of latent structures one is after. The limiting to a single kernel in [9] is in strict contrast to our proposed method which allows the use of any kernel function that is either odd or even, see the definition in Section II. This restriction is a mild one as the classes of odd and even kernels are very large (in fact, every positive semi-definite kernel induces both a corresponding odd and an even kernel). Finally, we remark that, from an abstract viewpoint, the objective of MNPCA is to represent the observed data as elements of a specific Hilbert space (non-linear features) and then find an efficient low-rank representation for these elements (dimension reduction). Other works that explore similar ideas include, for example, [10], [11].

The manuscript is organized as follows. Section II begins with some definitions and notation. In Section III we use the combination of SVD and even/odd kernels to motivate a well-defined non-linear analogy for the two-sided projection of a matrix. In Section IV we formulate the population version of the MNPCA-procedure. Section V focuses on the asymptotic properties of the sample version of the method, whereas its coordinate representation is constructed in Section VI. Tuning parameter selection is discussed in Section VII. Simulations and real data examples are given in Sections VIII and IX concludes with a discussion. All proofs of technical results are collected to the supplementary Appendix A, where a brief explanation of the used proof techniques is also given.

II. Notation and Definitions

For convenience, we have collected all the relevant notation in this section. As is standard, we use the word kernel to refer to any $\kappa : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ that is symmetric and positive semi-definite [12]. Our main theoretical tool in this work are (kernel-induced) reproducing kernel Hilbert spaces (RKHS) and, for the convenience of the reader, we next briefly review the main idea behind them.

Every kernel $\kappa : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ induces a Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ of functions f from $\mathbb{R}^p$ to $\mathbb{R}$. The members of the space $\mathcal{H}$ are essentially all possible functions of the form $\sum_{j=1}^{m} a_j \kappa(\cdot, x_j)$, where $m \in \mathbb{N}$, $a_j \in \mathbb{R}$ and $x_j \in \mathbb{R}^p$, along with their limits when $m \to \infty$. As we see next, particularly important members of $\mathcal{H}$ are the representations, or "features", $\kappa(\cdot, x) \in \mathcal{H}$, $x \in \mathbb{R}^p$ (that is, the representation $\kappa(\cdot, x)$ of a point x is a function). What makes the RKHS theoretically attractive is the *reproducing property*: for every $f \in \mathcal{H}$ and $x \in \mathbb{R}^p$, we have $f(x) = \langle f, \kappa(\cdot, x) \rangle$. That is, the evaluation $f(x)$ of a non-linear map f at x can be represented linearly (inner products are linear) in the *feature space* of the representative functions $\kappa(\cdot, x)$. This essentially allows us to formulate non-linear methods linearly and to efficiently study their behavior using tools of functional analysis. Finally, as the reproducing property holds only for members f of the space $\mathcal{H}$, we note that the choice of the kernel κ implicitly determines the set of

non-linear mappings we can consider. For further information on RKHS, we refer the reader to [13].

Let next κ_1, κ_2 be continuous kernels. We denote the RKHS induced by κ_1 and κ_2 by $(\mathcal{H}_1, \langle\cdot,\cdot\rangle), (\mathcal{H}_2, \langle\cdot,\cdot\rangle)$, respectively. That is, the Hilbert spaces $\mathcal{H}_1, \mathcal{H}_2$ are implicitly defined by the two kernels in the manner described in the previous paragraph. To avoid notational overload, we use the same symbols $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ for the inner products and norms, respectively, of both $\mathcal{H}_1$ and $\mathcal{H}_2$. The actual space in question can always be understood from the context. We use the notation $\mathcal{B}(\mathcal{H}_i, \mathcal{H}_j)$ to refer to the set of all bounded linear operators from $\mathcal{H}_i$ to $\mathcal{H}_j$. By the continuity of the kernels κ_1, κ_2, the spaces $\mathcal{H}_1, \mathcal{H}_2$ are separable and admit orthonormal bases [14].

Recall that an operator $F \in \mathcal{B}(\mathcal{H}_i, \mathcal{H}_j)$ is said to be a Hilbert-Schmidt operator if the quantity $\sum_{k=1}^{\infty}\sum_{\ell=1}^{\infty}\langle v_\ell, Fu_k\rangle^2$ is finite for some orthonormal bases $\{u_k\}$ and $\{v_\ell\}$ of $\mathcal{H}_i$ and $\mathcal{H}_j$, respectively, in which case its value does not depend on the choice of the bases and is termed the squared Hilbert-Schmidt norm $\|F\|_{\mathrm{HS}}^2$ of F. The vector space $\mathcal{S}(\mathcal{H}_i, \mathcal{H}_j)$ of all Hilbert-Schmidt operators in $\mathcal{B}(\mathcal{H}_i, \mathcal{H}_j)$ itself becomes a Hilbert space when endowed with the inner product

$$\langle F_1, F_2\rangle_{\mathrm{HS}} := \sum_{k=1}^{\infty}\sum_{\ell=1}^{\infty}\langle v_\ell, F_1u_k\rangle\langle v_\ell, F_2u_k\rangle.$$

As with the norms and inner products of the RKHS earlier, the notations $\|\cdot\|_{\mathrm{HS}}\langle\cdot,\cdot\rangle_{\mathrm{HS}}$ leave implicit the actual domains of the Hilbert-Schmidt norm and inner product. For $f \in \mathcal{H}_i$ and $g \in \mathcal{H}_j$, the tensor product $g \otimes f$ denotes the element of $\mathcal{S}(\mathcal{H}_i, \mathcal{H}_j)$ acting as $(g \otimes f)h = \langle f, h\rangle g$ for any $h \in \mathcal{H}_i$. It is straightforwardly checked that $\|g \otimes f\|_{\mathrm{HS}} = \|f\|\|g\|$. Finally, we note that the Hilbert-Schmidt norms satisfy $\langle F_1, F_2\rangle = \langle F_1^*, F_2^*\rangle_{\mathrm{HS}}$ and $\langle g, F_1 f\rangle = \langle (g \otimes f), F_1\rangle_{\mathrm{HS}}$ for all $F_1, F_2 \in \mathcal{S}(\mathcal{H}_i, \mathcal{H}_j)$, $f \in \mathcal{H}_i$ and $g \in \mathcal{H}_j$, where F^* denotes the adjoint of the operator F.

A key role in our development is played by the so-called even and odd kernels. A kernel κ is said to be odd if $\kappa(-x, y) = -\kappa(x, y)$ for all $x, y \in \mathbb{R}^p$. Analogously, a kernel κ is said to be even if $\kappa(-x, y) = \kappa(x, y)$ for all $x, y \in \mathbb{R}^p$. The following lemma, given originally as Corollary 1 in [15], details a simple way of constructing odd/even kernels.

Lemma 1: Let $\kappa : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ be a kernel that satisfies $\kappa(x, y) = \kappa(-x, -y)$ for all $x, y \in \mathbb{R}^p$. Then,

(a) The function $\kappa^- : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ acting as $(x, y) \mapsto \kappa(x, y) - \kappa(-x, y)$ is an odd kernel.
(b) The function $\kappa^+ : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ acting as $(x, y) \mapsto \kappa(x, y) + \kappa(-x, y)$ is an even kernel.

Lemma 1 essentially states that, given any kernel κ satisfying $\kappa(x, y) = \kappa(-x, -y)$, one can always construct the corresponding odd and even kernels κ^-, κ^+. In the sequel, we say that κ^-, κ^+ are the odd and even kernels *induced* by κ.

III. Two-Sided Non-Linear Mapping

Let X be a random $p_1 \times p_2$ matrix. In linear dimension reduction for matrix-valued random variables, the objective is to search for directions $a \in \mathbb{S}^{p_1-1}$, $b \in \mathbb{S}^{p_2-1}$, where $\mathbb{S}^{p-1}$ denotes the unit sphere in $\mathbb{R}^p$, such that the two-sided projection $a'Xb$ provides a meaningful reduction. Having obtained the directions $a_1, \ldots, a_{d_1}$ and $b_1, \ldots, b_{d_2}$ (typically under orthogonality constraints within the two sets), their combinations yield a total of $d_1 d_2$ projections that are most conveniently arranged into the $d_1 \times d_2$ matrix $Z := (a_j' X b_k)_{j=1}^{d_1}{}_{k=1}^{d_2}$. This dimension reduction can be seen to produce a latent variable with a natural matrix structure as, indeed, each row of Z shares the same a-vector and each column of Z the same b-vector. Methods subscribing to this paradigm include, e.g., [1], [16], [17], [18], [19], [20].

