Applied and Computational Harmonic Analysis 79 (2025) 101798

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

journal homepage: www.elsevier.com/locate/acha

Full Length Article

Permutation-invariant representations with applications to graph deep learning

Radu Balan [a,*,1], Naveed Haghani [b], Maneesh Singh [c]

[a] Department of Mathematics, University of Maryland, College Park, MD 20742, USA
[b] Applied Mathematics and Statistics and Scientific Computation Program, University of Maryland, College Park, MD 20742, USA
[c] Verisk Analytics, Jersey City, NJ 07310, USA

ARTICLE INFO

Communicated by David Donoho

MSC:
05C62
20C32
68R12

Keywords:
Permutation invariance
Bi-Lipschitz embedding
Frames

ABSTRACT

This paper presents primarily two Euclidean embeddings of the quotient space generated by matrices that are identified modulo arbitrary row permutations. The original application is in deep learning on graphs where the learning task is invariant to node relabeling. Two embedding schemes are introduced, one based on sorting and the other based on algebras of multivariate polynomials. While both embeddings exhibit a computational complexity exponential in problem size, the sorting based embedding is globally bi-Lipschitz and admits a low dimensional target space. Additionally, an almost everywhere injective scheme can be implemented with minimal redundancy and low computational cost. In turn, this proves that almost any classifier can be implemented with an arbitrary small loss of performance. Numerical experiments are carried out on two datasets, a chemical compound dataset (QM9) and a proteins dataset (PROTEINS_FULL).

1. Introduction

This paper is motivated by a class of problems in graph deep learning, where the primary task is either graph classification or graph regression. In either case, the result should be invariant to arbitrary permutations of graph nodes.

As we explain below, the mathematical problem analyzed in this paper is a special case of the permutation invariance issue described above. To set the notations consider the vector space $\mathbb{R}^{n\times d}$ of $n\times d$ matrices endowed with the Frobenius norm $\|X\| = \left(trace(XX^T)\right)^{1/2}$ and its associated Hilbert-Schmidt scalar product, $\langle X, Y\rangle = trace(XY^T)$. Let S_n denote the symmetric group of $n\times n$ permutation matrices. S_n is a finite group of size $|S_n| = n!$.

On $\mathbb{R}^{n\times d}$ we consider the equivalence relation $\sim$ induced by the symmetric group of permutation matrices S_n as follows. Let $X, Y \in \mathbb{R}^{n\times d}$. Then we say $X \sim Y$ if there is $P \in S_n$ so that $Y = PX$. In other words, two matrices are equivalent if one is a row permutation of the other. The equivalence relation induces a natural distance on the quotient space $\widehat{\mathbb{R}^{n\times d}} := \mathbb{R}^{n\times d}/\sim$,

$$\mathbf{d} : \widehat{\mathbb{R}^{n\times d}} \times \widehat{\mathbb{R}^{n\times d}} \to \mathbb{R} \quad , \quad \mathbf{d}(\hat{X}, \hat{Y}) = \min_{\Pi\in S_n} \|X - \Pi Y\|. \tag{1.1}$$

* Corresponding author.
E-mail addresses: rvbalan@umd.edu (R. Balan), nhaghan1@umd.edu (N. Haghani), dr.maneesh.singh@ieee.org (M. Singh).

[1] R.B. was supported in part by NSF under Grants DMS-1816608 and DMS-2108900 and by a Simons Foundation fellowship 818333. He is grateful to the anonymous referee for a careful and detailed review that led to a much-improved manuscript.

https://doi.org/10.1016/j.acha.2025.101798
Received 3 June 2022; Accepted 17 July 2025

Available online 6 August 2025

This makes $(\widehat{\mathbb{R}^{n\times d}}, \mathbf{d})$ a complete metric space.

Our main problem can now be stated as follows:

Problem 1.1. Given $n, d \geq 1$ positive integers, find m and a bi-Lipschitz map $\hat{\alpha} : (\widehat{\mathbb{R}^{n\times d}}, \mathbf{d}) \to (\mathbb{R}^m, \|\cdot\|_2)$.

Explicitly the problem can be restated as follows. One is asked to construct a map $\alpha : \mathbb{R}^{n\times d} \to \mathbb{R}^m$ that satisfies the following conditions:

1. If $X, Y \in \mathbb{R}^{n\times d}$ so that $X \sim Y$ then $\alpha(X) = \alpha(Y)$
2. If $X, Y \in \mathbb{R}^{n\times d}$ so that $\alpha(X) = \alpha(Y)$ then $X \sim Y$
3. There are constants $0 < a_0 \leq b_0$ so that for any $X, Y \in \mathbb{R}^{n\times d}$,

$$a_0 \min_{\Pi\in S_n} \|X - \Pi Y\| \leq \|\alpha(X) - \alpha(Y)\|_2 \leq b_0 \min_{\Pi\in S_n} \|X - \Pi Y\|. \tag{1.2}$$

Condition (1) allows us to lift α to the quotient space $\widehat{\mathbb{R}^{n\times d}}$. Thus $\hat{\alpha}(\hat{X}) = \alpha(X)$ is well-defined. Condition (2) says that $\hat{\alpha}$ is injective (or, that α is faithful with respect to the equivalence relation $\sim$). Condition (3) says that $\hat{\alpha}$ is bi-Lipschitz with constants a_0, b_0. By a slight abuse of notation, when α satisfies (1) we shall use the same letter to denote the map $\alpha : \mathbb{R}^{n\times d} \to \mathbb{R}^m$ as well as the induced map on the quotient space $\alpha : \widehat{\mathbb{R}^{n\times d}} \to \mathbb{R}^m$.

For $X, Y \in \mathbb{R}^{n\times d}$, $\mathbf{d}(X, Y)$ denotes the same quantity in (1.1). In this case $\mathbf{d}$ is only a semi-distance on $\mathbb{R}^{n\times d}$, i.e., it is symmetric, non-negative and satisfies the triangle inequality but fails the positivity condition.

One approach to embedding $\widehat{\mathbb{R}^{n\times d}}$ is to consider the convex set of probability measures on $\mathbb{R}^d$, $\mathcal{P}(\mathbb{R}^d)$, and the map

$$\alpha_\infty : \mathbb{R}^{n\times d} \to \mathcal{P}(\mathbb{R}^d) \ , \ \alpha_\infty(X) = \frac{1}{n}\sum_{k=1}^n \delta(\cdot - x_k), \tag{1.3}$$

where $[x_1, \ldots, x_n] = X^T$, i.e., x_k is the k^{th} row of X reshaped as a vector, and δ denotes the Dirac measure. When $\mathcal{P}(\mathbb{R}^d)$ is endowed with the Wasserstein-1 distance (the Earth Moving Distance), known also as the Kantorovich-Rubinstein metric,

$$d_{KR}(p, q) = \inf_{\substack{\pi \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d): \\ \pi(\cdot, \mathbb{R}^d) = p \\ \pi(\mathbb{R}^d, \cdot) = q}} \int_{\mathbb{R}^d\times\mathbb{R}^d} \|x - y\| \, d\pi(x, y) \tag{1.4}$$

the distance between $a_\infty(X)$ and $a_\infty(Y)$ becomes

$$d_{KR}(a_\infty(X), a_\infty(Y)) = \min_{\Pi\in S_n} \sum_{k=1}^n \|x_k - (\Pi Y)_k\|.$$

By the Kantorovich-Rubinstein theorem ([12]Theorem 1.14), d_{KR} extends to a norm on the linear space of bounded signed Borel measures on $\mathbb{R}^d$, $\mathcal{M}_b(\mathbb{R}^d)$. It is easy to verify that

$$\mathbf{d}(\hat{X}, \hat{Y}) \leq d_{KR}(a_\infty(X), a_\infty(Y)) \leq \sqrt{n}\mathbf{d}(\hat{X}, \hat{Y}),$$

which proves that a_∞ provides an embedding into a normed linear space. Yet this embedding does not solve the problem since the linear space $\mathcal{M}_b(\mathbb{R}^d)$ is infinite dimensional. As a related remark, we note that the Wasserstein-2 distance W_2 defined by:

$$W_2(p, q)^2 = \inf_{\substack{\pi \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d): \\ \pi(\cdot, \mathbb{R}^d) = p \\ \pi(\mathbb{R}^d, \cdot) = q}} \int_{\mathbb{R}^d\times\mathbb{R}^d} \|x - y\|_2^2 \, d\pi(x, y) \tag{1.5}$$

produces an isometric embedding in the metric space $(\mathcal{P}(\mathbb{R}^d), d_{W2})$. However, unlike the Kantorovich-Rubinstein metric, this distance does not extend to a norm on the linear space $\mathcal{M}_b(\mathbb{R}^d)$, although it is Lipschitz equivalent to a negative-order homogeneous Sobolev norm [34].

Instead of the previous infinite-dimensional embedding, we consider two different classes of embeddings. To illustrate these two constructions, consider the simplest case $d = 1$.

1. *Algebraic Embedding.* For $x \in \mathbb{R}^n$, $x = (x_1, \ldots, x_n)^T$, construct the polynomial $P_x(z) = (z - x_1)\cdots(z - x_n)$ and then expand the product: $P_x(z) = z^n + c_1(x)z^{n-1} + \cdots + c_n(x)$. Using Vieta's formulas and Newton-Girard identities, an algebraically equivalent description of P_x is given by the symmetric polynomials:

$$\alpha : \mathbb{R}^n \to \mathbb{R}^n \ , \ \alpha(x) = \left(\sum_{k=1}^n x_k, \sum_{k=1}^n x_k^2, \ldots, \sum_{k=1}^n x_k^n\right). \tag{1.6}$$

It is not hard to see that this map satisfies Conditions (1) and (2) and therefore lifts to an injective continuous map $\hat{\alpha}$ on $\hat{\mathbb{R}^n}$. Yet it is not Lipschitz [23], let alone bi-Lipschitz. The approach in [23] can be used to modify α to a Lipschitz continuous map, but, for the same reason as described in that paper, it cannot be "fixed" to a bi-Lipschitz embedding. In Section 2 we show how to construct an algebraic Lipschitz embedding in the case $d > 1$.

2. *Sorting Embedding.* For $x \in \mathbb{R}^n$, consider the sorting map

$$\downarrow : \mathbb{R}^n \to \mathbb{R}^n \ , \ \downarrow(x) = (x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(n)})^T \tag{1.7}$$

where the permutation π is so that $x_{\pi(1)} \geq x_{\pi(2)} \geq \cdots \geq x_{\pi(n)}$. It is obvious that $\downarrow$ satisfies Conditions (1) and (2) and therefore lifts to an injective map on $\widehat{\mathbb{R}^{n\times d}}$. As we see in Section 3, the map $\downarrow$ is bi-Lipschitz. In fact it is isometric, and hence produces an ideal embedding. Our work in Section 3 is to extend such construction to the more general case $d > 1$.

The algebraic embedding is a special case of the more general *kernel method* that can be thought of as a projection of the measure $a_\infty(X)$ onto a finite dimensional space, e.g., the space of polynomials spanned by $\{X, X^2, \cdots, X^n\}$. In applications such a kernel method is known as a "Readout Map" [47], based on "Sum Pooling".

The sorting embedding has been used in applications under the name of "Pooling Map" [47], based on "Max Pooling". A naïve extension of the unidimensional map (1.7) to the case $d > 1$ might employ the lexicographic order: order monotone decreasing the rows according to the first column, and break the tie by going to the next column. While this gives rise to an injective map,[2] it is easy to see it is not even continuous, let alone Lipschitz.

The main work in this paper is to extend the sorting embedding to the case $d > 1$ using a three-step procedure: first, embed $\mathbb{R}^{n\times d}$ into a larger vector space $\mathbb{R}^{n\times D}$; then, apply $\downarrow$ in each column independently; and finally perform a dimension reduction by a linear map into $\mathbb{R}^{2nd}$. Similar to the phase retrieval problem ([2,9,4]), the redundancy introduced in the first step counterbalances the loss of information (here, relative order of one column with respect to another) in the second step.

A summary of the main results presented in this paper is contained in the following result.

Theorem 1.2. *Consider the metric space* $(\widehat{\mathbb{R}^{n\times d}}, \mathbf{d})$.

1. *(Polynomial Embedding) There exists a Lipschitz injective map*

$$\hat{\alpha} : \widehat{\mathbb{R}^{n\times d}} \to \mathbb{R}^m$$

with $m = \binom{d+n}{d}$. *Two explicit constructions of this map are given in (2.8) and (2.9).*

2. *(Sorting-based Embedding) There exists a class of bi-Lipschitz maps*

$$\hat{\beta}_{A,B} : (\widehat{\mathbb{R}^{n\times d}}, \mathbf{d}) \to (\mathbb{R}^m, \|\cdot\|) \ , \ \hat{\beta}_{A,B}(\hat{X}) = B\left(\hat{\beta}_A(\hat{X})\right)$$

with $m = 2nd$, *where each map* $\hat{\beta}_{A,B}$ *is the composition of two bi-Lipschitz maps: a full-rank linear operator* $B : \mathbb{R}^{n\times D} \to \mathbb{R}^m$, *with the nonlinear bi-Lipschitz map* $\hat{\beta}_A : \widehat{\mathbb{R}^{n\times d}} \to \mathbb{R}^{n\times D}$ *parametrized by a matrix* $A \in \mathbb{R}^{d\times D}$, *so-called called "key" matrix. Explicitly,* $\hat{\beta}_A(\hat{X}) = \downarrow(XA)$, *where* $\downarrow$ *acts column-wise. These maps are characterized by the following properties:*

(a) *For* $D = 1 + (d-1)n!$, *any* $A \in \mathbb{R}^{d\times(1+(d-1)n!)}$ *whose columns form a full spark frame defines a bi-Lipschitz map* $\hat{\beta}_A$ *on* $\widehat{\mathbb{R}^{n\times d}}$. *Furthermore, a lower Lipschitz constant is given by the smallest* d^{th} *singular value among all* $d \times d$ *sub-matrices of A,* $\min_{J\subset[D],|J|=d} s_d(A[J])$.