In this section we formulate a *non-linear* extension of this concept (simultaneous two-sided projection) using RKHS. Our objective with the extension is to preserve the previous idea that extracting a total of d_1 "left" elements and d_2 "right" elements gives us a $d_1 \times d_2$ reduced matrix where the latent variables on a same row share the same row element and similarly for the columns. In the sequel, we let $\kappa_1 : \mathbb{R}^{p_1} \times \mathbb{R}^{p_1} \to \mathbb{R}$ and $\kappa_2 : \mathbb{R}^{p_2} \times \mathbb{R}^{p_2} \to \mathbb{R}$ denote the kernels corresponding to the two sides and we make the following assumption regarding them.

Assumption 1: The kernels κ_1, κ_2 are either both odd or both even.

The oddness/evenness of the two kernels is required later on to ensure that the lack of fixed signs in singular value decomposition does not compromise the uniqueness of our non-linear mappings. We additionally make the following assumption regarding the random matrix X.

Assumption 2: For some $r \leq \min\{p_1, p_2\}$, the random matrix X has almost surely rank r and its non-zero singular values are almost surely simple.

The first part of Assumption 2 (almost surely fixed rank) is made for convenience and could easily be omitted at the cost of more cluttered notation. We also note that the value of r depends on the exact distribution of X. For example, if X is drawn uniformly from a ball of finite radius in $\mathbb{R}^{p_1 \times p_2}$, then $r = \min\{p_1, p_2\}$, see [21, Corollary 1.2]. The second part (almost surely simple singular values) is satisfied, in particular, if X has an absolutely continuous distribution w.r.t. the Lebesgue measure (in which case the rank is $r = \min\{p_1, p_2\}$ almost surely).

Let $(u_j, v_j) \equiv (u_j(X), v_j(X)), j = 1, \ldots, r$, denote any pair of jth left and right singular vectors of the random matrix X. Under Assumption 2, each of the pairs (u_j, v_j), $j = 1, \ldots, r$, is almost surely uniquely defined up to the joint sign of the members of the pair. That is, if (u_j, v_j) is a jth singular pair of X, then the only other jth singular pair of X is $(-u_j, -v_j)$. We denote the jth singular value of X by $\sigma_j \equiv \sigma_j(X)$.

Let now $f \in \mathcal{H}_1$ and $g \in \mathcal{H}_2$ be arbitrary functions that play the role of the projection directions $a \in \mathbb{R}^{p_1}, b \in \mathbb{R}^{p_2}$ in our non-linear extension. We define the two-sided mapping of X to the pair (f, g) to be

$$\sum_{j=1}^{r}\sigma_j f(u_j)g(v_j) = \left\langle f, \left(\sum_{j=1}^{r}\sigma_j\{\kappa_1(\cdot, u_j) \otimes \kappa_2(\cdot, v_j)\}\right) g\right\rangle =: \langle f, Ug\rangle, \tag{1}$$

where the random operator $U \equiv U(X) := \sum_{j=1}^{r}\sigma_j\{\kappa_1(\cdot, u_j) \otimes \kappa_2(\cdot, v_j)\}$ can be seen as the non-linear "representation"

of the random matrix X. It is easy to show that U is a Hilbert-Schmidt operator, see (2) in Section IV. The mapping (1) is an exact non-linear analogy of the linear projection $a'Xb = \sum_{j=1}^{r} \sigma_j a' u_j b' v_j$, to which it reduces when the kernels κ_1, κ_2 are linear. The oddness or evenness of the kernels κ_1, κ_2 guarantees that the joint sign of any individual singular pair plays no role in the construction of the mapping. E.g., if both kernels are odd, we have for all $u \in \mathbb{R}^{p_1}$, $v \in \mathbb{R}^{p_2}$ that

$$\begin{aligned}\kappa_1(\cdot, -u) \otimes \kappa_2(\cdot, -v) &= \{-\kappa_1(\cdot, u)\} \otimes \{-\kappa_2(\cdot, v)\} \\ &= \kappa_1(\cdot, u) \otimes \kappa_2(\cdot, v).\end{aligned}$$

Consequently, the reduced variable $\langle f, Ug \rangle$ in (1) is almost surely uniquely defined, regardless of which particular singular value decomposition of X we use. Note that this would not be the case without the second part of Assumption 2 as then more freedom would be allowed in choosing the singular vectors corresponding to singular values with multiplicity greater than one.

Finally, we note that if both κ_1 and κ_2 are taken to be linear kernels, then the operator U is equal to the original matrix X, meaning that in this case uniqueness is achieved for U even without the second part of Assumption 2. Essentially, this comes down to the fact that a rotation of a left singular space of a matrix can always be cancelled by applying the inverse of this rotation to the corresponding right singular space, and using linear kernels transfers this property from X to U. Hence, we conclude that the requirement of almost surely distinct non-zero singular values in Assumption 2 is specific to the non-linear case.

IV. MNPCA

Recall that (2D)^{2}PCA [1] is a method of linear dimension reduction that can be seen as an extension of principal component analysis to matrices (when $p_2 = 1$ it is equivalent to the standard PCA). In (2D)^{2}PCA, the left projection directions $a_1, \ldots, a_{d_1}$ are found as the first d_1 eigenvectors of the matrix $\mathrm{E}[\{X - \mathrm{E}(X)\}\{X - \mathrm{E}(X)\}']$, whereas their right-hand side counterparts $b_1, \ldots, b_{d_2}$ are analogously taken to be the first d_2 eigenvectors of the matrix $\mathrm{E}[\{X - \mathrm{E}(X)\}'\{X - \mathrm{E}(X)\}]$. Given the projection directions, the reduced matrix Z containing the $d_1 d_2$ combined projections is formed as $Z := (a_k'\{X - \mathrm{E}(X)\}b_\ell)_{k=1\,\ell=1}^{d_1\quad d_2}$. If multiple eigenvalues are encountered, the corresponding eigenvectors and projections are not uniquely defined.

Prior to defining MNPCA, our non-linear analogue of (2D)^{2}PCA, we first construct the first and second moments of the operator U defined in Section III. For this, we make a weak assumption about the kernels κ_1, κ_2 that will simplify the presentation to come. We note that this assumption is made simply for convenience and that our theory would function perfectly well even without it, assuming that the moment assumptions in Sections IV and V are suitably adjusted.

Assumption 3: There exist constants $C_1, C_2 > 0$ such that, for all $u \in \mathbb{S}^{p_1 - 1}, v \in \mathbb{S}^{p_2 - 1}$,

$$\kappa_1(u, u) < C_1 \quad \text{and} \quad \kappa_2(v, v) < C_2.$$

Assumption 3 is satisfied, e.g., for even and odd kernels induced (in the sense of Lemma 1) by all Gaussian, Laplace and polynomial kernels.

Recall then that we defined the random operator U in Section III as

$$U = \sum_{j=1}^{r} \sigma_j \{\kappa_1(\cdot, u_j) \otimes \kappa_2(\cdot, v_j)\},$$

where (u_j, v_j) is a jth singular pair of the almost surely rank-r random matrix X and σ_j denotes the corresponding singular value. To construct moments for U, we define the expected value of an arbitrary random operator Y taking values in $\mathcal{S}(\mathcal{H}_i, \mathcal{H}_j)$ in the usual way, i.e., as any Hilbert-Schmidt operator $\mathrm{E}(Y) \in \mathcal{S}(\mathcal{H}_i, \mathcal{H}_j)$ satisfying

$$\langle A, \mathrm{E}(Y) \rangle_{\mathrm{HS}} = \mathrm{E}\langle A, Y \rangle_{\mathrm{HS}},$$

for all $A \in \mathcal{S}(\mathcal{H}_i, \mathcal{H}_j)$. By the Riesz representation theorem [22], the expectation $\mathrm{E}(Y)$ exists and is unique as soon as $\mathrm{E}\|Y\|_{\mathrm{HS}} < \infty$. Defined like this, the expectation of a random operator is straightforwardly verified to satisfy the following intuitive properties (where we implicitly assume that the relevant expectations exist as unique): (i) The expectation is linear in the sense that $\mathrm{E}(a_1 Y_1 + a_2 Y_2) = a_1 \mathrm{E}(Y_1) + a_2 \mathrm{E}(Y_2)$ for all scalars $a_1, a_2 \in \mathbb{R}$ and all random operators Y_1, Y_2. (ii) For any random operator Y, we have $\mathrm{E}(Y^*) = \{\mathrm{E}(Y)\}^*$. In particular, if the operator Y is self-adjoint, then so is $E(Y)$. (iii) For any fixed operator A and any random operator Y, we have $\mathrm{E}(AY) = A\mathrm{E}(Y)$.

Under Assumption 3, the Hilbert-Schmidt norm of U has a particularly simple upper bound:

$$\begin{aligned}\|U\|_{\mathrm{HS}} &\le \sum_{j=1}^{r} \sigma_j \|\kappa_1(\cdot, u_j) \otimes \kappa_2(\cdot, v_j)\|_{\mathrm{HS}} \\ &\le (C_1 C_2)^{1/2} r \|X\|_2, \end{aligned} \tag{2}$$

where $\|X\|_2 = \sigma_1$ is the spectral norm of the random matrix X. Consequently, the expectation $\mathrm{E}(U)$ is well-defined and unique as soon as $\mathrm{E}\|X\|_2 < \infty$. However, we instead make the following, stricter assumption that is needed when we next construct the second moment of U.

Assumption 4: We have $\mathrm{E}\|X\|_2^2 < \infty$.

Assumption 4, along with its fourth moment counterpart appearing later as Assumption 5, essentially requires that the random matrix X is not too heavy-tailed. This means that our theoretical results for MNPCA cannot be guaranteed to hold in scenarios typically exhibiting this kind of behavior, such as with financial data. However, we note that the same is true also for any method based on covariances (second moments), such as the classical PCA or (2D)^{2}PCA. From a theoretical perspective, Assumption 4 is needed to guarantee that the covariance operator H_1 defined below in (3) exists.

By the sub-multiplicativity of the Hilbert-Schmidt norm, $\|UU^*\|_{\mathrm{HS}} \le \|U\|_{\mathrm{HS}}^2$, showing that Assumption 4 indeed guarantees that $\mathrm{E}(UU^*)$ is well-defined. Having constructed $\mathrm{E}(U)$ and $\mathrm{E}(UU^*)$, we take the operator analogy of the matrix

$\mathrm{E}[\{X-\mathrm{E}(X)\}\{X-\mathrm{E}(X)\}']$ to be the Hilbert-Schmidt operator $H_1 \equiv H_1(X) \in \mathcal{S}(\mathcal{H}_1, \mathcal{H}_1)$ defined as

$$\begin{aligned} H_1 &:= \mathrm{E}[\{U-\mathrm{E}(U)\}\{U-\mathrm{E}(U)\}^*] \\ &= \mathrm{E}(UU^*) - \mathrm{E}(U)\mathrm{E}(U)^*. \end{aligned} \tag{3}$$

In (2D)^{2}PCA, the left projection directions are obtained as the eigenvectors of the matrix $\mathrm{E}[\{X-\mathrm{E}(X)\}\{X-\mathrm{E}(X)\}']$. We next show that the equivalent is well-defined in the non-linear case. Denoting $Y := U - \mathrm{E}(U)$, as YY^* is self-adjoint, so is the operator H_1. Moreover, like its linear counterpart, H_1 is also positive semi-definite, as is seen by writing, for an arbitrary $f \in \mathcal{H}_1$,

$$\begin{aligned} \langle f, H_1 f\rangle &= \langle (f\otimes f), H_1\rangle_{\mathrm{HS}} \\ &= \mathrm{E}\langle (f\otimes f), YY^*\rangle_{\mathrm{HS}} \\ &= \mathrm{E}\|Y^* f\|^2. \end{aligned}$$

As Hilbert-Schmidt operators are compact [22, p.267], H_1 admits the spectral decomposition $H_1 = \sum_{k=1}^{\infty} \lambda_k (a_k \otimes a_k)$ where $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ and $\{a_k\}$ is an orthonormal basis of $\mathcal{H}_1$ [23, Theorem 4.10.4]. We can similarly obtain the orthonormal basis $\{b_k\}$ of $\mathcal{H}_2$ corresponding to the operator $H_2 := \mathrm{E}[\{U-\mathrm{E}(U)\}^*\{U-\mathrm{E}(U)\}]$.

We next define MNPCA using the previously defined operators. We assume, for now, that the reduced ranks d_1, d_2 are known and postpone the discussion of their estimation later to Section VII on tuning parameter selection. Let then $a_1, \ldots, a_{d_1}$ and $b_1, \ldots, b_{d_2}$ be any first d_1 and d_2 eigenvectors of the self-adjoint positive semi-definite operators H_1 and H_2, respectively. The $d_1 \times d_2$ matrix Z of MNPCA-components of X is now found element-wise as

$$z_{k\ell} := \langle a_k, \{U-\mathrm{E}(U)\} b_\ell\rangle. \tag{4}$$

Each row of the matrix Z shares the same non-linear row element a_k (and analogously for the columns of Z), implying that it is meaningful to view Z as a matrix, instead of simply as a collection of latent variables. Moreover, as lower indices k correspond to larger eigenvalues and greater amount of information, we expect the most interesting part of the MNPCA-matrix Z to be its top left corner.

We next provide an interpretation for the first left-hand side function a_1 found by MNPCA. As discussed earlier, a_1 is found by maximizing the quadratic form $f \mapsto \langle f, H_1 f\rangle$. We now show that this quadratic form admits a simplified form in the special case where the right-hand side kernel κ_2 satisfies the following: for any orthonormal set of vectors $v_1, \ldots, v_r \in \mathbb{R}^{p_2}$, the matrix $R := \{\kappa_2(v_j, v_k)\}_{j,k=1}^r$ is a constant not depending on $v_1, \ldots, v_r$. It is straightforwardly checked that, e.g., both the even and odd kernels induced by Gaussian kernels satisfy this condition.

Lemma 2: Let κ_2 be as discussed above. Then,

$$\langle f, H_1 f\rangle = (b_1 - b_2)\mathrm{tr}(\Sigma) + b_2 1_r' \Sigma 1_r,$$

where $1_r \in \mathbb{R}^r$ is a vector of ones, $\Sigma := \mathrm{Cov}(Z)$, $Z := (\sigma_1 f(u_1), \ldots, \sigma_r f(u_r)) \in \mathbb{R}^r$, $b_1 := \kappa_2(e_1, e_1)$, $b_2 := \kappa_2(e_1, e_2)$ and e_j denotes the jth standard basis vector of $\mathbb{R}^{p_2}$.

Note that, due to κ_2 being a kernel function, we always have $b_1 \geq b_2$. Lemma 2 allows us to make two interpretations: (i) As both $\mathrm{tr}(\Sigma)$ and $1_r'\Sigma 1_r$ measure the size of the covariance matrix Σ, the result essentially implies that, to maximize the quadratic form, we must choose f such that the random vector $Z = (\sigma_1 f(u_1), \ldots, \sigma_r f(u_r))$ *exhibits as much variation as possible*. The elements of Z are scaled with the singular values $\sigma_1 > \cdots > \sigma_r$, meaning that if σ_1 dominates the other singular values, f is essentially chosen to maximize the variance of $f(u_1)$. In the case of more balanced singular values, also the later singular spaces play a role in choosing f. (ii) On a finer scale, the weighting between $\mathrm{tr}(\Sigma)$ and $1_r'\Sigma 1_r$ is determined by the non-rigidity of the kernel function. If $b_2 \approx 0$, then κ_2 maps orthogonal vectors into almost orthogonal features and the value of $\langle f, H_1 f\rangle$ is determined almost fully by its first term, the variances of the elements of Z. Whereas, if b_2 differs significantly from zero, then κ_2 turns orthogonal vectors into non-orthogonal features and also the off-diagonal elements of Σ affect the choice of f.