(b) *For any matrix ("key")* $A \in \mathbb{R}^{d\times D}$ *such that the map* $\hat{\beta}_A$ *is injective, then* $\hat{\beta}_A : (\widehat{\mathbb{R}^{n\times d}}, \mathbf{d}) \to (\mathbb{R}^{n\times D}, \|\cdot\|)$ *is bi-Lipschitz. Furthermore, an upper Lipschitz constant is given by* $s_1(A)$, *the largest singular value of A.*

(c) *Assume* $A \in \mathbb{R}^{d\times D}$ *is such that the map* $\hat{\beta}_A$ *is injective (we call such an A "universal key"). Then for almost any linear map* $B : \mathbb{R}^{n\times D} \to \mathbb{R}^{2nd}$ *the map* $\hat{\beta}_{A,B} = B \circ \hat{\beta}_A$ *is bi-Lipschitz.*

An immediate consequence of this result is the following corollary, whose proof is included in subsection 3.5:

Corollary 1.3. *Let* $\beta : \mathbb{R}^{n\times d} \to \mathbb{R}^m$ *induce a bi-Lipschitz embedding* $\hat{\beta} : \widehat{\mathbb{R}^{n\times d}} \to \mathbb{R}^m$ *of the metric space* $(\widehat{\mathbb{R}^{n\times d}}, \mathbf{d})$ *into* $(\mathbb{R}^m, \|\cdot\|_2)$.

1. *For any continuous function* $f : \mathbb{R}^{n\times d} \to \mathbb{R}$ *invariant to row-permutation (i.e.,* $f(PX) = f(X)$ *for every* $X \in \mathbb{R}^{n\times d}$ *and* $P \in \mathcal{S}_n$*) there exists a continuous function* $g : \mathbb{R}^m \to \mathbb{R}$ *such that* $f = g \circ \beta$. *Conversely, for any* $g : \mathbb{R}^m \to \mathbb{R}$ *continuous function, the function* $f = g \circ \beta : \mathbb{R}^{n\times d} \to \mathbb{R}$ *is continuous and row-permutation invariant.*

[2] For $\varepsilon > 0$, the lexicographic sorting according to the first column maps $X = \begin{bmatrix} 1+\varepsilon & 1 \\ 1 & -1 \end{bmatrix}$ to $X_{sort} = X$, and $Y = \begin{bmatrix} 1-\varepsilon & 1 \\ 1 & -1 \end{bmatrix}$ to $Y_{sort} = \begin{bmatrix} 1 & -1 \\ 1-\varepsilon & 1 \end{bmatrix}$. While $d(X,Y) = 2\varepsilon$, $\|X_{sort} - Y_{sort}\| = \sqrt{2\varepsilon^2 + 8}$.

2. *For any Lipschitz continuous function $f : \mathbb{R}^{n\times d} \to \mathbb{R}$ invariant to row-permutation (i.e., $f(PX) = f(X)$ for every $X \in \mathbb{R}^{n\times d}$ and $P \in \mathcal{S}_n$) there exists a Lipschitz continuous function $g : \mathbb{R}^m \to \mathbb{R}$ such that $f = g \circ \beta$. Conversely, for any $g : \mathbb{R}^m \to \mathbb{R}$ Lipschitz continuous function, the function $f = g \circ \beta : \mathbb{R}^{n\times d} \to \mathbb{R}$ is Lipschitz continuous and row-permutation invariant.*

The structure of the paper is as follows. Section 2 contains the algebraic embedding method and encoders α described at part (1) of Theorem 1.2. Corollary 2.3 contains part (1) of the main result stated above. Section 3 introduces the sorting based embedding procedure and describes the key-based encoder β. Necessary and sufficient conditions for key universality are presented in Proposition 3.9; the injectivity of the encoder described at part (2.a) of Theorem 1.2 is proved in Theorem 3.10; the bi-Lipschitz property of any universal key described at part (2.b) of Theorem 1.2 is shown in Theorem 3.12; the dimension reduction statement (2.c) of Theorem 1.2 is included in Theorem 3.15. Proof of Corollary 1.3 is presented in subsection 3.5. Section 4 contains applications to graph deep learning. These applications use Graph Convolution Networks and the numerical experiments are carried out on two graph datasets: a chemical compound dataset (`QM9`) and a protein dataset (`PROTEINS_FULL`).

While the motivation of this analysis is provided by graph deep learning applications, this is primarily a mathematical paper. Accordingly the formal theory is presented first, and then is followed by the machine learning application. Those interested in the application (or motivation) can skip directly to Section 4.

Notations. For an integer $d \geq 1$, $[d] = \{1, 2, \dots, d\}$. For a matrix $X \in \mathbb{R}^{n\times d}$, $x_1, \dots x_d \in \mathbb{R}^n$ denote its columns, $X = [x_1 | \cdots | x_d]$. All norms are Euclidean; for a matrix X, $\|X\| = \sqrt{trace(X^T X)} = \sqrt{\sum_{k,j} |X_{k,j}|^2}$ denotes the Frobenius norm; for vectors x, $\|x\| = \|x\|_2 = \sqrt{\sum_j |x_j|^2}$.

1.1. Prior works

Several methods for representing orbits of vector spaces under the action of permutation (sub)groups have been studied in literature. Here we describe some of these results, without claiming an exhaustive literature survey.

A rich body of literature emanated from the early works on symmetric polynomials and group invariant representations of Hilbert, Noether, Klein and Frobenius. They are part of standard commutative algebra and finite group representation theory.

Prior works on permutation invariant mappings have predominantly employed some form of summing procedure [31,46], though some have alternatively employed some form of sorting procedure [48].

The idea of summing over the output nodes of an equivariant network has been well studied ([36]). The algebraic invariant theory goes back to Hilbert and Noether (for finite groups) and then continuing with the continuous invariant function theory of Weyl and Wigner (for compact groups), who posited that a generator function $\psi : X \to \mathbb{R}$ gives rise to a function $E : X \to \mathbb{R}$ invariant to the action of a finite group G on X, $(g, x) \mapsto g.x$, via the averaging formula $E(x) = \frac{1}{|G|} \sum_{g\in G} \psi(g.x)$.

More recently, this approach provided the framework for universal approximation results of G-invariant functions. [31] showed that invariant or equivariant networks must satisfy a fixed point condition. The equivariant condition is naturally realized by GNNs. The invariance condition is realized by GNNs when followed by summation on the output layer, as was further shown in [24], [32] and [36]. [46] proved universal approximation results over compact sets for continuous functions invariant to the action of finite or continuous groups. In [19], the authors obtained bounds on the separation power of GNNs in terms of the Weisfeiler-Leman (WL) tests by tensorizing the input-output mapping. [42] studied approximations of equivariant maps, while [13] showed that if a GNN with sufficient expressivity is well trained, it can solve the graph isomorphism problem.

The authors of [43] designed an algorithm for processing sets with no natural orderings. The algorithm applies an attention mechanism to achieve permutation invariance with the attention keys being generated by a Long-Short Term Memory (LSTM) network. Attention mechanisms amount to a weighted summing and therefore can be considered to fall within the domain of summing based procedures.

In [28], the authors designed a permutation invariant mapping for graph embeddings. The mapping employs two separate neural networks, both applied over the feature set for each node. One neural network produces a set of new embeddings, the other serves as an attention mechanism to produce a weighed sum of those new embeddings.

Sorting based procedures for producing permutation invariant mappings over single dimensional inputs have been addressed and used by [47], notably in their *max pooling* procedure.

The authors of [37] developed a permutation invariant mapping *pointnet* for point sets that is based on a *max* function. The mapping takes in a set of vectors, processes each vector through a neural network followed by an scalar output function, and takes the maximum of the resultant set of scalars.

The paper [48] introduced *sort-pooling*. *Sort-pooling* orders the latent embeddings of a graph according to the values in a specific, predetermined column. All rows of the latent embeddings are sorted according to the values in that column. While this gives rise to an injective map, it is easy to see it is not even continuous, let alone Lipschitz. The same issue arises with any lexicographic ordering, including the well-known Weisfeiler-Leman embedding [44]. Our paper introduces a novel method that bypasses this issue.

As shown in [32], the sum pooling-based GNNs provides universal approximations for of any permutation invariant continuous function but only on *compacts*. Our sorting based embedding removes the compactness restriction as well as it extends to all Lipschitz maps.

Last but not least, two very recent preprints [16,11] we became aware of after finishing this draft, propose similar or related constructions. The paper [16] considers a family of encoding schemes invariant to certain subgroups of the orthogonal group. In

particular, their construction in Remark 2.2 is similar to the $\hat{\beta}_{A,B}$ encoder. Remarkably, they show that the dimension D in part 2.a of Theorem 2.1 can be lowered to $2nd+1$. The authors of [11] consider a certain embedding of the quotient space using maxima of co-orbits. For the case of natural distance d, this maximum value can be found efficiently by solving Linear Assignment problems.

While this paper is primarily mathematical in nature, methods developed here are applied to two graph datasets, `QM9` and `PROTEINS_FULL`. Researchers have applied various graph deep learning techniques to both datasets. In particular, [20] studied extensively the `QM9` dataset, and compared their method with many other algorithms proposed by that time. In addition to machine learning problems considered here, bi-Lipschitz embeddings of the metric space $(\widehat{\mathbb{R}^{n\times d}}, \mathbf{d})$ may be modified to produce embeddings of certain spaces of probability measures such as those considered in [33].

2. Algebraic embeddings

The algebraic embedding presented in this section can be thought of a special kernel to project equation (1.3) onto.

2.1. Kernel methods

The kernel method employs a family of continuous kernels (test) functions, $\{K(x;y)\ ;\ x\in\mathbb{R}^d\ ,\ y\in Y\}$ parametrized/indexed by a set Y. The measure representation $\mu = a_\infty(X)$ in (1.3) yields a nonlinear map

$$\alpha : \mathbb{R}^{n\times d} \to C(Y),\quad X \mapsto F(y) = \int_{R^d} K(x;y)d\mu$$

given by

$$\alpha(X)(y) = \frac{1}{n}\sum_{k=1}^{n} K(x_k;y)$$

The embedding Problem 1.1) can be restated as follows. One is asked to find a finite family of kernels $\{K(x;y)\ ;\ x\in\mathbb{R}^d\ ,\ y\in Y\}$, $m=|Y|$ so that

$$\hat{\alpha} : (\widehat{\mathbb{R}^{n\times d}}, \mathbf{d}) \to l^2(Y) \sim (\mathbb{R}^m, \|\cdot\|_2),\quad (\hat{\alpha}(\hat{X}))_y = \frac{1}{n}\sum_{k=1}^{n} K(x_k;y) \tag{2.1}$$

is injective, Lipschitz or bi-Lipschitz.

Two natural choices for the kernel K are the Gaussian kernel and the complex exponential (i.e., the Fourier) kernel:

$$K_G(x,y) = e^{-\|x-y\|^2/\sigma^2}, K_F(x,y) = e^{2\pi i\langle x,y\rangle},$$

where in both cases $Y\subset\mathbb{R}^d$. The two kernels are naturally related via Bochner's theorem, as shown in [38]. The Fourier kernel raises deep questions about relationship between singular values of irregular Fourier frames and minimal frequency separation distance. Partial results on this problem can be derived from [27].

In this paper we analyze a different kernel, namely the polynomial kernel $K_P(x,y) = x_1^{y_1}x_2^{y_2}\cdots x_d^{y_d}$, $Y\subset\{0,1,2,\dots,n\}^d$.

2.2. The polynomial embedding

Since the polynomial representation is intimately related to the Hilbert-Noether algebraic invariants theory [21] and the Hilbert-Weyl theorem, it is advantageous to start our construction from a different perspective.

The linear space $\mathbb{R}^{n\times d}$ is isomorphic to $\mathbb{R}^{nd}$ by stacking the columns one on top of each other. In this case, the action of the permutation group S_n can be recast as the action of the subgroup $I_d\otimes S_n$ of the bigger group S_{nd} on $\mathbb{R}^{nd}$. Specifically, let us denote by $\sim_G$ the equivalence relation

$$x,y\in\mathbb{R}^{nd}\ \ ,\ \ x\sim_G y \Longleftrightarrow y=\Pi x\ ,\ \text{for some } \Pi\in G$$

induced by a subgroup G of S_{nd}. In the case $G = I_d\otimes S_n = \{diag_d(P)\ ,\ P\in S_n\}$ of block diagonal permutation, obtained by repeating d times the same $P\in S_n$ permutation along the main diagonal, two vectors $x,y\in\mathbb{R}^{nd}$ are $\sim_G$ equivalent iff there is a permutation matrix $P\in S_n$ so that $y(1+(k-1)n:kn) = Px(1+(k-1)n:kn)$ for each $1\le k\le d$. In other words, each of the d disjoint n-subvectors in y and x are related by the same permutation. In this framework, the Hilbert-Weyl theorem (Theorem 4.2, Chapter XII, in [29]) states that the ring of invariant polynomials is finitely generated. The Göbel's algorithm (Section 3.10.2 in [21]) provides a recipe to find a complete set of invariant polynomials. In the following we provide a direct approach to construct a complete set of polynomial invariants.

Let $\mathbb{R}[\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_d]$ denote the algebra of polynomials in d-variables with real coefficients. Let us denote $X\in\mathbb{R}^{n\times d}$ a generic data matrix. Each row of this matrix defines a linear form over $\mathbf{x}_1,...\mathbf{x}_d$, $\lambda_k = X_{k,1}\mathbf{x}_1+\cdots+X_{k,d}\mathbf{x}_d$. Let us denote by $\mathbb{R}[\mathbf{x}_1,\dots,\mathbf{x}_d][\mathbf{t}]$ the algebra of polynomials in variable $\mathbf{t}$ with coefficients in the ring $\mathbb{R}[\mathbf{x}_1,\dots,\mathbf{x}_d]$. Notice $\mathbb{R}[\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_d][\mathbf{t}] = \mathbb{R}[\mathbf{t},\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_d]$ by rearranging the terms according to degree in $\mathbf{t}$. Thus $\lambda_k\in\mathbb{R}[\mathbf{x}_1,\dots,\mathbf{x}_d]\subset\mathbb{R}[\mathbf{x}_1,\dots,\mathbf{x}_d][\mathbf{t}]$ can be encoded as zeros of a polynomial P_X of degree n in variable $\mathbf{t}$ with coefficients in $\mathbb{R}[\mathbf{x}_1,\dots,\mathbf{x}_d]$:

$$P_X(\mathbf{t},\mathbf{x}_1,\dots,\mathbf{x}_d)=\prod_{k=1}^{n}(\mathbf{t}-\lambda_k(\mathbf{x}_1,\dots,\mathbf{x}_d))=\prod_{k=1}^{n}(\mathbf{t}-X_{k,1}\mathbf{x}_1-\dots-X_{k,d}\mathbf{x}_d) \tag{2.2}$$

Due to identification $\mathbb{R}[\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_d][\mathbf{t}]=\mathbb{R}[\mathbf{t},\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_d]$, we obtain that $P_X\in\mathbb{R}[\mathbf{t},\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_d]$ is a homogeneous polynomial of degree n in $d+1$ variables. Let $\mathbb{R}_n[\mathbf{t},\mathbf{x}_1,\dots,\mathbf{x}_d]$ denote the vector space of homogeneous polynomials in $d+1$ variables of degree n with real coefficients. Notice the real dimension of this vector space is

$$\dim_{\mathbb{R}}\mathbb{R}_n[\mathbf{t},\mathbf{x}_1,\dots,\mathbf{x}_d]=\begin{pmatrix}n+d\\ d\end{pmatrix}=\begin{pmatrix}n+d\\ n\end{pmatrix}. \tag{2.3}$$

By noting that P_X is monic in $\mathbf{t}$ (the coefficient of $\mathbf{t}^n$ is always 1) we obtain an injective embedding of $\widehat{\mathbb{R}^{n\times d}}$ into $\mathbb{R}^m$ with $m=\dim_{\mathbb{R}}\mathbb{R}_n[\mathbf{t},\mathbf{x}_1,\dots,\mathbf{x}_d]-1$ via the coefficients of P_X similar to (1.6). This is summarized in the following theorem:

Theorem 2.1. *The map $\alpha_0:\mathbb{R}^{n\times d}\to\mathbb{R}^{m-1}$ with $m=\begin{pmatrix}n+d\\ d\end{pmatrix}$ given by the (non-trivial) coefficients of polynomial $P_X\in\mathbb{R}_n[\mathbf{t},\mathbf{x}_1,\dots,\mathbf{x}_d]$ lifts to an analytic embedding $\hat{\alpha}_0$ of $(\widehat{\mathbb{R}^{n\times d}},\mathbf{d})$ into $\mathbb{R}^m$. Specifically, for $X\in\mathbb{R}^{n\times d}$ expand the polynomial*

$$P_X(\mathbf{t},\mathbf{x}_1,\dots,\mathbf{x}_d)=\prod_{k=1}^{n}(\mathbf{t}-X_{k,1}\mathbf{x}_1-\dots-X_{k,d}\mathbf{x}_d)=\mathbf{t}^n+\sum_{\substack{p_0,p_1,\dots,p_d\geq 0\\ p_0+\dots+p_d=n\\ p_0<n}}c_{p_0,p_1,\dots,p_d}\mathbf{t}^{p_0}\mathbf{x}_1^{p_1}\cdots\mathbf{x}_d^{p_d} \tag{2.4}$$

Then

$$X\in\mathbb{R}^{n\times d}\mapsto\alpha_0(X)=(c_{p_0,p_1,\dots,p_d})_{(p_0,p_1,\dots,p_d)\in I_{n,d}} \tag{2.5}$$

where the index set is given by

$$I_{n,d}=\{(p_0,p_1,\dots,p_d)\ ,\ 0\leq p_0,p_1,\dots,p_d\ ,\ p_0<n\ ,\ p_0+p_1+\dots+p_d=n\} \tag{2.6}$$

and is of cardinality $|I_{n,d}|=m-1$. The map $\hat{\alpha}_0:\widehat{\mathbb{R}^{n\times d}}\to\mathbb{R}^{m-1}$ is the lifting of α_0 to the quotient space.

Proof. Since for any permutation π with associated permutation matrix $\Pi\in\mathcal{S}_n$,

$$P_{\Pi X}(\mathbf{t},\mathbf{x}_1,\cdots,\mathbf{x}_d)=\prod_{k=1}^{n}(\mathbf{t}-X_{\pi(k),1}\mathbf{x}_1-\dots-X_{\pi(k),d}\mathbf{x}_d)=P_X(\mathbf{t},\mathbf{x}_1,\dots,\mathbf{x}_d),$$

it follows that α_0 is invariant to the action of $\mathcal{S}_n$, $\alpha_0(X)=\alpha_0(\Pi X)$. Thus α_0 lifts to a map $\hat{\alpha}_0$ on $\widehat{\mathbb{R}^{n\times d}}$.

The coefficients of polynomial P_X depend analytically on its roots (Vieta's formulas), hence on entries of matrix X.

The only remaining claim is that if $X,Y\in\mathbb{R}^{n\times d}$ so that $\alpha_0(X)=\alpha_0(Y)$ then there is $\Pi\in\mathcal{S}_n$ so that $Y=\Pi X$. Assume $P_X=P_Y$. For each choice $(\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_d)=(f(1),\dots,f(d))$ in $\mathbb{R}^d$, the n real zeros of the two polynomials in $\mathbf{t}$, $P_X(\mathbf{t},f(1),\dots,f(d))$ and $P_Y(\mathbf{t},f(1),\dots,f(d))$, coincide. Therefore $Xf\sim Yf$ for each $f\in\mathbb{R}^d$. Let $D=1+(d-1)n!$ and choose $F\in\mathbb{R}^{d\times D}$ so that each subset of d columns are linearly independent, in other words, the set $\mathcal{F}=\{f_1,f_2,\dots,f_D\}$ formed by the D columns of F is a full spark frame in $\mathbb{R}^d$, see [1]. As proved in [1], almost every such set is a full spark frame. Then for each $1\leq k\leq D$ there is a permutation $\Pi_k\in\mathcal{S}_n$ so that $Xf_k=\Pi_kYf_k$. By the pigeonhole principle, since $|\mathcal{S}_n|=n!$, there are $1\leq k_1<k_2<\dots<k_d\leq D$ so that $\Pi_{k_1}=\Pi_{k_2}=\dots=\Pi_{k_d}$. Then $(X-\Pi_{k_1}Y)f_{k_j}=0$ for every $1\leq j\leq d$. Since $\{f_{k_1},\dots,f_{k_d}\}$ is linearly independent, it follows that $X-\Pi_{k_1}Y=0$. Thus $X\sim Y$, which ends the proof of this result. □

Remark 2.2. The invariants produced by map α_0 are proportional to those produced by the Göbel's algorithm in [21], §3.10.2. Indeed, the *nd* primary invariants are given by

$$\{c_{p,n-p,0,\dots,0}\ ,\ 0\leq p\leq n-1\}\cup\dots\cup\{c_{p,0,\dots,0,n-p}\ ,\ 0\leq p\leq n-1\}$$

corresponding to the elementary symmetric polynomials in entries of each column. The secondary invariants correspond to the remaining coefficients that have at least 2 nonzero indices among $p_1,\dots,p_d$.

The embedding provided by α_0 is analytic and injective but is not globally Lipschitz because of the polynomial growth rate. Next, we show how a simple modification of this map will make it Lipschitz. First, let us denote by L_0 the Lipschitz constant of α_0 when restricted to the closed unit ball $B_1(\mathbb{R}^{n\times d}):\{X\in\mathbb{R}^{n\times d}\ ,\ \|X\|\leq 1\}$ of $\mathbb{R}^{n\times d}$, i.e. $\|\alpha_0(X)-\alpha_0(Y)\|\leq L_0\|X-Y\|$ for any $X,Y\in\mathbb{R}^{n\times d}$ with $\|X\|,\|Y\|\leq 1$. Second, let

$$\varphi_0:\mathbb{R}\to[0,1]\ ,\ \varphi_0(x)=min(1,\frac{1}{x})=\begin{cases}1 & if\quad x\leq 1\\ \frac{1}{x} & if\quad x>1\end{cases} \tag{2.7}$$

be a Lipschitz monotone decreasing function with Lipschitz constant 1.

Corollary 2.3. *Consider the map:*

$$\alpha_1 : \mathbb{R}^{n\times d} \to \mathbb{R}^m \quad , \quad \alpha_1(X) = \begin{pmatrix} \alpha_0\Big(\varphi_0(\|X\|)X\Big) \\ \|X\| \end{pmatrix}, \tag{2.8}$$

with $m = \binom{n+d}{d}$. *The map* α_1 *lifts to an injective and globally Lipschitz map* $\hat{\alpha}_1 : \widehat{\mathbb{R}^{n\times d}} \to \mathbb{R}^m$ *with Lipschitz constant* $Lip(\hat{\alpha}_1) \leq \sqrt{1+L_0^2}$.

Proof. Clearly $\alpha_1(\Pi X) = \alpha_1(X)$ for any $\Pi \in \mathcal{S}_n$ and $X \in \mathbb{R}^{n\times d}$. Assume now that $\alpha_1(X) = \alpha_1(Y)$. Then $\|X\| = \|Y\|$ and since $\hat{\alpha}_0$ is injective on $\widehat{\mathbb{R}^{n\times d}}$ it follows $\varphi(\|X\|)X = \Pi\varphi(\|Y\|)Y$ for some $\Pi \in \mathcal{S}_n$. Thus $X \sim Y$ which proves α_1 lifts to an injective map on $\widehat{\mathbb{R}^{n\times d}}$.

Now we show $\hat{\alpha}_1$ is Lipschitz on $(\widehat{\mathbb{R}^{n\times d}}, \mathbf{d})$ of appropriate Lipschitz constant. Let $X, Y' \in \mathbb{R}^{n\times d}$ and $\Pi_0 \in \mathcal{S}_n$ so that $\mathbf{d}(\hat{X}, \widehat{Y'}) = \|X - \Pi_0 Y'\|$. Let $Y = \Pi_0 Y'$ so that $\mathbf{d}(\hat{X}, \hat{Y}) = \|X - Y\|$.

Choose two matrices $X, Y \in \mathbb{R}^{n\times d}$. We claim $\|\alpha_1(X) - \alpha_1(Y)\| \leq \sqrt{1+L_0^2} \cdot \|X - Y\|$. This follows from two observations:

(i) The map

$$X \mapsto \rho(X) := \varphi_0(\|X\|)X$$

is the nearest-point map to (or, the metric projection map onto) the convex closed set $B_1(\mathbb{R}^{n\times d})$. This means $\|\varphi_0(\|X\|)X - Z\| \leq \|X - Z\|$ for any $Z \in B_1(\mathbb{R}^{n\times d})$.

(ii) The nearest-point map to a convex closed subset of a Hilbert space is Lipschitz with constant 1, i.e. it shrinks distances, see [35].

These two observations yield:

$$\begin{aligned} \|\alpha_1(X) - \alpha_1(Y)\|^2 &= \|\alpha_0(\rho(Y)) - \alpha_0(\rho(Y))\|^2 + |\|X\| - \|Y\||^2 \\ &\leq L_0^2 \cdot \|\rho(X) - \rho(Y)\|^2 + \|X - Y\|^2 \leq (1+L_0^2)\|X - Y\|^2. \end{aligned}$$

This concludes the proof of this result. □

A simple modification of ϕ_0 can produce a C^∞ map by smoothing it out around $x = 1$.

On the other hand the lower Lipschitz constant of $\hat{\alpha}_1$ is 0 due to terms of the form $X_{i,j}^k$ with $k \geq 2$. In [23], the authors built a Lipschitz map by a retraction to the unit sphere instead of unit ball. Inspired by their construction, a modification of α_0 in their spirit reads:

$$\alpha_2 : \mathbb{R}^{n\times d} \to \mathbb{R}^m, \quad \alpha_2(X) = \begin{pmatrix} \|X\|\alpha_0\Big(\frac{X}{\|X\|}\Big) \\ \|X\| \end{pmatrix}, \text{ if } X \neq 0, \text{ and } \alpha_2(0) = 0. \tag{2.9}$$

It is easy to see that α_2 satisfies the non-parallel property in [23] and is Lipschitz with a slightly better constant than α_1 (the constant is determined by the tangential derivatives of α_0). But, for the same reasons as in [23] this map is not bi-Lipschitz.

2.3. Dimension reduction in the case $d = 2$ and consequences

In this subsection we analyze the case $d = 2$. The embedding dimension for α_0 is $\binom{n}{2} - 1 = \frac{n(n-1)}{2} - 1$. On the other hand, consider the following approach. Each row of X defines a complex number $z_1 = X_{1,1} + iX_{1,2}$, ..., $z_n = X_{n,1} + iX_{n,2}$ that can be encoded by one polynomial of degree n with complex coefficients $Q \in \mathbb{C}_n[t]$,

$$Q(\mathbf{t}) = \prod_{k=1}^{n}(\mathbf{t} - z_k) = \mathbf{t}^n + \sum_{k=0}^{n-1}\mathbf{t}^k q_k$$

The coefficients of Q provide a $2n$-dimensional real embedding ζ_0,

$$\zeta_0 : \mathbb{R}^{n\times 2} \to \mathbb{R}^{2n} \quad , \quad \zeta_0(X) = (Re(q_{n-1}), Im(q_{n-1}), \dots, Re(q_0), Im(q_0))$$

with properties similar to those of α_0. One can similarly modify this embedding to obtain a globally Lipschitz embedding $\hat{\zeta}_1$ of $\hat{R_{n,2}}$ into $\mathbb{R}^{2n+1}$.