We close this section by pointing out that, while MNPCA reduces to (2D)^{2}PCA under a linear kernel, interestingly it does *not* reduce to KPCA when X is a vector, $p_2 = 1$ but, rather, to a weighted version of KPCA. To see this, decompose the observed random vector as $X = \|X\| V$, where $V = X/\|X\|$. Then, in KPCA with the kernel κ the principal components are determined by the covariance operator $\mathrm{E}\{\kappa(\cdot, X)\otimes\kappa(\cdot, X)\} - \mathrm{E}\{\kappa(\cdot, X)\}\otimes\mathrm{E}\{\kappa(\cdot, X)\}$. Whereas, in MNPCA with the kernels κ, κ_2, the operator representation of X equals $U = \|X\|\{\kappa(\cdot, V)\otimes\kappa_2(\cdot, 1)\}$ and the operator H_1 is thus,

$$\begin{aligned} H_1 = \mathrm{E}\{\|X\|^2 \kappa(\cdot, V)\otimes\kappa(\cdot, V)\} \\ - \mathrm{E}\{\|X\|\kappa(\cdot, V)\}\otimes\mathrm{E}\{\|X\|\kappa(\cdot, V)\}. \end{aligned}$$

This shows that MNPCA with $p_2 = 1$ is equivalent to running KPCA for the normalized random vector $X/\|X\|$ and weighting the kernel with the magnitude $\|X\|$. Thus, in particular, if the random vector resides in the unit sphere, $\|X\| = 1$, the two methods are the same. We also see that the two methods are equal if κ is linear, and this relationship is equivalent to the well-known fact that KPCA with linear kernel equals PCA. We leave the study of the above novel weighted KPCA for future work.

V. Sample Consistency

We next turn our attention to the sample version of MNPCA and its asymptotic properties. Without loss of generality, we restrict our discussion to the left-hand side of the model, the equivalent results for the right-hand side following instantly by symmetry.

Let $X_1, \ldots, X_n$ be a sample of $p_1 \times p_2$ matrices from the distribution of X and let (u_{ij}, v_{ij}) and σ_{ij} denote the jth singular pair of the ith observed matrix and the corresponding singular value, respectively. For each $i = 1, \ldots, n$, we let $U_i \in \mathcal{B}(\mathcal{H}_2, \mathcal{H}_1)$ denote the linear operator

$$U_i := \sum_{j=1}^{r} \sigma_j \{\kappa_1(\cdot, u_j)\otimes\kappa_2(\cdot, v_j)\}.$$

Under Assumption 2 and for odd/even kernels κ_1, κ_2, the operators $U_1, \ldots, U_n$ are almost surely unique and a computation similar to (2) reveals that they are Hilbert-Schmidt operators. By defining the "average" operator as $\bar{U}_n := (1/n)\sum_{i=1}^n U_i$, the sample version $H_{n1} \in \mathcal{S}(\mathcal{H}_1, \mathcal{H}_1)$ of the operator H_1 is defined as,

$$H_{n1} := \frac{1}{n}\sum_{i=1}^{n}(U_i - \bar{U}_n)(U_i - \bar{U}_n)^*.$$

We remind here that the role of the subscript n in H_{n1} is to denote that the operator is a sample-level object. Similarly, the subscript 1 means that the H_{n1} is an operator relating to the left-hand side of the model. As our first asymptotic result, we show that H_{n1} converges to H_1 in the Hilbert-Schmidt norm at the standard root-n rate, as soon as the fourth moment of X is bounded. In classical statistics, finite moments of order $2k$ are required to obtain the convergence of the sample kth moment via the central limit theorem at the root-n rate [24]. In this sense, the need for Assumption 5 below is perfectly reasonable in the current scenario where we are studying the convergence of a quantity based on second moments.

Assumption 5: We have $\mathrm{E}\|X\|_2^4 < \infty$.

Theorem 1: Under Assumptions 1, 2, 3 and 5, we have, as $n \to \infty$,

$$\sqrt{n}\|H_{n1} - H_1\|_{\mathrm{HS}} = \mathcal{O}_p(1).$$

The notation "$Y_n = \mathcal{O}_p(1)$" in Theorem 1 means that the sequence Y_n of random variables is stochastically bounded [24]. Under suitable regularity conditions, the convergence of H_{n1} in Theorem 1 guarantees that also the corresponding eigenspaces are consistent. A standard assumption for this in the kernel dimension reduction literature is that the operator in question has finite rank and its positive eigenvalues are distinct [25], [26], [27]. A finite rank essentially ensures that a large enough sample can be used to capture the full information content of the population distribution, whereas having distinct eigenvalues makes certain that the individual latent variables can be identified. Thus, Assumption 6 can be seen as being very practical, and if it was omitted, conducting dimension reduction in practice would partially lose its meaning. We also note that Assumption 6 is required only for Corollary 1 below, no other result in this work requires it. The proof of the corollary is omitted as it follows directly from [25, Theorem 2].

Assumption 6: The operator H_1 has finite rank d_1 and its positive eigenvalues are distinct.

Corollary 1: Let Assumptions 1, 2, 3, 5 and 6 hold. Denote by a_k and a_{nk}, $k = 1, \ldots, d_1$, any kth eigenvectors of H_1 and H_{n1}, respectively, with their signs chosen such that $\langle a_k, a_{nk}\rangle \geq 0$. Then, as $n \to \infty$,

$$\sqrt{n}\|a_{nk} - a_k\| = \mathcal{O}_p(1).$$

The proof of Theorem 1 reveals that $\mathrm{E}(U)$ can be estimated root-n-consistently by the operator $\bar{U}_n$, guaranteeing together with Corollary 1 that the sample MNPCA-components

$$z_{i,k\ell} := \langle a_{nk}, (U_i - \bar{U}_n)b_{n\ell}\rangle, \quad i = 1, \ldots, n,$$

are themselves a good approximation to their population counterparts in (4). As with their population version, the sample MNPCA-components are naturally collected into the matrices $Z_1, \ldots, Z_n$ of size $d_1 \times d_2$.

We close this section by noting that the $\sqrt{n}$-rate of convergence is very standard in the context of principal component analysis and, essentially, any moment-based statistical inference on i.i.d. data. Namely, it is satisfied by PCA [28], (2D)^{2}PCA [29], kernel PCA [25], and various methods for matrix-valued random variables, such as [18], [19].

VI. Coordinate Representation

The results of the preceding section were stated on the operator level, and, in order to apply MNPCA in practice, we next develop a finite-dimensional representation of the method. For this, assume that we are given a sample $X_1, \ldots, X_n$ of $p_1 \times p_2$ matrices from the distribution of X. As before, we let (u_{ij}, v_{ij}) and σ_{ij} denote the jth singular pair of the ith observed matrix and the corresponding singular value, respectively.

In standard kernel methodology for vector-valued data, it is typical to take the sample counterpart of the RKHS induced by the kernel κ to be the linear span of the representatives $\kappa(\cdot, x_i)$ of the observed sample of vectors $x_1, \ldots, x_n$. In our case of matrix data, the natural counterpart to this procedure is to use the representatives of the singular vectors instead. A key question is then how many singular vectors from each X_i should be used. This choice has a direct impact on the computational complexity of MNPCA as using, say, m singular vectors from each observation yields a kernel matrix of the size $mn \times mn$, leading to increased computational burden for larger m. On the other hand, a larger m also guarantees a richer function space, making the choice a trade-off (and m essentially a tuning parameter). For notational simplicity, we have in the following restricted ourselves to using only the first singular vectors, $m = 1$, but the formulas could easily be adapted for other m as well. We thus define the sample counterpart of the RKHS $\mathcal{H}_1$ as,

$$\mathcal{H}_{n1} := \mathrm{span}(\mathcal{B}_{n1}),$$

where the spanning set $\mathcal{B}_{n1} := \{\kappa_1(\cdot, u_{i1}) \mid i = 1, \ldots, n\}$ is, for notational convenience, taken to satisfy the following condition, which implies, in particular, that $\mathcal{B}_{n1}$ forms a basis for $\mathcal{H}_{n1}$ and that $\dim(\mathcal{H}_{n1}) = n$.

Assumption 7: The elements of $\mathcal{B}_{n1}$ are linearly independent.

For an arbitrary member f of $\mathcal{H}_{n1}$, we define its coordinate $[f] \in \mathbb{R}^n$ to be the vector of its coefficients in the basis $\mathcal{B}_{n1}$. Thus, letting $k_1 : \mathbb{R}^p \to \mathbb{R}^n$ denote the function acting as $k_1(x) = (\kappa_1(x, u_{11}), \ldots, \kappa_1(x, u_{n1}))'$, we have

$$f(x) = [f]'k_1(x), \tag{5}$$

for every $x \in \mathbb{R}^{p_1}$, $f \in \mathcal{H}_{n1}$. Let $K_1 \in \mathbb{R}^{n\times n}$ denote the kernel matrix whose (i, j)-element is $\langle \kappa_1(\cdot, u_{i1}), \kappa_1(\cdot, u_{j1})\rangle = \kappa_1(u_{i1}, u_{j1})$. To turn $\mathcal{H}_{n1}$ into a Hilbert space, we equip $\mathcal{H}_{n1}$ with the inner product

$$\langle f, g\rangle_n := [f]'K_1[g], \tag{6}$$

whose positive-definiteness is guaranteed by Assumption 7. Note, however, that the resulting space is, strictly speaking, not an RKHS as we take the domain of its elements to be the full space $\mathbb{R}^{p_1}$ instead of the set $\{u_{11},\dots,u_{n1}\}$. This means, in particular, that the reproducing property $f(x)=\langle f,\kappa_1(\cdot,x)\rangle_n$ holds only when $x\in\{u_{11},\dots,u_{n1}\}$ (however, for x not satisfying this, we still have the relation (5)). Finally, we construct the right-hand side counterparts $\mathcal{H}_{n2},K_2$ similarly, under the analogous assumptions.

Under the above specifications, the sample algorithm for MNPCA (as described in Section IV) is given in the next theorem.

Theorem 2: Denote $F_i:=\sum_{j=1}^{r}\sigma_{ij}k_1(u_{ij})k_2(v_{ij})'$, $i=1,\dots,n$, and let $a_1,\dots,a_{d_1}$ and $b_1,\dots,b_{d_2}$ be any first d_1 and d_2 eigenvectors of the $n\times n$ matrices

$$P_1:=K_1^{-1/2}\left(\frac{1}{n}\sum_{i=1}^{n}F_iK_2^{-1}F_i'-\bar{F}K_2^{-1}\bar{F}'\right)K_1^{-1/2},\quad(7)$$

and

$$P_2:=K_2^{-1/2}\left(\frac{1}{n}\sum_{i=1}^{n}F_i'K_1^{-1}F_i-\bar{F}'K_1^{-1}\bar{F}\right)K_2^{-1/2},$$

respectively, where $\bar{F}:=(1/n)\sum_{i=1}^{n}F_i$. Then, the $d_1\times d_2$ matrix Z_i of the MNPCA-components of the ith observation is given by

$$z_{i,jk}:=a_j'K_1^{-1/2}(F_i-\bar{F})K_2^{-1/2}b_k.\quad(8)$$

Two notes are in order. First, the proof of Theorem 2 reveals that to obtain the MNPCA-component matrix of an out-of-sample observation X_0, it is sufficient to replace F_i in (8) with the equivalent matrix having the singular vectors/values of X_0 in place of those of X_i. Secondly, the presence of the inverses K_1^{-1},K_2^{-1} might make the procedure numerically unstable in practice, and in our later examples we have replaced them with the corresponding regularized inverses $K^{\dagger}:=(K+\varepsilon\|K\|_2)^{-1}$ where $\varepsilon=0.2$. Similarly, one might want to truncate the decompositions F_i after some small number of singular values, say, two or three (effectively assuming that the rank in Assumption 2 is $r=2$ or $r=3$).

The sample MNPCA-procedure described in Theorem 2 can be seen as a true non-linear generalization of (2D)^{2}PCA in the sense that it reverts back to the standard (2D)^{2}PCA when a linear kernel is used, as long as the sets of leading singular vectors $u_{11},\dots,u_{n1}$ and $v_{11},\dots,v_{n1}$ span the full spaces $\mathbb{R}^{p_1}$ and $\mathbb{R}^{p_2}$, respectively. This result, formalized in Theorem 3 below, is proven in the appendix. Note also that all linear kernels are odd, satisfying our requirement of odd/even kernels.

Theorem 3: Let κ_1,κ_2 be linear kernels and assume that $n\geq\max\{p_1,p_2\}$ and that the matrices $U:=(u_{11},\dots,u_{n1})'$, $V:=(v_{11},\dots,v_{n1})'$ have full rank. Then, treating the inverses in Theorem 2 as Moore-Penrose generalized inverses, the MNPCA-components of the ith observation are,

$$Z_i:=A'(X_i-\bar{X})B,$$

where A,B contain, respectively, any first d_1 and d_2 eigenvectors of the matrices

$$\frac{1}{n}\sum_{i=1}^{n}(X_i-\bar{X})(X_i-\bar{X})'\quad\text{and}\quad\frac{1}{n}\sum_{i=1}^{n}(X_i-\bar{X})'(X_i-\bar{X}).$$

TABLE I
THE COMPUTATIONAL COMPLEXITIES OF DIFFERENT METHODS UNDER THE SIMPLIFIED SCENARIO WHERE $p_1=p_2=:p$ AND THE NUMBER OF LATENT DIMENSIONS IS NEGLIGIBLE COMPARED TO n,p

MNPCA	PCA	KPCA	K2DPCA	(2D)^{2}PCA
n^4	$\min\{np^4,n^2p^2\}$	n^2	n^2p^2	np^3

We next compare the computational complexity of MNPCA to those of standard PCA, kernel PCA (KPCA), K2DPCA (as proposed in [4]) and (2D)^{2}PCA. For simplicity, we focus only on the required number of matrix multiplication and eigendecomposition operations and assume that the latent dimensions d_1,d_2,d are negligible in size compared to the parameters n,p_1,p_2.

For MNPCA, as detailed in Theorem 2, the computation of the matrix (7) and the extraction of the full set of latent variables (8) both have $\mathcal{O}(n^4)$ complexity. These are the most expensive operations involved, meaning that the complexity of the full procedure is $\mathcal{O}(n^4)$. Recall that we assumed earlier that only the first singular space is used to estimate $\mathcal{H}_1$ and $\mathcal{H}_2$, i.e., $m=1$. In the case of general m, it is simple to check that the complexity of MNPCA is $\mathcal{O}(m^3n^4)$, verifying that m indeed has a major impact on the complexity.

For PCA and KPCA, we assume that the sample of matrices has been vectorized into a sample of p_1p_2-dimensional vectors. Now, the complexity of estimating the d first principal components of an arbitrary $n\times p$ dataset Y is either $\mathcal{O}(np^2)$ or $\mathcal{O}(n^2p)$, depending on whether we decompose the matrix $Y'Y$ or YY', respectively. Hence, the complexity of PCA in the current problem is $\mathcal{O}(\min\{np_1^2p_2^2,n^2p_1p_2\})$. Finally, for KPCA, the only operation needed is that of extracting the d first eigenvalue-eigenvector pairs of the kernel matrix, which has the complexity $\mathcal{O}(n^2)$. For K2DPCA, based on the previous paragraph and Theorem 3 in [4], the complexity of the full procedure is $\mathcal{O}(n^2p_1^2)$.