It is instructive to recast this embedding in the framework of commutative algebras. Indeed, let $\langle \mathbf{x}_1 - 1, \mathbf{x}_2^2 + 1\rangle$ denote the ideal generated by polynomials $\mathbf{x}_1 - 1$ and $\mathbf{x}_2^2 + 1$ in the algebra $\mathbb{R}[\mathbf{t}, \mathbf{x}_1, \mathbf{x}_2]$. Consider the quotient space $\mathbb{R}[\mathbf{t}, \mathbf{x}_1, \mathbf{x}_2]/\langle \mathbf{x}_1 - 1, \mathbf{x}_2^2 + 1\rangle$ and the quotient map $\sigma : \mathbb{R}[\mathbf{t}, \mathbf{x}_1, \mathbf{x}_2] \mapsto \mathbb{R}[\mathbf{t}, \mathbf{x}_1, \mathbf{x}_2]/\langle \mathbf{x}_1 - 1, \mathbf{x}_2^2 + 1\rangle$. In particular, let $S = \sigma(\mathbb{R}_n[\mathbf{t}, \mathbf{x}_1, \mathbf{x}_2])$ denote the vector space projected through this quotient map. Then a basis for S is given by $\{1, \mathbf{t}, \dots, \mathbf{t}^n, \mathbf{x}_2, \mathbf{x}_2\mathbf{t}, \dots, \mathbf{x}_2\mathbf{t}^{n-1}, \mathbf{x}_2\mathbf{t}^n\}$. Thus $\dim S = 2n + 2$. Let

$\mathfrak{S} = \{P_X\ ,\ X \in \mathbb{R}^{n\times 2}\} \subset \mathbb{R}_2[\mathbf{t}, \mathbf{x}_1, \mathbf{x}_2]$ denote the set of polynomials realizable as in (2.4). Then the fact that $\hat{\zeta}_0 : \mathbb{R}^{n\times 2} \to \mathbb{R}^{2n}$ is injective is equivalent to the fact that $\sigma|_{\mathfrak{S}} : \mathfrak{S} \to S$ is injective. On the other hand, note

$$\sigma(\mathfrak{S}) \subset \mathbf{t}^n + span_{\mathbb{R}}\{1, \mathbf{t}, \dots, \mathbf{t}^{n-1}, \mathbf{x}_2, \mathbf{x}_2\mathbf{t}, \dots, \mathbf{x}_2\mathbf{t}^{n-1}\}$$

where the last linear subspace is of dimension $2n$.

In the case $d = 2$ we obtain the identification $\mathbb{R}[\mathbf{t}, \mathbf{x}_1, \mathbf{x}_2]/\langle \mathbf{x}_1 - 1, \mathbf{x}_2^2 + 1\rangle = \mathbb{C}[\mathbf{t}]$ due to uniqueness of polynomial factorization.

This observation raises the following *open problem*:

For $d > 2$, is there a non-trivial ideal $I = \langle Q_1, \dots, Q_r\rangle \subset \mathbb{R}[\mathbf{t}, \mathbf{x}_1, \dots, \mathbf{x}_d]$ so that the restriction $\sigma|_{\mathfrak{S}}$ of the quotient map $\sigma : \mathbb{R}[\mathbf{t}, \mathbf{x}_1, \dots, \mathbf{x}_d] \to \mathbb{R}[\mathbf{t}, \mathbf{x}_1, \dots, \mathbf{x}_d]/I$ is injective? Here $\mathfrak{S}$ denote the set of polynomials in $\mathbb{R}_n[\mathbf{t}, \mathbf{x}_1, \dots, \mathbf{x}_d]$ realizable via (2.4).

Remark 2.4. One may ask the question whether the quaternions can be utilized in the case $d = 4$. While the quaternions form an associative division algebra, unfortunately polynomials have in general an infinite number of factorizations; this prevents an immediate extension of the previous construction to the case $d = 4$.

Remark 2.5. Similar to the construction in [23], a linear dimension-reduction technique may be applicable here (which, in fact, may answer the open problem above); this would reduce the embedding dimension to $m = 2nd + 1$ (twice the intrinsic dimension plus one for the homogenization variable). We did not explore this approach since, even if possible, it would not produce a bi-Lipschitz embedding. Instead we analyze the linear dimension-reduction technique in the next section in the context of sorting based embeddings. After finishing this draft, we noticed that the dimension reduction result is addressed in [16].

3. Sorting based embedding

In this section we present the extension of the sorting embedding (1.7) to the case $d > 1$.

The embedding is performed by composing linear and nonlinear transformations in a way that we find reminiscent of constructions associated with the phase retrieval problem. Consider a matrix $A \in \mathbb{R}^{d\times D}$ and the induced nonlinear transformation:

$$\beta_A : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times D},\quad \beta_A(X) = \downarrow(XA) \tag{3.1}$$

where $\downarrow$ is the monotone decreasing sorting operator acting in each column independently. Specifically, let $Y = XA \in \mathbb{R}^{n\times D}$ and note its column vectors $Y = [y_1, y_2, \dots, y_D]$. Then

$$\beta_A(X) = \begin{bmatrix} \Pi_1 y_1 & \Pi_2 y_1 & \cdots & \Pi_D y_D \end{bmatrix}$$

for some sorting permutations $\Pi_1, \Pi_2, \dots, \Pi_D \in \mathcal{S}_n$ which place their corresponding columns into decreasing rearrangement:

$$(\Pi_k y_k)_1 \geq (\Pi_k y_k)_2 \geq \cdots \geq (\Pi_k y_k)_n.$$

Note the obvious invariance $\beta_A(\Pi X) = \beta_A(X)$ for any $\Pi \in \mathcal{S}_n$ and $X \in \mathbb{R}^{n\times d}$. Hence β_A lifts to a map $\widehat{\beta_A}$ on $\widehat{\mathbb{R}^{n\times d}}$.

Remark 3.1. To explain the similarity we perceive with the phase retrieval problem: Recall, e.g., [4] where the data are obtained via a linear transformation of the input signal followed by the nonlinear operation of taking the absolute value of the output coefficients. Here the nonlinear operation is implemented by sorting the coefficients. In both cases, it represents the action of a particular subgroup of the unitary group – modulation by unimodular complex numbers in one case, and rearrangement by permutation in the other.

Remark 3.2. The very recent work [16] presents a similar encoding schemes for this problem. The authors of [11] replace the n entries in each column by the maximum of $\langle X, PW\rangle$ over all permutation matrices $P \in \mathcal{S}_n$ for a fixed weight matrix.

In this section we analyze necessary and sufficient conditions so that maps of type (3.1) are injective, or injective almost everywhere. First a few definitions.

Definition 3.3. A matrix $A \in \mathbb{R}^{d\times D}$ is called a *universal key* (for $\mathbb{R}^{n\times d}$) if $\widehat{\beta_A}$ is injective on $\widehat{\mathbb{R}^{n\times d}}$.

In general we refer to A as a *key* for encoder β_A.

Definition 3.4. Fix a matrix $X \in \mathbb{R}^{n\times d}$. A matrix $A \in \mathbb{R}^{d\times D}$ is said to be *admissible* (or to be an *admissible key*) for X if for any $Y \in \mathbb{R}^{n\times d}$ so that $\beta_A(X) = \beta_A(Y)$ then $Y = \Pi X$ for some $\Pi \in \mathcal{S}_n$.

In other words, $\widehat{\beta_A}^{-1}(\widehat{\beta_A}(\hat{X})) = \{\hat{X}\}$. We let $\mathcal{A}_D(X)$, or simply $\mathcal{A}(X)$, denote the set of admissible keys for X.

Definition 3.5. Fix $A \in \mathbb{R}^{d\times D}$. A matrix $X \in \mathbb{R}^{n\times d}$ is said to be *separated* by A if $A \in \mathcal{A}(X)$.

For a key A, we let $\mathfrak{S}_n(A)$, or simply $\mathfrak{S}(A)$, denote the set of *matrices separated by* A. Thus a matrix $X \in \mathfrak{S}_n(A)$ if and only if, for any matrix $Y \in \mathbb{R}^{n\times d}$, if $\beta_A(X) = \beta_A(Y)$ then $X \sim Y$.

Thus a key A is universal if and only if $\mathfrak{S}_n(A) = \mathbb{R}^{n\times d}$.

Our goal is to produce keys that are admissible for all matrices in $\mathbb{R}^{n\times d}$, or at least for almost every data matrix. As we show in Proposition 3.7 below this requires that $D \geq d$ and A is full rank. In particular this means that the columns of A form a frame for $\mathbb{R}^d$.

3.1. Characterizations of $\mathcal{A}(X)$ and $\mathfrak{S}(A)$

We start off with simple linear manipulations of sets of admissible keys and separated data matrices.

Proposition 3.6. *Fix $A \in \mathbb{R}^{d\times D}$ and $X \in \mathbb{R}^{n\times d}$.*

1. *For an invertible $d \times d$ matrix $T \in \mathbb{R}^{d\times d}$,*

$$\mathfrak{S}_n(TA) = \mathfrak{S}_n(A)T^{-1}. \tag{3.2}$$

In other words, if X is separated by A then XT^{-1} is separated by TA.

2. *For any permutation matrix $L \in S_D$ and diagonal invertible matrix $\Lambda \in \mathbb{R}^{D\times D}$,*

$$\mathfrak{S}_n(AL\Lambda) = \mathfrak{S}_n(A\Lambda L) = \mathfrak{S}_n(A). \tag{3.3}$$

In other words, if X is separated by A then X is separated also by $AL\Lambda$ as well as by $A\Lambda L$.

3. *Assume $T \in \mathbb{R}^{d\times d}$ is a $d \times d$ invertible matrix. Then*

$$\mathcal{A}_D(XT) = T^{-1}\mathcal{A}_D(X). \tag{3.4}$$

In other words, if A is an admissible key for X then $T^{-1}A$ is an admissible key for XT.

Proof. The proof is immediate, but we include it here for convenience of the reader.

(1) Denote $B = TA$. Let $Y \in \mathbb{R}^{n\times d}$. Then

$$\beta_B(Y) = \beta_B(X) \Longleftrightarrow \downarrow(XB) = \downarrow(YB) \Longleftrightarrow \downarrow(XTA) = \downarrow(YTA) \Longleftrightarrow \beta_A(XT) = \beta_A(YT).$$

Thus, if $X \in \mathfrak{S}_n(A)$ and $Y' \in \mathbb{R}^{n\times d}$ so that $\beta_B(Y') = \beta_B(X')$ with $X' = XT^{-1}$, then $\beta_A(Y'T) = \beta_A(X)$. Therefore there exists $\Pi \in S_n$ so that $Y'T = \Pi X$. Thus $Y' \sim X'$. Hence $X' \in \mathfrak{S}_n(B)$. This shows $\mathfrak{S}_n(A)T^{-1} \subset \mathfrak{S}_n(TA)$. The reverse inclusion follows by replacing A with TA and T with T^{-1}. Together they prove (3.2).

(2) Let $Y \in \mathbb{R}^{n\times d}$ such that $\beta_{AL\Lambda}(X) = \beta_{AL\Lambda}(Y)$. For every $1 \leq j \leq D$ let $k \in [D]$ be so that $L_{jk} = 1$.

If $\Lambda_{kk} > 0$ then $\downarrow((XA)_j) = \downarrow((YA)_j)$.

If $\Lambda_{kk} < 0$ then $\downarrow(-(XA)_j) = \downarrow(-(YA)_j)$. But this implies also $\downarrow((XA)_j) = \downarrow((YA)_j)$ since $\downarrow(-z) = L_0\downarrow(z)$ where L_0 is the permutation matrix that has 1 on its main antidiagonal.

Either way, $\downarrow((XA)_j) = \downarrow((YA)_j)$. Hence $\downarrow(XA) = \downarrow(YA)$. Therefore $X \sim Y$ and thus $X \in \mathfrak{S}_n(AL\Lambda)$. This shows $\mathfrak{S}_n(A) \subset \mathfrak{S}_n(AL\Lambda)$. The reverse inclusion follows by a similar argument. Finally, notice $\{L\Lambda\}$ forms a group since $L^{-1}\Lambda L$ is also a diagonal matrix. This shows $\mathfrak{S}_n(A\Lambda L) = \mathfrak{S}(AL\Lambda')$ for some diagonal matrix Λ', and the conclusion (3.3) then follows.

(3) The relation (3.4) follows from noticing $\beta_{T^{-1}A}(Y) = \beta_A(YT)$. □

Relation (3.3) shows that, since A is assumed full rank, without loss of generality we can assume the first d columns are linearly independent. Let V denote the first d columns of A so that

$$A = V\begin{bmatrix} I & | & \tilde{A}\end{bmatrix} \tag{3.5}$$

where $\tilde{A} \in \mathbb{R}^{d\times(D-d)}$. The following result shows that, unsurprisingly, when $D = d > 1$, almost every matrix X is not separated by A. By Proposition 3.6 we can reduce the analysis to the case $A = I$ by a change of coordinates.

Proposition 3.7. *Assume $D = d > 1$, $n > 1$. Then*

1. *The set of data matrices not separated by I_d includes:*

$$\begin{aligned}\mathbb{B} := \{X \in \mathbb{R}^{n\times d}\ ,\ \exists i,j,k,l\ ,\ 1 \leq i < j \leq n, 1 \leq k < l \leq d\ \Rightarrow\ X_{i,k} \neq X_{j,k}\ \&\ X_{i,l} \neq X_{j,l}\} \subset \\ \subset \mathfrak{S}_n(I_d)^c.\end{aligned} \tag{3.6}$$

2. *The set $\mathbb{B}$ is generic with respect to the Zariski topology, i.e., open and dense. Specifically, its complement is the zero set of the polynomial*

$$P(X) = \sum_{1\leq i<j\leq n}\ \sum_{1\leq k<l\leq d}(X_{i,k} - X_{j,k})^2(X_{i,l} - X_{j,l})^2.$$

3. *For an invertible matrix* $A \in \mathbb{R}^{d\times d}$,

$$\mathbb{B} \cdot A^{-1} \subset \mathfrak{S}_n(A)^c.$$

Hence almost every matrix (w.r.t. Lebesgue measure) $X \in \mathbb{R}^{n\times d}$ *is* not *separated by* A.

Proof. (1) We need to show that any matrix X that on some columns k and l has distinct elements on same row positions is not separated by I_d. Indeed if X is such a matrix, let Y denote a copy of X except on those 4 entries where we set

$$Y_{i,k} = X_{j,k} \ , \ Y_{j,k} = X_{i,k} \ , \ Y_{i,l} = X_{i,l} \ , \ Y_{j,l} = X_{j,l}.$$

Note $X \nsim Y$ yet $\downarrow(X) = \downarrow(Y)$. Hence such matrices are not separated by I_d.

(2) By negation, the complement of $\mathbb{B}$ is given by

$$\mathbb{B}^c = \{X \in \mathbb{R}^{n\times d} \ , \ \forall i,j,k,l \ , \ 1 \le i < j \le n, 1 \le k < l \le d \ \& (X_{i,k} = X_{j,k} \ or \ X_{i,l} = X_{j,l})\}$$

This shows $\mathbb{B}^c$ is the zero set of polynomial P as claimed. Thus $\mathbb{B}^c$ is a closed Zariski set [10]. Its complement is generic with respect to the Zariski topology since $\mathbb{B}^c \neq \mathbb{R}^{n\times d}$.

(3) The inclusion is immediate. Density claim follows from this inclusion. □

On the other hand, extending the identity matrix by only one column produces an almost universal key:

Proposition 3.8. *Assume* $d \ge 2$ *and* $n \ge 3$.

Let $a \in \mathbb{R}^d$ *be a vector with non-zero entries, i.e.,* $\prod_{i=1}^d a_i \neq 0$. *Let* $A = \left[\begin{array}{c|c} I_d & a \end{array}\right] \in \mathbb{R}^{d\times(d+1)}$ *be a key. Then* $\mathfrak{S}_n(A)$ *is generic with respect to the Zariski topology (i.e., open and dense), however* $\mathfrak{S}_n(A) \neq \mathbb{R}^{n\times d}$. *In particular, its complement* $\mathfrak{S}_n(A)^c := \mathbb{R}^{n\times d} \setminus \mathfrak{S}_n(A)$ *is non-empty but has Lebesgue measure zero. Thus almost every matrix* $X \in \mathbb{R}^{n\times d}$ *is separated by* A.

Proof. First, we show that $\mathfrak{S}_n(A) \neq \mathbb{R}^{n\times d}$. Consider the matrices $X, Y \in \mathbb{R}^{n\times d}$ full of zeros except for the 3×2 top left corner, where:

$$X_{1,1} = Y_{1,1} = \frac{1}{a_1} \ , \ X_{2,1} = Y_{2,1} = -\frac{1}{a_1} \ , \ X_{3,1} = Y_{3,1} = 0,$$

$$X_{1,2} = Y_{3,2} = -\frac{1}{a_2} \ , \ X_{2,2} = Y_{1,2} = 0 \ , \ X_{3,2} = Y_{2,2} = \frac{1}{a_2}.$$

Clearly $\beta_A(X) = \beta_A(Y)$ (the two left columns and the last column contain 1, 0 repeated $n-2$ times and -1) and yet $X \nsim Y$.

Next, we show that $\mathfrak{S}_n(A)^c$ is included in a finite union of linear spaces each of positive codimension. This proves the claim.

To simplify notation, we introduce the following two operators. Let $\Pi, \Pi_0, \Pi_1, \cdots, \Pi_d \in \mathcal{S}_n$ denote permutation matrices of size n. For $X \in \mathbb{R}^{n\times d}$ denote by $x_1, \dots, x_d$ its columns. Thus $X = \left[x_1 | x_2 | \cdots | x_d\right]$.

$$L_{\Pi_0,\Pi_1,\dots,\Pi_d} : \mathbb{R}^{n\times d} \to \mathbb{R}^d \ , \ L_{\Pi_0,\Pi_1,\dots,\Pi_d} X = \Pi_0 X a - (a_1\Pi_1 x_1 + \cdots a_d \Pi_d x_d)$$

and

$$M_{\Pi,\Pi_1,\dots,\Pi_d} : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d} \ , \ M_{\Pi,\Pi_1,\dots,\Pi_d} X = \Pi X - \left[\begin{array}{c|c|c|c} \Pi_1 x_1 & \cdots & \Pi_d x_d \end{array}\right]$$

A matrix $X \in \mathbb{R}^{n\times d}$, $X = [x_1|\cdots|x_d]$ is not separated by $A = [I_d|a] \in \mathbb{R}^{d\times(d+1)}$, i.e., $X \in \mathfrak{S}_n(A)^c$ if there are permutation matrices $\Pi_1, \dots, \Pi_d \in \mathcal{S}_n$ such that the matrix $Y = [\Pi_1 x_1 | \cdots | \Pi_d x_d]$ satisfies:

$$Y \nsim X \ \ and \ \ \downarrow(X \cdot a) = \downarrow(Y \cdot a)$$

This is equivalent to say:

$$\exists \Pi_0 \in \mathcal{S}_n \ , \ \Pi_0 X a - (a_1\Pi_1 x_1 + \cdots + a_d \Pi_d x_d) = 0$$

$$\forall \Pi \in \mathcal{S}_n \exists k \in [d] \ , \ (\Pi - \Pi_k) x_k \neq 0$$

Hence

$$\mathfrak{S}_n(A)^c = \bigcup_{(\Pi_0,\Pi_1,\dots,\Pi_d) \in \mathcal{S}_n^{d+1}} \left(\ker L_{\Pi_0,\Pi_1,\dots,\Pi_d} \setminus \left(\bigcup_{\Pi \in \mathcal{S}_n} \ker M_{\Pi,\Pi_1,\dots,\Pi_d} \right) \right)$$

Let Δ denote the diagonal in $\mathcal{S}_n^{d+1}$,

$$\Delta = \{(\Pi_0, \Pi_1, \Pi_2, \dots, \Pi_d) \in \mathcal{S}_n^{d+1} \ , \ \Pi_1 = \Pi_2 = \cdots = \Pi_d\}$$

parametrized by the first two permutation matrices. For any $(\Pi_0, \Pi_1, \dots, \Pi_d) \in \Delta$, we have $\ker M_{\Pi_1,\Pi_1,\dots,\Pi_d} = \mathbb{R}^{n\times d}$. Thus

$$\ker\ L_{\Pi_0,\Pi_1,\dots,\Pi_d} \setminus \left(\bigcup_{\Pi\in S_n} \ker\ M_{\Pi,\Pi_1,\dots,\Pi_d} \right) = \emptyset$$

It follows:

$$\mathfrak{S}_n(A)^c = \bigcup_{(\Pi_0,\Pi_1,\dots,\Pi_d)\in S_n^{d+1}\setminus\Delta} \left(\ker\ L_{\Pi_0,\Pi_1,\dots,\Pi_d} \setminus \left(\bigcup_{\Pi\in S_n} \ker\ M_{\Pi,\Pi_1,\dots,\Pi_d} \right) \right)$$

Consider now $(\Pi_0,\Pi_1,\dots,\Pi_d)\in S_n^{d+1}\setminus\Delta$. Then

$$\ker\ L_{\Pi_0,\Pi_1,\dots,\Pi_d} = \ker\ L_{I,\Pi_0^{-1}\Pi_1,\dots,\Pi_0^{-1}\Pi_d}$$

Hence there is $k\in[d]$ so that $\Pi_0^{-1}\Pi_k \neq I$. Choose $x_k\in\mathbb{R}^n$ so that $(\Pi_0^{-1}\Pi_k)x_k \neq x_k$. Set $x_j = 0$ for $j\in[d]$, $j\neq k$ and consider the matrix $X = [x_1|\cdots|x_d]$. Then $L_{\Pi_0,\Pi_1,\dots,\Pi_d}X = a_k(\Pi_0-\Pi_k)x_k \neq 0$. This shows that $\ker\ L_{\Pi_0,\Pi_1,\dots,\Pi_d} \neq \mathbb{R}^{n\times d}$ and hence it is a subspace of positive codimension. We obtain:

$$\mathfrak{S}_n(A)^c \subset \bigcup_{(\Pi_0,\Pi_1,\dots,\Pi_d)\in S_n^{d+1}\setminus\Delta} \ker\ L_{\Pi_0,\Pi_1,\dots,\Pi_d}$$

This shows that $\mathfrak{S}_n(A)^c$ is included in a finite union of proper subspaces of $\mathbb{R}^{n\times d}$ which in turn is a closed set with respect to the Zariski topology of empty interior. This ends the proof of this result. □

The next result provides a characterization of the set $\mathfrak{S}_n(A)$. To do so we need to introduce additional notation that extends the operators $L_{\Pi_0,\dots,\Pi_d}$ and $M_{\Pi,\dots,\Pi_d}$ defined in the proof of Proposition 3.8. For $E_1,E_2,\dots,E_d\in\mathbb{R}^{n\times n}$ and $b\in\mathbb{R}^d$, with $b = \left(b_1,b_2,\cdots,b_d\right)^T$ define

$$L_{E_1,E_2,\dots,E_d;b} : \mathbb{R}^{n\times d}\to\mathbb{R}^n \ \ , \ \ X = \left[x_1|x_2|\cdots|x_d\right] \mapsto L_{E_1,E_2,\dots,E_d;b}(X) = b_1E_1x_1 + \cdots + b_dE_dx_d. \tag{3.7}$$

Proposition 3.9. *Fix $a_1,\dots,a_{D-d}\in\mathbb{R}^d$ and consider the key $A = \left[I_d|a_1|\cdots|a_{D-d}\right]\in\mathbb{R}^{d\times D}$. Let $X = \left[x_1|x_2|\cdots|x_d\right]$.*

1. *$X\in\mathfrak{S}_n(A)^c := \mathbb{R}^{n\times d}\setminus\mathfrak{S}_n(A)$ if and only if there are $\Pi_1,\Pi_2,\dots,\Pi_d,\Xi_1,\dots,\Xi_{D-d}\in S_n$ such that:*
   (a) *$\forall j\in[D-d]$, $\left[(\Xi_j-\Pi_1)x_1,\dots,(\Xi_j-\Pi_d)x_d\right]a_j = 0$*
   (b) *$\forall\Pi\in S_n\ \exists k\in[d]$ so that $(\Pi_k-\Pi)x_k\neq 0$.*
2. *The following hold true:*

$$\mathfrak{S}_n(A) = \bigcap_{\substack{\Pi_1,\dots,\Pi_d\in S_n\\ \Xi_1,\dots,\Xi_{D-d}\in S_n}} \left[\bigcup_{j=1}^{D-d} \left(\ker\ L_{\Xi_j-\Pi_1,\dots,\Xi_j-\Pi_d;a_j} \right)^c \bigcup \bigcup_{\Pi\in S_n} \bigcap_{k=1}^{d} \ker\ L_{\Pi_1-\Pi,\dots,\Pi_d-\Pi;\delta_k} \right] \tag{3.8}$$

and

$$\mathfrak{S}_n(A)^c = \bigcup_{\substack{\Pi_1,\dots,\Pi_d\in S_n\\ \Xi_1,\dots,\Xi_{D-d}\in S_n}} \left(\bigcap_{j=1}^{D-d} \ker\ L_{\Xi_j-\Pi_1,\dots,\Xi_j-\Pi_d;a_j} \right) \bigcap \left(\bigcup_{\Pi\in S_n} \bigcap_{k=1}^{d} \ker\ L_{\Pi_1-\Pi,\dots,\Pi_d-\Pi;\delta_k} \right)^c \tag{3.9}$$

where $\delta_k = (0,\dots,0,1,0,\dots,0)^T$ is the unit Kronecker sequence with 1 in the $k-th$ position.

Proof. The proof is a consequence of linear algebra analysis applied to map β_A.
(1) Assume X is not separated by A. Then there is $Y\in\mathbb{R}^{n\times d}$ so that $\beta_A(X) = \beta_A(Y)$ yet $X\nsim Y$.
Let $Y = [y_1|\cdots|y_d]$. Then $\beta_A(X) = \beta_A(Y)$ implies that there are permutation matrices $\Pi_1,\dots,\Pi_d,\Xi_1,\dots,\Xi_{D-d}\in S_n$ so that:

$$y_1 = \Pi_1x_1,\dots,y_d = \Pi_dx_d, Ya_1 = \Xi_1Xa_1,\dots,Ya_{D-d} = \Xi_{D-d}Xa_{D-d}$$

Substituting the expressions for $y_1,\dots,y_d$ provided by the first d equations into the latter $D-d$ equations, we obtain part 1.(a).
For same Y, the condition $X\nsim Y$ implies that for every $\Pi\in S_n$, $Y-\Pi X\neq 0$. Thus part 1(b) is proved.
(2 & 3) Equation (3.9) is a transcription of part (1). Equation (3.8) follows from (3.9) by taking the complement. □

3.2. Construction of universal keys

In this subsection we construct universal keys. Proposition 3.9 provides us with an algorithm to check whether a key A is universal. Unfortunately the algorithm has an exponential complexity in data size.

If the key $A \in \mathbb{R}^{d\times D}$ is universal then A must have full rank. Therefore there are permutation matrix $L \in S_D$ and invertible $T \in GL(d,\mathbb{R})$ so that $A = T\left[I_d \ \tilde{A}\right] L$, with $\tilde{A} \in \mathbb{R}^{d\times(D-d)}$. Proposition 3.6 shows that A is a universal key if and only if $\left[I_d \ \tilde{A}\right]$ is a universal key. This observation allows us to prove the main result of this subsection stated earlier as part b of Theorem 2.1. Recall a set of vectors $\{f_1,\dots,f_m\}$ in a linear space V of finite dimension $n \le m$ is called a *full spark frame* if any subset of n vectors is linearly independent. See [1,30] for more information and explicit constructions of full spark frames.