In (2D)^{2}PCA, computing $(1/n)\sum_{i=1}^{n}X_iX_i'-\bar{X}\bar{X}'$ requires $n+1$ multiplications of $p_1\times p_2$ and $p_2\times p_1$ matrices, giving the complexity $\mathcal{O}(np_1^2p_2)$. The extraction of the first d_1 eigenvector-eigenvalue pairs of a $p_1\times p_1$ matrix is an $\mathcal{O}(p_1^2)$-operation. Finally, taking into account also the right-hand side of the model, the computation of the latent matrices in (2D)^{2}PCA has the total complexity of $\mathcal{O}(np_1^2p_2+np_1p_2^2)$.

The computational complexities of the five methods under a simplified scenario where $p_1=p_2=:p$ are summarized in Table I. Based on the table, the ranking of the methods depends crucially on the dimensions of the data. In the extreme scenario where $n\gg p$, the fastest method is (2D)^{2}PCA which avoids both the vectorization and the need to operate on $n\times n$ kernel matrices. Whereas, if $p\gg n$, then KPCA is the fastest (but fails to acknowledge the matrix structure of the data), with

MNPCA coming in second. Finally, we note that the previous computations ignore the complexity involved in computing the kernel functions κ_1, κ_2.

VII. Tuning Parameter Selection

The tuning parameters of MNPCA include the number m of singular spaces used to approximate the spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, the rank r involved in computing the matrices F_i in Theorem 2, the latent dimensionalities (d_1, d_2) and any additional tuning parameters involved with the kernels κ_1, κ_2. We next give suggestions on how to choose these in practice.

The number m of singular spaces differs from the other tuning parameters in the sense that increasing m is always better from the viewpoint of estimation accuracy (barring any possible numerical instability), enabling better coverage of the RKHS. However, as discussed in the previous section, the computational burden of MNPCA increases in the third power of m and, hence, our suggestion is to choose as large value of m as is possible within the given computational limits.

The rank r can be seen to control a bias-variance trade-off on the level of individual observations. That is, too small values of r risk discarding some defining features of the observations X_i whereas large r might bring with it noisy singular spaces, distracting from efficient estimation. In an exploratory context, we suggest experimenting with several small values of r, say 1–5, whereas, when using MNPCA as a preprocessing step for another method with measurable performance, cross-validation can also be used.

As in most forms of PCA, the selection of d_1 (and, equivalently d_2) can be based on a scree plot of the eigenvalues of the matrix P_1 (P_2) in Theorem 2. The large dimensionality n of the matrix means that standard cut-offs such as 80% explained variance might not be useful due to the long tail of small but non-zero noise eigenvalues. Instead, we suggest using the following heuristic: retain all principal components whose eigenvalues λ_i exceed $\bar{\lambda} + 2 \cdot \mathrm{sd}(\lambda_i)$, separately for the two sides of the model.

Finally, to choose the tuning parameters of the kernels κ_1, κ_2, an obvious choice is to use cross-validation. Alternatively, in absence of any performance criterion, we suggest using a suitable default value. For example, for the even/odd kernel induced by the Gaussian kernel $(x, y) \mapsto \exp\{\|x - y\|^2/(2\sigma^2)\}$, a natural choice is to use $\sigma^2 = \|G\|/n$ where the matrix G has the inner product $u'_{i1}u_{j1}$ as its (i, j)th element. This choice makes σ^2 comparable in magnitude to the average value of $\|x - y\|^2$ and is additionally invariant to any sign-changes to individual singular vectors u_{i1}. This value will be used as a "baseline" in our later data examples.

VIII. Data Examples

The R-codes for running MNPCA are available on the web page of one of the authors, https://users.utu.fi/jomivi/software.

A. Simulation

We next evaluate the performance of MNPCA using simulated image data. As competitors, we take $(2D)^2$PCA (a linear baseline) and its non-linear extension K2DPCA, as proposed in [4]. Given $x \in [-\pi, \pi]$ and $\alpha \in (-1, 1)$, we let $u(x; \alpha)$ denote the 10-dimensional vector whose jth element equals $\cos\{(1 - \alpha)(x - \pi + \frac{j-1}{10}2\pi)\}$. We then fix $\alpha \in (-1, 1)$ and generate images representing two groups as follows: for images from Group 1, we first randomly generate $\theta_1, \theta_2, \theta_3, \theta_4$ i.i.d. from $\mathrm{Unif}(-\pi, \pi)$. A 10×10 image is then constructed as $u(\theta_1; \alpha)u(\theta_2; \alpha)' + u(\theta_3; \alpha)u(\theta_4; \alpha)'$. An image from Group 2 is generated with identical steps but by using $-\alpha$ in place of α, meaning that the parameter α controls the distance between the two groups; larger α corresponds to better separated groups. In this study, we consider a total of three values $\alpha = 0.125, 0.100, 0.075$. Samples of images from the two groups generated with $\alpha = 0.125$ are shown in Fig. 1. The two groups (top two vs. bottom two rows in Fig. 1) are not that easy to discern visually but, as we will later see, the two groups are actually perfectly separable after a non-linear mapping.

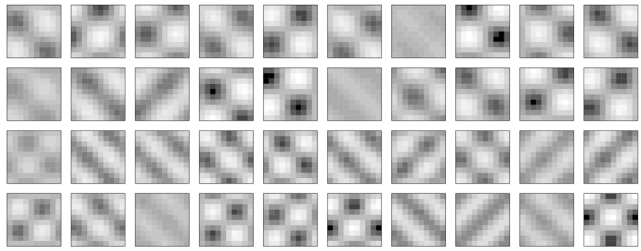

Fig. 1. A sample of 20 images from Group 1 (top two rows) and 20 images from Group 2 (bottom two rows). The members of Group 2 can be seen to exhibit a denser checkerboard pattern compared to Group 1.

We consider two different sample sizes $n = 50, 100$, generating in both cases 50% of the observations from each group. In every replicate of the simulation, we use each of the methods to estimate a total of 4 components (2×2 latent matrices), fit a QDA classifier to them and, finally, use the trained classifier to predict the classes of a separate test image set of size $n_0 = 50$. We use the Gaussian kernel for all non-linear methods and distinguish two versions of MNPCA, even and odd, giving us a total of four methods to compare. For simplicity and to alleviate computational burden we used $m = 1$ singular spaces to estimate the RKHS and the truncated rank $r = 2$.

The resulting average classification accuracies in the test set over 500 replicates of the simulation are shown in Fig. 2. The horizontal axes of the panels correspond to the value of the tuning parameter σ^2 and are relative in the sense that the tick mark a denotes the value $\sigma^2 = 2^a\sigma_0^2$ where σ_0^2 is a "default" value estimated from the data (once). For MNPCA (the lines denoted by "Even" and "Odd"), we used the default value proposed in Section VII and for K2DPCA we use the value $\sigma_0^2 = (1/n^2)\sum_{i,j=1}^n \|x_i - x_j\|^2$ where $x_1, \ldots, x_n$ are the vectorized images. As the parameter σ^2 takes the fixed values specified earlier, no actual tuning is involved in running any of the methods. Note that $(2D)^2$PCA does not use tuning parameters, meaning that its line in Fig. 2 is perfectly horizontal.