Theorem 3.10. *Consider the metric space $(\widehat{\mathbb{R}^{n\times d}}, \mathbf{d})$. Set $D = 1 + (d-1)n!$ and let $A \in \mathbb{R}^{d\times D}$ be a matrix whose columns form a full spark frame, i.e., any subset of d columns is linearly independent. Then the key A is universal and the induced map $\hat{\beta}_A : \widehat{\mathbb{R}^{n\times d}} \to \mathbb{R}^{n\times D}$, $X \mapsto \beta_A(X) = \downarrow(XA)$ is injective. Furthermore, $\hat{\beta}_A$ is bi-Lipschitz, with estimates of the bi-Lipschitz constants $a_0 = \min_{J\subset[D],|J|=d} s_d(A[J])$ and $b_0 = s_1(A)$, where $s_1(A)$ denotes the largest singular value of A, $A[J]$ denotes the submatrix of A formed by columns indexed by J, and $s_d(A[J])$ denotes the d^{th} singular value (in this case, the smallest) of $A[J]$. Specifically, for any $X, Y \in \mathbb{R}^{n\times d}$,*

$$a_0 \cdot \mathbf{d}(\hat{X},\hat{Y}) \le \|\beta_A(X) - \beta_A(Y)\| \le b_0 \cdot \mathbf{d}(\hat{X},\hat{Y}) \tag{3.10}$$

where all norms are Frobenius norms.

Proof. Let $a_1,\dots,a_D$ denote the columns of A, so that $A = [a_1|\cdots|a_D]$.

Fix $X, Y \in \mathbb{R}^{n\times d}$ two matrices. Then there are permutation matrices $P_0, \Pi_1,\dots,\Pi_D, \Xi_1,\dots,\Xi_D \in S_n$ so that $\mathbf{d}(\hat{X},\hat{Y}) = \|P_0X - Y\|$ and

$$\beta_A(X) = \left[\ \Pi_1 X a_1 \ \ | \ \ \cdots \ \ | \ \ \Pi_D X a_D\ \right],\ \beta_A(Y) = \left[\ \Xi_1 Y a_1 \ \ | \ \ \cdots \ \ | \ \ \Xi_D Y a_D\ \right].$$

Thus

$$\|\beta_A(X) - \beta_A(Y)\|^2 = \sum_{k=1}^{D} \|(\Pi_k X - \Xi_k Y)a_k\|_2^2 = \sum_{k=1}^{D} \|(\Xi_k^T \Pi_k X - Y)a_k\|_2^2 \tag{3.11}$$

Permutations Π_k and Ξ_k satisfy the optimality condition: $\|\Pi_k X a_k - \Xi_k Y a_k\|_2 = \min_{P\in S_n} \|PXa_k - Ya_k\|_2$. Hence $\|\Pi_k X a_k - \Xi_k Y a_k\|_2 \le \|P_0 X a_k - Y a_k\|_2$. Therefore:

$$\|\beta_A(X) - \beta_A(Y)\|^2 \le \sum_{k=1}^{D} \|P_0 X a_k - Y a_k\|_2^2 = \|(PX - Y)A\|^2 \le s_1(A)^2 \|P_0 X - Y\|^2, \tag{3.12}$$

leading to the upper bound in (3.10).

The lower bound in (3.10) follows from the pigeonhole principle similar to the one employed in the proof of Theorem 2.1. In equation (3.11) there are $D = 1 + (d-1)n!$ terms. Since only $n!$ permutations are distinct, there is a permutation Q that repeats at least d times. Say $J = \{j_1, j_2, \dots, j_d\} \subset [D]$ is a set of indices so that $\Xi_{j_1}^T \Pi_{j_1} = \cdots = \Xi_{j_d}^T \Pi_{j_d} = Q$. Then

$$\begin{aligned}\|\beta_A(X) - \beta_A(Y)\|^2 &\ge \sum_{k=1}^{d} \|(\Xi_{j_k}^T \Pi_{j_k} X - Y)a_{j_k}\|_2^2 = \|(QX - Y)A[J]\|^2 \\ &\ge s_d(A[J])^2 \|QX - Y\|^2 \ge s_d(A[J])^2 \|P_0 X - Y\|^2 \ge a_0^2 \mathbf{d}(\hat{X},\hat{Y})^2.\end{aligned}$$

The lower bound in (3.10) implies that $\hat{\beta}_A : \widehat{\mathbb{R}^{n\times d}} \to \mathbb{R}^{n\times D}$ is injective and hence A is a universal key. This ends the proof of Theorem 3.10. □

Remark 3.11. The size of encoder β_A grows exponential in n (specifically, grows linearly in $n!$). Theorem 3.10 provides only a sufficient condition for key A to be universal. In fact, the recent work in [16] shows that, generically, it is sufficient to take $D = 2nd + 1$.

3.3. Bi-Lipschitz properties of universal keys

In this subsection we prove that any universal key defines a bi-Lipschitz encoding map, regardless of D.

Theorem 3.12. *Assume the key $A \in \mathbb{R}^{d\times D}$ is universal, i.e., the induced map $\hat{\beta}_A : \widehat{\mathbb{R}^{n\times d}} \to \mathbb{R}^{n\times D}$, $X \mapsto \beta_A(X) = \downarrow(XA)$ is injective. Then $\hat{\beta}_A$ is bi-Lipschitz, that is, there are constants $a_0 > 0$ and $b_0 > 0$ so that for all $X, Y \in \mathbb{R}^{n\times d}$,*

$$a_0 \cdot \mathbf{d}(\hat{X},\hat{Y}) \le \|\beta_A(X) - \beta_A(Y)\| \le b_0 \cdot \mathbf{d}(\hat{X},\hat{Y}) \tag{3.13}$$

where all are Frobenius norms. Furthermore, an estimate for b_0 is provided by the largest singular value of A, $b_0 = s_1(A)$.

Proof. The upper bound in (3.13) follows as in the proof of Theorem 3.10, from equations (3.11) and (3.12). Notice that no property is assumed in order to obtain the upper Lipschitz bound.

The lower bound in (3.13) is more difficult. It is shown by contradiction following the strategy utilized in the Complex Phase Retrieval problem [6].

Assume $\inf_{X\not\sim Y} \frac{\|\beta_A(X)-\beta_A(Y)\|_2^2}{\mathbf{d}(\hat{X},\hat{Y})^2} = 0$.

Step 1: Reduction to local analysis. Since $\mathbf{d}(t\hat{X}, t\hat{Y}) = t\,\mathbf{d}(\hat{X},\hat{Y})$ for all $t>0$, the quotient $\frac{\|\beta_A(X)-\beta_A(Y)\|_2}{\mathbf{d}(\hat{X},\hat{Y})}$ is scale invariant. Therefore, there are sequences $(X^t)_t, (Y^t)_t$ with $\|Y^t\| \le \|X^t\| = 1$ and $\mathbf{d}(\hat{X}^t,\hat{Y}^t) > 0$ so that $\lim_{t\to\infty} \frac{\|\beta_A(X^t)-\beta_A(Y^t)\|_2}{\mathbf{d}(\hat{X}^t,\hat{Y}^t)} = 0$. By compactness of the closed unit ball, one can extract convergence subsequences. To ease notation, assume $(X^t)_t, (Y^t)_t$ are these subsequences. Let $X^\infty = \lim_t X^t$ and $Y^\infty = \lim_t Y^t$ denote their limits. Notice $\lim_t \|\beta_A(X^t) - \beta_A(Y^t)\|_2 = 0$. This implies $\|\beta_A(X^\infty) - \beta_A(Y^\infty)\| = 0$ and thus $\beta_A(X^\infty) = \beta_A(Y^\infty)$. Since $\widehat{\beta_A}$ is assumed injective, it follows that $\widehat{X^\infty} = \widehat{Y^\infty}$.

This means that, if the lower Lipschitz bound vanishes, then this is achieved by vanishing of a local lower Lipschitz bound. To follow the terminology in [6], the type I local lower Lipschitz bound vanishes at some $Z_0 \in \mathbb{R}^{n\times d}$, with $\|Z_0\| = 1$:

$$A(Z_0) := \lim_{r\to 0} \inf_{\substack{\hat{X}\neq\hat{Y} \\ \mathbf{d}(\hat{X},\hat{Z}_0)<r \\ \mathbf{d}(\hat{Y},\hat{Z}_0)<r}} \frac{\|\beta_A(X)-\beta_A(Y)\|_2^2}{\mathbf{d}(\hat{X},\hat{Y})^2} = 0. \tag{3.14}$$

Note that, in general, the infimum of the type I local lower Lipschitz bound over the unit sphere may be strictly larger than the global lower Lipschitz bound (see Theorems 2.1 and Theorem 2.2 in [6] and Theorem 4.3 in [5]). The compactness argument forces the local lower Lipschitz bound to vanish when the global lower bound vanishes.

Step 2. Local Linearization. The following stability subgroups of S_n play an important role:

$$G = \{P\in S_n \ : \ PZ_0 = Z_0\} \ , \ \ H_j = \{P\in S_n \ : \ PZ_0a_j = Z_0a_j\} \ , \ 1\le j\le D.$$

Obviously $I_n \in G \subset H_j \subset S_n$, for every $j\in[D]$. The group G is the stabilizer of Z_0, whereas H_j is the stabilizer of Z_0a_j. Let $\delta_0 = \min_{P\in S_n\setminus G} \|(I_n-P)Z_0\|$ denote the smallest variation of Z_0 under row permutations. Note $\delta_0 > 0$ by the definition of G.

Consider $X = Z_0 + U$ and $Y = Z_0 + V$ where $U, V \in \mathbb{R}^{n\times d}$ are "aligned" in the sense that $d(\hat{X},\hat{Y}) = \|U - V\|$. This property requires that $\|U-V\| \le \|PX - Y\|$, for every $P\in S_n$. The next result replaces equivalently this condition by requirements involving (U,V) and the group G only.

Lemma 3.13. *Assume $\|U\|, \|V\| < \frac{1}{4}\delta_0$, where $\delta_0 = \min_{P\in S_n\setminus G}\|(I_n-P)Z_0\|$. Let $X = Z_0+U$, $Y = Z_0+V$. Then:*

1. $\mathbf{d}(\hat{X},\widehat{Z_0}) = \|U\|$ *and* $\mathbf{d}(\hat{Y},\widehat{Z_0}) = \|V\|$.
2. $\mathbf{d}(\hat{X},\hat{Y}) = \min_{P\in G}\|U-PV\| = \min_{P\in G}\|PU-V\|$
3. *The following are equivalent:*
   (a) $\mathbf{d}(\hat{X},\hat{Y}) = \|U-V\|$.
   (b) *For every $P\in G$,* $\|U-V\| \le \|PU-V\|$.
   (c) *For every $P\in G$,* $\langle U,V\rangle \ge \langle PU, V\rangle$.

Proof of Lemma 3.13. (1) Note that is $U=0$ then the claim follows. Assume $U\neq 0$. Then

$$\mathbf{d}(\hat{X},\widehat{Z_0}) = \min_{P\in S_n}\|X - PZ_0\| = \min_{P\in S_n}\|(I_n-P)Z_0 + U\| \le \|U\|$$

On the other hand, assume the minimum is achieved for a permutation $P_0\in S_n$. If $P_0\in G$ then $\mathbf{d}(\hat{X},\widehat{Z_0}) = \|(I_n-P_0)Z_0+U\| = \|U\|$. If $P_0\notin G$ then

$$\mathbf{d}(\hat{X},\widehat{Z_0}) \ge \|(I_n-P_0)Z_0\| - \|U\| > \frac{3\delta_0}{4} > \|U\| \ge \mathbf{d}(\hat{X},\widehat{Z_0})$$

which yields a contradiction. Hence $\mathbf{d}(\hat{X},\widehat{Z_0}) = \|U\|$. Similarly, one shows $\mathbf{d}(\hat{X},\widehat{Z_0}) = \|V\|$.

(2) Clearly

$$\mathbf{d}(\hat{X},\hat{Y}) = \min_{P\in S_n}\|PX-Y\| \le \min_{P\in G}\|PX-Y\| = \min_{P\in G}\|PU-V\|$$

On the other hand, for $P\in S_n\setminus G$ and $Q\in G$,

$$\|PX-Y\| = \|(P-I_n)Z_0 + PU - V\| \ge \|(I_n-P)Z_0\| - \|U\| - \|V\| \ge$$
$$\ge \delta_0 - 2\|U\| - 2\|V\| + \|QU-V\| \ge \min_{Q\in G}\|QU-V\| \ge \mathbf{d}(\hat{X},\hat{Y}).$$

(3)
(a)⇒(b).

If $\mathbf{d}(\hat{X},\hat{Y}) = \|U - V\|$ then

$$\|U - V\| \le \|PX - Y\| = \|(P - I_n)Z_0 + PU - V\| \ , \ \forall P \in S_n.$$

In particular, for $P \in G$, $(P - I_n)Z_0 = 0$ and the above inequality reduces to (b).

(b)⇒(a).

Assume (b). For $P \in G$,

$$\|U - V\| = \|X - Y\| \le \|PU - V\| = \|PX - Y\|.$$

For $P \in S_n \setminus G$,

$$\|PX - Y\| = \|(P - I_n)Z_0 + PU - V\| \ge \|(I_n - P)Z_0\| - \|U\| - \|V\| \ge$$

$$\ge \delta_0 - 2\|U\| - 2\|V\| + \|U - V\| \ge \|U - V\| = \|X - Y\|.$$

This shows $\mathbf{d}(\hat{X},\hat{Y}) = \|X - Y\| = \|U - V\|$.