We make the following observations from Fig. 2: (i) $(2D)^2$PCA, while improving over a random guess prediction, fails to reach a satisfactory level of accuracy even in the

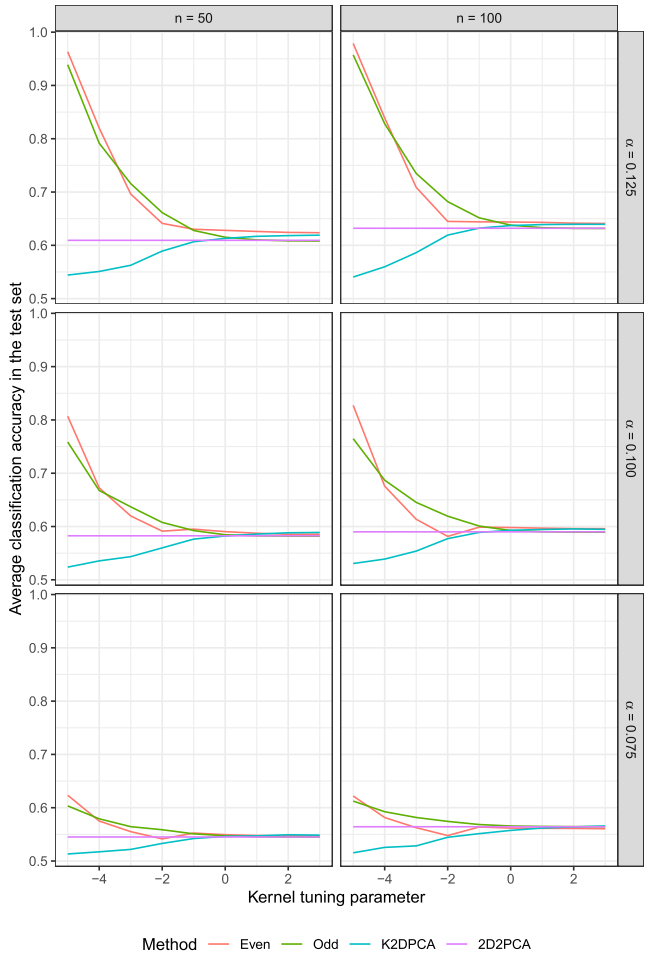

Fig. 2. The results of the simulation study. Each line traces the average classification accuracy of the corresponding method as a function of the tuning parameter σ^2. The panels correspond to the choice of sample size (columns) and the α-parameter controlling the difficulty of the separation task (smaller α equals more difficult task).

easiest scenario with $\alpha = 0.125$. This is because the separating boundary between the two groups is highly non-linear and, in particular, no single pixel (or even the average behavior of a row or column) can be used to identify whether an image belongs to Group 1 or 2 because of the oscillating mechanism we used to generate the data. (ii) With larger values of σ^2, K2DPCA manages to improve over (2D)^{2}PCA. but only by a very small margin. In fact, with an improper choice of the tuning parameter, the accuracy of K2DPCA drops well below that of (2D)^{2}PCA. (iii) Both the even and odd version of MNPCA manage to find a non-linear mapping (corresponding to small values of σ^2) that perfectly separates the groups, yielding significantly improved performance over the other two methods. Actually, even though the choice of the tuning parameter σ^2 greatly affects its performance, MNPCA is still, even for sub-optimally selected σ^2, roughly as efficient as the competitors at their best. We also note that there is very little difference between the choice of odd or even Gaussian kernel.

Finally, we briefly remark on the validity of Assumptions 2, 4, 5 and 6 in this example (Assumptions 1 and 3 are automatically satisfied by our choice of kernels). The data is generated from a continuous distribution so Assumption 2 is satisfied (up to machine precision). The data take values in bounded interval, implying that Assumptions 4 and 5 hold. For Assumption 6, as the operator H_1 is not available to us, we rather inspected the eigengaps of the symmetric matrix $K_1^{1/2}[H_{n1}]K_1^{-1/2}$, whose eigenvalues match those of the coordinate form $[H_{n1}]$ of the sample estimator of H_1. Assumption 6 can be seen to hold if no non-zero eigengap follows a zero eigengap. We inspected the distributions (not shown here) of these eigengaps over several replicates of the data for different parameter settings and came to the conclusion that there is no reason to doubt the validity of Assumption 6 either in the current scenario.

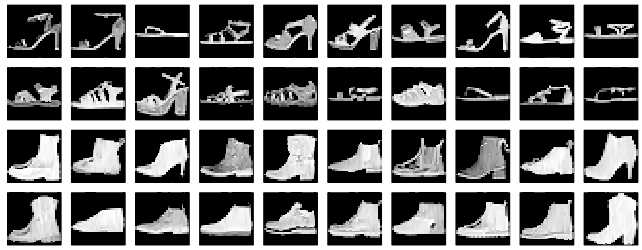

Fig. 3. A sample of 20 images from class 5 (sandals, top two rows) and 20 images from class 9 (ankle boots, bottom two rows) in the FashionMNIST data set.

B. Real Data Example

We next apply MNPCA to the FashionMNIST data, containing gray-scale 28×28 images of clothing objects and available at Kaggle[1]. We consider only the 10000 images designated as a "test set" and restrict our attention there to the 2000 images of classes 5 and 9, sandals and ankle boots. Fig. 3 illustrates a selection of 40 random images from these two classes. Our objective is the same as in the simulation study, to extract a small amount of components from a training data and use these to fit a QDA classifier for predicting the labels in a separate test data set. We consider two sample sizes $n = 50, 100$ for the training data, taking $n_0 = 50$ test images in both cases. We perform a total of 100 repetitions of the study for both sample sizes, always drawing the training and test sets randomly from the full data set of 2000 images. For the tuning parameters we use the same specifications as in the simulation study.

Due to clear visual differences between the two groups in Fig. 3, the separating boundary is likely to be, if not linear, then at least closely approximable by a linear direction. As such, the differences between the four methods are not expected to be as drastic as in the earlier simulation. Indeed, the results illustrated in Fig. 4 reveal that this is what happens: (2D)^{2}PCA is not the best method but it comes very close to the others in terms of classification accuracy. For $n = 50$, K2DPCA achieves, by a small margin, the best performance, whereas, when the sample size is doubled, MNPCA surpasses K2DPCA, regardless of the type of the kernel. We also observe that, especially when

[1]https://www.kaggle.com/datasets/zalando-research/fashionmnist

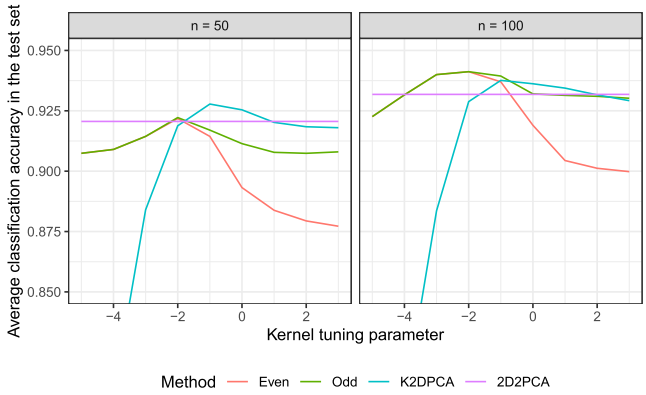

Fig. 4. The results of the FashionMNIST data example. Each line traces the average classification accuracy of the corresponding method as a function of the tuning parameter σ^2. The panels correspond to the choice of sample size.

Fig. 5. The scatter plot of the first two diagonal components of Z_i found by MNPCA with $\sigma^2 = 0.25\sigma_0^2$ in one replicate of the study with $n = 100$. The two groups have been color-coded for visual clarity.

$n = 100$ and using the odd Gaussian kernel, MNPCA can be seen as a very "safe" choice, offering a reliable performance regardless of the value of the tuning parameter σ^2.

To better understand why MNPCA produced improved results over its linear counterpart (2D)^{2}PCA, we studied the latent variables produced by the two methods. Our investigations revealed that the better results of MNPCA are predominantly caused by it finding, in general, tighter clusters than (2D)^{2}PCA. This has been demonstrated in Figs. 5 and 6 which show the scatter plots of the first two diagonal components of Z_i estimated from the training data with MNPCA and (2D)^{2}PCA, respectively, in one instance of the study with $n = 100$. Both scatter plots show good separation of the two groups (colored red and blue for clarity) but it is clear that MNPCA produces a more defined boundary between the groups, explaining why it outdoes (2D)^{2}PCA in Fig. 4. For MNPCA we used, based on Fig. 4, even Gaussian kernel with the tuning parameter value $\sigma^2 = 0.25\sigma_0^2$.