(b)⟺(c). This is immediate after squaring (b) and simplifying the terms. □

Consider now sequences $(\hat{X}^t)_t, (\hat{Y}^t)_t$ that converge to $\hat{Z}_0$ and achieve lower bound 0 as in (3.14). Choose representatives X_t and Y_t in their equivalence classes that satisfy the hypothesis of Lemma 3.13 so that $X_t = Z_0 + U_t$, $Y_t = Z_0 + V_t$, $\|U_t\|, \|V_y\| < \frac{1}{4}\delta_0$, $\mathbf{d}(\hat{X}_t, \hat{Z}_0) = \|U_t\|$, $\mathbf{d}(\hat{Y}_t, \hat{Z}_0) = \|V_t\|$ and $\mathbf{d}(\hat{X}_t, \hat{Y}_t) = \|U_t - V_t\| > 0$. With $A = [a_1|\cdots|a_D]$ we obtain:

$$\|\beta_A(X_t) - \beta_A(Y_t)\|_2^2 = \sum_{j=1}^{D} \| \downarrow (X_t a_j) - \downarrow (Y_t a_j)\|_2^2 = \sum_{j=1}^{D} \|(Z_0 + U_t)a_j - \Pi_{j,t}(Z_0 + V_t)a_j\|_2^2,$$

for some $\Pi_{j,t} \in S_n$. In fact $\Pi_{j,t} \in argmin_{\Pi \in H_j} \|U_t - \Pi V_t)a_j\|_2$. Pass to sub-sequences (that will be indexed by t to ease notation) so that $\Pi_{j,t} = \Pi_j$ for some $\Pi_j \in S_n$. Thus

$$\|\beta_A(X_t) - \beta_A(Y_t)\|_2^2 = \sum_{j=1}^{D} \|(I_n - \Pi_j)Z_0 a_j + (U_t - \Pi_j V_t)a_j\|_2^2.$$

Since the above sequence must converge to 0 as $t \to \infty$, while $U_t, V_t \to 0$, it follows that necessarily $\Pi_j \in H_j$ and the expressions simplify to

$$\|\beta_A(X_t) - \beta_A(Y_t)\|_2^2 = \sum_{j=1}^{D} \|(U_t - \Pi_j V_t)a_j\|_2^2.$$

Thus equation (3.14) implies that for every $j \in [D]$,

$$\lim_{t\to\infty} \frac{\|(U_t - \Pi_j V_t)a_j\|_2^2}{\|U_t - V_t\|^2} = 0, \tag{3.15}$$

where $\Pi_j \in H_j$, $\|U_t\|, \|V_t\| \to 0$, and U_t, V_t are aligned so that $\langle U_t, V_t\rangle \ge \langle PU_t, V_t\rangle$ for every $P \in G$. Equivalently, relation (3.14) can be restated as:

$$0 = \inf_{\substack{U,V \in \mathbb{R}^{n\times d} \\ s.t. \\ U \ne V \\ \langle U,V\rangle \ge \langle PU,V\rangle, \forall P \in G}} \frac{\sum_{j=1}^{D} \|(U - \Pi_j V)a_j\|_2^2}{\|U - V\|^2}, \tag{3.16}$$

for some permutations $\Pi_j \in H_j$, $j \in [D]$. By Lemma 3.13 the constraint in the optimization problem above implies $\|U - V\| = \min_{P\in G} \|U - PV\|$. Hence (3.16) implies:

$$0 = \inf_{\substack{U,V \in \mathbb{R}^{n\times d} \\ s.t. \\ U \ne PV, \forall P \in G}} \max_{P\in G} \frac{\sum_{j=1}^{D} \|(U - \Pi_j V)a_j\|_2^2}{\|U - PV\|^2}, \tag{3.17}$$

for some permutation matrices Π_j's. While the above optimization problem seems a relaxation of (3.16), in fact (3.17) implies (3.16) with a possibly change of permutation matrices Π_j, but remaining still in H_j.

Step 3. Existence of a Minimizer.

The optimization problem (3.16) is a Quadratically Constrained Ratio of Quadratics (QCRQ) optimization problem. A significant number of papers have been published on this topic [7,8]. In particular, [3] presents a formal setup for analysis of QCRQ problems.

Our interest is to utilize some of these techniques in order to establish the existence of a minimizer for (3.16) or (3.17). Specifically we show:

Lemma 3.14. *Assume the key A has linearly independent rows (equivalently, the columns of A form a frame for $\mathbb{R}^d$) and the lower Lipschitz bound of $\hat{\beta}_A$ is 0. Then there are $\tilde{U}, \tilde{V} \in \mathbb{R}^{n\times d}$ so that:*

1. *$\tilde{U} \neq P\tilde{V}$, for every $P \in G$;*
2. *For every $j \in [D]$, $(\tilde{U} - \Pi_j \tilde{V})a_j = 0$.*

Proof of Lemma 3.14. We start with the formulation (3.17). Therefore there are sequences $(U_t, V_t)_{t\geq 1}$ so that $U_t \neq PV_t$ for any $P \in G, t \geq 1$, and yet for any $P \in G$,

$$\lim_{t\to\infty} \frac{\sum_{j=1}^D \|(U_t - \Pi_j V_t)a_j\|_2^2}{\|U_t - PV_t\|^2} = 0.$$

Let $E = \{(U,V) \in \mathbb{R}^{n\times d} \times \mathbb{R}^{n\times d} \ , \ (U - \Pi_j)V)a_j = 0 \ , \ \forall j \in [D]\}$ denote the null space of the linear operator

$$T : \mathbb{R}^{n\times d} \times \mathbb{R}^{n\times d} \to \mathbb{R}^D \ , \ (U,V) \mapsto \left[\ (U - \Pi_1 V)a_1 \quad | \quad \cdots \quad | \quad (U - \Pi_D V)a_D \ \right],$$

associated to the numerator of the above quotient. Let $F_P = \{(U,V) \in \mathbb{R}^{n\times d} \times \mathbb{R}^{n\times d} \ , \ U - PV = 0\}$ be the null space of the linear operator

$$R_P : \mathbb{R}^{n\times d} \times \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d} \ , \ (U,V) \mapsto U - PV.$$

A consequence of (3.17) is that for every $P \in G$, $E \setminus F_P \neq \emptyset$. In particular, $F_p \cap E$ is a subspace of E of positive codimension. Using the Baire category theorem (or more elementary linear algebra arguments), we conclude that

$$E \setminus \left(\cup_{P\in G} F_P\right) \neq \emptyset.$$

Let $(\tilde{U}, \tilde{V}) \in E \setminus \left(\cup_{P\in G} F_P\right)$. This pair satisfies the conclusions of Lemma 3.14. □

Step 4. Contradiction with the universality property of the key.

So far we obtained that if the lower Lipschitz bound of $\hat{\beta}_A$ vanishes than there are $Z_0, \tilde{U}, \tilde{V} \in \mathbb{R}^{n\times d}$ with $Z_0 \neq 0$ and $\tilde{U} \neq P\tilde{V}$, for all $P \in G$ that satisfy the conclusions of Lemma 3.14. Notice $\langle Z_0, Z_0\rangle = \langle PZ_0, Z_0\rangle$ for all $P \in G$ and $(Z_0 - \Pi_j Z_0)a_j = 0$ for all $j \in [D]$. Choose $s > 0$ but small enough so that $s\|\tilde{U}\|, s\|\tilde{V}\| < \frac{1}{4}\delta_0$ with $\delta_0 = \min_{P\in S_n\setminus G} \|(I_n - P)Z_0\|$. Let $X = Z_0 + s\tilde{U}$ and $Y = Z_0 + s\tilde{V}$. Then Lemma 3.13 implies $\mathbf{d}(\hat{X}, \hat{Y}) = \min_{P\in G} \|\tilde{U} - P\tilde{V}\| > 0$. Hence $\hat{X} \neq \hat{Y}$. On the other hand, for every $j \in [D]$, $Xa_j = \Pi_j Ya_j$. Thus $\hat{\beta}_A(\hat{X}) = \hat{\beta}_A(\hat{Y})$, contradicting the assumption that $\hat{\beta}_A$ is injective. □

3.4. Dimension reduction

Theorem 3.10 provides an Euclidean bi-Lipschitz embedding of very high dimension, $D = 1 + (d-1)n!$. On the other hand, Theorem 3.12 shows that any universal key $A \in \mathbb{R}^{d\times D}$ for $\widehat{\mathbb{R}^{n\times d}}$, and hence any injective map $\hat{\beta}_A$ is bi-Lipschitz. In this subsection we show that any bi-Lipschitz Euclidean embedding $\hat{\beta}_A : \widehat{\mathbb{R}^{n\times d}} \to \mathbb{R}^{n\times D}$ with $D > 2d$ can be further compressed to a smaller dimension space $\mathbb{R}^m$ with $m = 2nd$ thus yielding bi-Lipschitz Euclidean embeddings of redundancy 2. This is shown in the next result.

Theorem 3.15. *Assume $A \in \mathbb{R}^{d\times D}$ is a universal key for $\widehat{\mathbb{R}^{n\times d}}$ with $D \geq 2d$. Then, for $m \geq 2nd$, a generic linear operator $B : \mathbb{R}^{n\times D} \to \mathbb{R}^m$ with respect to Zariski topology on $\mathbb{R}^{n\times D\times m}$, the map*

$$\hat{\beta}_{A,B} : \widehat{\mathbb{R}^{n\times d}} \to \mathbb{R}^{2nd} \ , \ \hat{\beta}_{A,B}(\hat{X}) = B\left(\hat{\beta}_A(\hat{X})\right) \tag{3.18}$$

is bi-Lipschitz. In particular, almost every full-rank linear operator $B : \mathbb{R}^{n\times D} \to \mathbb{R}^{2nd}$ produces such a bi-Lipschitz map.

Remark 3.16. The proof shows that, in fact, the complement set of linear operators B that produce bi-Lipschitz embeddings is included in the zero-set of a polynomial.

Remark 3.17. Putting together Theorems 3.10, 3.12, 3.15 we obtain that the metric space $\widehat{\mathbb{R}^{n\times d}}$ admits a global bi-Lipschitz embedding in the Euclidean space $\mathbb{R}^{2nd}$. This result seems analogous to a conclusion from a Whitney embedding theorem (see §1.3 in [22]) with the important caveat that the Whitney embedding result applies to smooth manifolds, whereas here $\widehat{\mathbb{R}^{n\times d}}$ is merely a non-smooth algebraic variety.

Remark 3.18. These three theorems are summarized in part two of Theorem 2.1 presented in the first section.

Remark 3.19. While the embedding dimension grows linearly in nd, in fact $m = 2nd$, the computational complexity of constructing $\hat{\beta}_{A,B}$ is NP due to the $1 + (d-1)n!$ intermediary dimension. The target dimension of $\hat{\beta}_{A,B}$ is $m = 2nd$. However, to construct $\hat{\beta}_{A,B}(X)$ we need to construct $\hat{\beta}_A(X)$ which is of dimension $n \times D$. This intermediary dimension may be very large, if the smallest universal key has $D = 1 + (d-1)n!$ columns.

Remark 3.20. As the proofs show, for $D \geq 1 + (d-1)n!$, a generic (A, B) with respect to Zariski topology, $A \in \mathbb{R}^{d \times D}$ and linear map $B : \mathbb{R}^{n \times D} \to \mathbb{R}^{2nd}$, our result produces a bi-Lipschitz embedding $(\hat{\beta}_{A,B}, \mathbf{d})$ of $\widehat{\mathbb{R}^{n \times d}}$ into $(\mathbb{R}^{2nd}, \|\cdot\|_2)$.

Proof of Theorem 3.15. The proof follows a similar argument as the one used in Theorem 3 of [23]. See also [15].

Without loss of generality, assume $m < nD$.

Notice $\beta_A : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times D}$ is already homogeneous of degree 1 (with respect to positive scalars). Let $\Delta : \mathbb{R}^{n \times d} \times \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times D}$ be defined by $\Delta(X, Y) = \beta_A(X) - \beta_A(Y)$. Denote $E = Ran(\Delta) = \{\beta_A(X) - \beta_A(Y)\ ,\ X, Y \in \mathbb{R}^{n \times d}\}$.

Recall that the key A has columns $(a_k)_{k \in [D]}$, so that $A = [a_1 | \cdots | a_D]$. Notice that

$$\Delta(X, Y) = \left[P_1 X a_1 - Q_1 Y a_1 | \cdots | P_D X a_D - Q_D Y a_D\right]$$

for some $P_1, \ldots, P_D, Q_1, \ldots, Q_D \in S_n$, so that for each $k \in [D]$, P_k, Q_k are permutations producing the decreasing rearrangements of vectors $X a_k$ and $Y a_k$, respectively. In particular,

$$E \subset F := \bigcup_{\gamma \in (S_n)^{2D}} F_\gamma \ , \ F_\gamma := Ran(L_\gamma),$$

where the $(n!)^{2D}$ linear operators $L_\gamma : \mathbb{R}^{n \times d} \times \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times D}$, are defined by

$$L_\gamma(X, Y) = \left[P_1 X a_1 - Q_1 Y a_1 | \cdots | P_D X a_D - Q_D Y a_D\right]$$

when $\gamma = (P_1, \ldots, P_D, Q_1, \ldots, Q_D) \in (S_n)^{2D}$.

Claim: We claim that, for $m \geq 2nd$ and a generic linear operator $B : \mathbb{R}^{n \times D} \to \mathbb{R}^m$, we have $\ker(B) \cap F = \{0\}$. Such a generic linear operator has kernel of dimension $\dim(\ker(B)) = nD - m \leq n(D - 2d)$. It is therefore sufficient to show that, for a generic subspace $V \subset \mathbb{R}^{n \times D}$ of dimension $r \leq n(D-2d)$, for every $\gamma \in (S_n)^{2D}$, $V \cap F_\gamma = \{0\}$. This last claim follows from the observation $\dim(F_\gamma) \leq 2nd$.