Regarding the validity of our assumptions in this example, we conducted the eigengap experiment described in Section VIII-A also for the FashionMNIST data, with

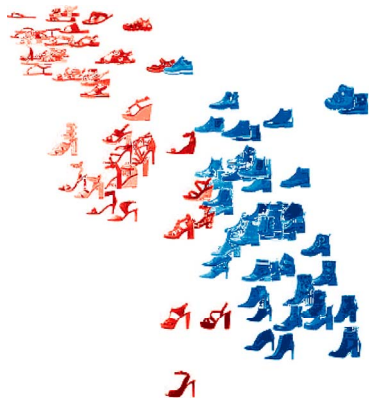

Fig. 6. The scatter plot of the first two diagonal components of Z_i found by (2D)^{2}PCA in one replicate of the study with $n = 100$. The two groups have been color-coded for visual clarity.

analogous results (no evidence to suggest that Assumption 6 would not hold). Assumptions 4 and 5 hold as the data (gray-scale intensities of pixels) reside on a bounded set. And, since the pixel intensities are, while high-resolution, not continuous, we studied the distribution (not shown here) of the first eigengap of X_i which was significantly separated from zero, showing that also Assumption 2 can be seen to hold here. Note that it is sufficient to inspect only the first eigengap since we use the truncated rank $r = 2$ in the experiment.

IX. Discussion

The work proposed here offers multiple opportunities for future study, which we detail next. Firstly, since its proposal in [1], the linear (2D)^{2}PCA-paradigm has later been extended to other settings besides PCA, for example, to supervised dimension reduction [16] and independent component analysis [19]. This naturally begs the question whether the two-sided non-linearization applied in the current work can be extended to these scenarios.

Secondly, in our data examples there was not a major qualitative difference between the odd and even Gaussian kernel. However, this might be context-dependent and it would be interesting to theoretically compare these two classes of kernels. Despite their similarity here, it could be that their behavior differs from one another in some fundamental way.

Thirdly, besides matrix-valued data, also tensor data is currently routinely produced by applications such as medical imaging. As such, extending our non-linearization approach from two-sided to multi-sided would allow for the development of non-linear tensorial dimension reduction methodology. Such an extension is not straightforward as our work here hinges crucially on the use of the singular value decomposition, which is known not to have a direct analogy in the case of tensors [30].

Fourthly, as with any unsupervised method, choosing a proper value for the tuning parameters of the used kernel is not straightforward in MNPCA, for the lack of a criterion for measuring the success of the method. However, in both the simulation study and the real data example MNPCA offered, with both even and odd Gaussian kernels, a very good performance

with the choice $\sigma^2 = 2^{-3}\sigma_0^2$ where σ_0^2 is the value given in Section VII. This empirically observed value could thus be used as a starting point for a more involved study of the tuning of MNPCA.

ACKNOWLEDGMENT

The authors are grateful to the three anonymous reviewers for their comments which helped greatly in improving the presentation and quality of the manuscript.

REFERENCES

[1] D. Zhang and Z.-H. Zhou, "(2D)^{2}PCA: Two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, nos. 1–3, pp. 224–231, 2005.
[2] T. Gao, X. Li, Y. Chai, and Y. Tang, "Deep learning with stock indicators and two-dimensional principal component analysis for closing price prediction system," in *Proc. 7th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Piscataway, NJ, USA: IEEE Press, 2016, pp. 166–169.
[3] J. Yang, D. Zhang, A. F. Frangi, and J.-y. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, Jan. 2004.
[4] H. Kong, L. Wang, E. K. Teoh, X. Li, J.-G. Wang, and R. Venkateswarlu, "Generalized 2D principal component analysis for face image representation and recognition," *Neural Netw.*, vol. 18, nos. 5–6, pp. 585–594, 2005.
[5] V. D. M. Nhat and S. Lee, "Kernel-based 2DPCA for face recognition," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, Piscataway, NJ, USA: IEEE Press, 2007, pp. 35–39.
[6] D. Zhang, S. Chen, and Z.-H. Zhou, "Recognizing face or object from a single image: Linear vs. kernel methods 2D patterns," in *Proc. Joint IAPR Int. Workshops Statist. Techn. Pattern Recognit. (SPR) Structural Syntactic Pattern Recognit. (SSPR)*, Berlin, Heidelberg: Springer, 2006, pp. 889–897.
[7] C. Yu, H. Qing, and L. Zhang, "K2DPCA plus 2DPCA: an efficient approach for appearance based object recognition," in *Proc. 3rd Int. Conf. Bioinf. Biomed. Eng.*, Piscataway, NJ, USA: IEEE Press, 2009, pp. 1–4.
[8] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces," *J. Mach. Learn. Res.*, vol. 5, , pp. 73–99, Jan. 2004.
[9] C. Liu, X. Wei-sheng, and W. Qi-di, "Tensorial kernel principal component analysis for action recognition," *Math. Problems Eng.*, vol. 2013, no. 1, pp. 1–16, 2013.
[10] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2007, pp. 1177–1184.
[11] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious online learning with kernels via sparse projections in function space," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 83–126, 2019.
[12] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[13] A. Berlinet and C. Thomas-Agnan, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. New York, NY, USA: Springer Science & Business Media, 2011.
[14] M. Hein and O. Bousquet, "Kernels, associated structures and generalizations," Tech. Rep. 127, Max Planck Institute for Biological Cybernetics, 2004.
[15] M. Krejnik and A. Tyutin, "Reproducing kernel Hilbert spaces with odd kernels in price prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 10, pp. 1564–1573, Oct. 2012.
[16] B. Li, M. K. Kim, and N. Altman, "On dimension folding of matrix-or array-valued statistical objects," *Ann. Statist.*, pp. 1094–1121, 2010.
[17] Y. Xue and X. Yin, "Sufficient dimension folding for regression mean function," *J. Comput. Graphical Statist.*, vol. 23, no. 4, pp. 1028–1043, 2014.
[18] S. Ding and R. D. Cook, "Tensor sliced inverse regression," *J. Multivariate Analysis*, vol. 133, pp. 216–231, 2015.
[19] J. Virta, B. Li, K. Nordhausen, and H. Oja, "Independent component analysis for tensor-valued data," *J. Multivariate Anal.*, vol. 162, pp. 172–192, Nov. 2017.
[20] J. Virta, B. Li, K. Nordhausen, and H. Oja, "JADE for tensor-valued observations," *J. Comput. Graphical Statist.*, vol. 27, no. 3, pp. 628–637, 2018.
[21] X. Feng and Z. Zhang, "The rank of a random matrix," *Appl. Mathematics Comput.*, vol. 185, no. 1, pp. 689–694, 2007.
[22] J. B. Conway, *A Course in Functional Analysis*, vol. 96. New York, NY, USA: Springer, 1990.
[23] L. Debnath and P. Mikusinski, *Introduction to Hilbert Spaces With Applications*. San Diego, CA, USA: Academic press, 1999.
[24] R. J. Serfling, *Approximation Theorems of Mathematical Statistics*. Hoboken, NJ, USA: Wiley, 2009.
[25] L. Zwald and G. Blanchard, "On the convergence of eigenspaces in kernel principal component analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 18, 2005.
[26] B. Li and J. Song, "Nonlinear sufficient dimension reduction for functional data," *Ann. Statist.*, vol. 45, no. 3, 2017.
[27] B. Li and J. Song, "Dimension reduction for functional data based on weak conditional moments," *Ann. Statist.*, vol. 50, no. 1, pp. 107–128, 2022.
[28] Y. Fujikoshi, V. V. Ulyanov, and R. Shimizu, *Multivariate Statistics: High-Dimensional and Large-Sample Approximations*. Hoboken, NJ, USA: Wiley, 2011.
[29] H. Hung, P. Wu, I. Tu, and S. Huang, "On multilinear principal component analysis of order-two tensors," *Biometrika*, vol. 99, no. 3, pp. 569–583, 2012.
[30] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

Joni Virta received the Ph.D. degree in statistics from the University of Turku, Finland, in 2018. He is currently an Academy Research Fellow and Assistant Professor of Statistics with the University of Turku. His research interests include multivariate statistics, linear and non-linear dimension reduction, and asymptotic statistics.

Andreas Artemiou received the M.Sc. and Ph.D. degrees in statistics from Pennsylvania State University, in 2008 and 2010, respectively. He was a Lecturer, Senior Lecturer and Reader of Statistics with Cardiff University, from 2013 to 2023, and an Assistant Professor with Michigan Technological University. Currently, he is a Professor of Quantitative Methods and Analytics with the University of Limassol, Cyprus. His research interests include high-dimensional methods, dimension reduction, and machine learning/statistical learning.