We now show how this claim proves the Theorem. Let B be such a linear map, and let $\beta_{A,B} : \mathbb{R}^{n \times d} \to \mathbb{R}^m$ be the map $\beta_{A,B}(X) = B(\downarrow (XA))$. Then $\beta_{A,B}(X) = \beta_{A,B}(Y)$ implies $\Delta(X, Y) = \beta_A(X) - \beta_A(Y) \in \ker(B)$. Thus $\Delta(X, Y) = 0$ which implies $\beta_A(X) = \beta_A(Y)$. Since $\hat{\beta}_A$ is injective on $\widehat{\mathbb{R}^{n \times d}}$ it follows $\hat{X} = \hat{Y}$. Thus $\hat{\beta}_{A,B}$ is injective. On the other hand, for each $\gamma = (P_1, \ldots, P_D, Q_1, \ldots, Q_D) \in S_n^{2D}$, the restriction of B to the linear space $Ran(L_\gamma)$ is injective, and thus bounded below as a linear map: there is $a_\gamma > 0$ so that for every $X, Y \in \mathbb{R}^{n \times d}$, $\|B(L_\gamma(X, Y))\| \geq a_\gamma \|L_\gamma(X, Y)\|$. Let $a_\infty = \min_\gamma a_\gamma > 0$. Thus

$$\|\beta_{A,B}(X) - \beta_{A,B}(Y)\| = \|B(L_{\gamma_0}(X, Y))\| \geq a_\infty \|L_{\gamma_0}(X, Y)\| = a_\infty \|\beta_A(X) - \beta_A(Y)\|,$$

where $\gamma_0 \in (S_n)^{2D}$ is a particular $2D$-tuple of permutations. This shows that $B|_{\beta_A(\mathbb{R}^{n \times d})} : \beta_A(\mathbb{R}^{n \times d}) \to \mathbb{R}^m$ is bi-Lipschitz. By Theorem 3.12, the map $\hat{\beta}_A$ is bi-Lipschitz. Therefore we get $\hat{\beta}_{A,B}$ is bi-Lipschitz as well. □

3.5. Proof of Corollary 1.3

(1) It is clear that any continuous f induces a continuous $\varphi : \beta(\mathbb{R}^{n \times d}) \to \mathbb{R}$ via $\varphi(\beta(X)) = f(X)$. Furthermore, $F := \beta(\mathbb{R}^{n \times d}) = \hat{\beta}(\widehat{\mathbb{R}^{n \times d}})$ is a closed subset of $\mathbb{R}^m$ since $\hat{\beta}$ is bi-Lipschitz. Then a consequence of the Tietze extension theorem (see problem 8 in §12.1 of [40]) implies that φ admits a continuous extension $g : \mathbb{R}^m \to \mathbb{R}$. Thus $g(\beta(X)) = f(X)$ for all $X \in \mathbb{R}^{n \times d}$. The converse is trivial.

(2) As in part (1), the Lipschitz continuous function f induces a Lipschitz continuous function $\varphi : F \to \mathbb{R}$. Since $F \subset \mathbb{R}^m$ is a subset of a Hilbert space, by Kirszbraun extension theorem (see [45]), φ admits a Lipschitz continuous extension (even with the same Lipschitz constant!) $g : \mathbb{R}^m \to \mathbb{R}$ so that $g(\beta(X)) = f(X)$ for every $X \in \mathbb{R}^{n \times d}$. The converse is trivial. □

4. Applications to graph deep learning

In this section we take an empirical look at the permutation invariant mappings presented in this paper. We focus on the problems of *graph classification*, for which we employ the `PROTEINS_FULL` dataset [14], and of *graph regression*, for which we employ the quantum chemistry `QM9` dataset [39]. In both problems we want to estimate a function $F : (A, Z) \to p$, where (A, Z) characterizes a vertex-decorated graph, with $A \in \mathbb{R}^{n \times n}$ an adjacency matrix and $Z \in \mathbb{R}^{n \times r}$ an associated feature matrix, the i^{th} row encodes an array of r features annotating the i^{th} node. p is a scalar output where we have $p \in \{0, 1\}$ for binary classification and $p \in \mathbb{R}$ for regression.

We estimate F using a deep network that is trained in a supervised manner. The network is comprised of three successive components applied in series: Γ, ϕ, and η. Γ represents a graph deep network [26], which produces a set of embeddings $X \in \mathbb{R}^{N \times d}$ across the nodes in the graph. Here $N \geq n$ is chosen to accommodate the graph with the largest number of nodes. In this case, the last $N - n$ rows of Y are filled with 0's. $\phi : \mathbb{R}^{N \times d} \to \mathbb{R}^m$ represents a permutation invariant mapping such as those proposed in this paper. $\eta : \mathbb{R}^m \to \mathbb{R}$ is a fully connected neural network. The entire end-to-end network is shown in Fig. 1.

Fig. 1. The processing pipeline of the graph deep learning tasks.

In this paper, we model Γ using a Graph Convolutional Network (GCN) outlined in [26]. Let $\mathbf{D} \in \mathbb{R}^{n\times n}$ be the associated degree matrix for our graph $\mathcal{G}$. Also let $\tilde{A}$ be the associated adjacency matrix of $\mathcal{G}$ with added self connection: $\tilde{A} = I + A$, where I is the $n \times n$ identity matrix, and $\tilde{\mathbf{D}} = \mathbf{D} + I$. Finally, we define the modified adjacency matrix $\hat{A} = \tilde{\mathbf{D}}^{-1/2}\tilde{A}\tilde{\mathbf{D}}^{-1/2}$. A GCN layer is defined as $H^{(l+1)} = \sigma(\hat{A}H^{(l-1)}W^{(l)})$. Here $H^{(l-1)}$ represents the GCN state coming into the l^{th} layer, σ represents a chosen nonlinear element-by-element operation such as ReLU, and $W^{(l)}$ represents a matrix of trainable weights assigned to the l^{th} layer whose number of rows matches the number of columns in H^l and number of columns is set to the size of the embeddings at the (l+1)'th layer. The initial state $H^{(0)}$ of the network is set to the feature set of the nodes of the graph $H^{(0)} = Z$.

For ϕ we employ seven (7) different methods that are described next.

1. `ordering`: For the ordering method, we set $D = d+1$, $\phi_{ordering}(X) = \beta_A(X) = \downarrow(XA)$ with $A = [I\ 1]$ the identity matrix bordered by a column of ones. The ordering and identity-based mappings have the notable disadvantage of not producing the same output embedding size for different sized graphs. To accommodate this and have consistently sized inputs for η, we choose to zero-pad $\phi(X)$ for these methods to produce a vector in $\mathbb{R}^m$, where $m = ND = N(d+1)$ and N is the size of the largest graph in the dataset.
2. `kernels`: For the kernels method,

$$(\phi_{kernel}(X))_j = \sum_{k=1}^{n} K_G(x_k, a_j) = \sum_{k=1}^{n} exp(-\|x_k - a_j\|^2), \quad j \in [m],$$

   for $X = [x_1|\cdots|x_n]^T$, where kernel vectors $a_1, \ldots, a_m \in \mathbb{R}^d$ are generated randomly, each element of each vector is drawn from a standard normal distribution. Each resultant vector is then normalized to produce a kernel vector of unit l_2-norm. When inputting the embedding X to the kernels mapping, we first normalized the embedding for each respective node.
3. `identity`: In this case $\phi_{id}(X) = X$, which is obviously not a permutation invariant map.
4. `data augmentation`: In this case $\phi_{data\ augment}(X) = X$ but data augmentation is used. Our data augmentation scheme works as follows. We take the training set and create multiple permutations of the adjacency and associated feature matrix for each graph in the training set. We add each permuted graph to the training set to be included with the original graphs. In our experiments we use four added permutations for each graph when employing data augmentation.
5. `sum-pooling`: The sum-pooling method sums the feature values across the set of nodes: $\phi_{sum\ pooling}(X) = \mathbf{1}_{n\times 1}^T X$.
6. `sort-pooling`: The sort-pooling method flips entire rows of X so that the last column is ordered descendingly, $\phi_{sort\ pool}(X) = \Pi X$ where $\Pi \in S_n$ so that $\Pi X(:,d) = \downarrow(X(:,d))$.
7. `set-2-set`: This method employs a recurrent neural network that achieves permutation invariance through attention-based weighted summations. It has been introduced in [43].

In all cases, the output dimension of the ϕ-layer is $m = ND = N(d+1)$, where N is the size of the largest graph in dataset, and zero-pad $\phi(X)$ where appropriate.

For our deep neural network η we use a simple multilayer perceptron of size described in the next subsection.

Size parameters related to Γ and η components are largely held constant across the different implementations. However the network parameters are trained independently for each method.

4.1. Graph classification

4.1.1. Methodology

For our experiments in graph classification we consider the `PROTEINS_FULL` dataset obtained from [25] and originally introduced in [14]. The dataset consists of 1113 proteins falling into one of two classes: those that function as enzymes and those that do not. Across the dataset there are 450 enzymes in total. The graph for each protein is constructed such that the nodes represent amino acids and the edges represent the bonds between them. The number of amino acids (nodes) vary from around 20 to a maximum of 620 per protein with an average of 39.06. Each protein comes with a set of features annotating each node. The features represent characteristics of the associated amino acid associated to the node. The number of features is $r = 29$. We run the end-to-end model with three GCN layers in Γ, each with 50 hidden units. η consists of three dense multi-layer perceptron layers, each with 150 hidden units. In our studies, we considered four possible d: 1, 10, 50 and 100.

For each method and embedding size we train for 300 epochs. For most methods this gives each method the same exposure to training data; however, the `data augmentation` method will have effectively five times the exposure by construction. We use a batch size of 128 graphs. The loss function minimized during training is the binary cross entropy loss (BCE) defined as

$$BCE = -\frac{1}{B}\sum_{t=1}^{B} p_t \log(\sigma(\eta(\phi(X^{(t)})))) + (1-p_t)\log(1-\sigma(\eta(\phi(X^{(t)})))), \tag{4.1}$$

where $B = 128$ is the batch size, $p_t = 1$ when the t^{th} graph (protein) is an enzyme and $p_t = 0$ otherwise, $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function that maps the output $\eta(\phi(X^{(t)})$ of the 3-layer fully connected network η to $[0, 1]$. Three performance metrics were computed: accuracy (*ACC*), area under the receiver operating characteristic curve (*AUC*), and average precision (*AP*) as area under the precision-recall curve from precision scores. These measures are defined as follows (see `sklearn.metrics` module documentation in pytorch, or [18]).

For a threshold $\tau \in [0, 1]$, the classification decision $\hat{p}_t(\tau)$ is given by:

$$\hat{p}_t(\tau) = \begin{cases} 1 & \text{if} \quad \sigma(\eta(\phi(X^{(t)})) \geq \tau \\ 0 & \text{if} \quad \text{otherwise} \end{cases} . \tag{4.2}$$

By default $\tau = \frac{1}{2}$. For a given threshold, one computes four scores: true positive (*TP*), false positive (*FP*), true negative (*TN*) and false negative (*FN*), defined via:

$$TP(\tau) = \frac{1}{B_1} \sum_{t=1}^{B} 1_{\hat{p}_t(\tau)=1} 1_{p_t=1}, \quad TN(\tau) = \frac{1}{B_0} 1_{\hat{p}_t(\tau)=0} 1_{p_t=0} \tag{4.3}$$

$$FP(\tau) = \frac{1}{B_0} \sum_{t=1}^{B} 1_{\hat{p}_t(\tau)=1} 1_{p_t=0} = 1 - TN(\tau), \quad FN(\tau) = \frac{1}{B_1} \sum_{t=1}^{B} 1_{\hat{p}_t(\tau)=0} 1_{p_t=1} = 1 - TP(\tau), \tag{4.4}$$

where $B_0 = \sum_{t=1}^{B} 1_{p_t=0}$ and $B_1 = \sum_{t=1}^{B} 1_{p_t=1} = B - B_0$.

These four statistics predict Precision $P(\tau)$. Recall $R(\tau)$ (also known as sensitivity or true positive rate), and Specificity $S(\tau)$ (also known as true negative rate)

$$P(\tau) = \frac{TP(\tau)}{TP(\tau) + FP(\tau)} \quad , \quad R(\tau) = \frac{TP(\tau)}{TP(\tau) + FN(\tau)} \quad , \quad S(\tau) = \frac{TN(\tau)}{TN(\tau) + FP(\tau)} \tag{4.5}$$

Accuracy (ACC) is defined as the fraction of correct classifications for default threshold $\tau = \frac{1}{2}$ over the set of batch samples:

$$ACC = \frac{1}{B} \sum_{t=1}^{B} 1_{p_t=\hat{p}_t(\frac{1}{2})} = \frac{B_0}{B} TN(\frac{1}{2}) + \frac{B_1}{B} TP(\frac{1}{2}). \tag{4.6}$$

Area under the receiver operating characteristic curve (AUC) is computed from prediction scores as the area under true positive rate (TPR) vs. false positive rate (FPR) curve, i.e. the recall vs. 1-specificity curve

$$AUC = \frac{1}{2} \sum_{k=1}^{K} (S(\tau_{k-1}) - S(\tau_k))(R(\tau_{k-1}) + R(\tau_k)), \tag{4.7}$$

where K is the number of thresholds in that study. *Average precision (AP)* summarizes a precision-recall curve as the weighted mean of precision achieved at each threshold, with the increase in recall from the previous thresholds used as the weight:

$$AP = \sum_{k=1}^{K} (R(\tau_k) - R(\tau_{k-1}))P(\tau_k). \tag{4.8}$$

We track the binary cross entropy (BCE) through training and we compute it on the holdout set and a random node permutation of the holdout set. The lower the value of BCE, the better.

We look at the three performance metrics on the training set, the holdout set, and a random node permutation of the holdout set. For all three performance metrics, the higher the score the better.

The Supplementary Material presents traces of these metrics: see Figures A.2 and A.3 for binary cross entropy (BCE); Figures A.4, and A.5 for accuracy (*ACC*); Figures A.6, and A.7 for area under the receiver operating characteristic curve (*AUC*); and Figures A.8, and A.9 for average precision (*AP*).

4.1.2. Discussion

Tables 1-3 list values of the three performance metrics (*ACC, AUC, AP*) at the end of training (after 300 epochs). Performances over the course of training are plotted in Figures A.2 through A.9 available in the Supplementary Material section.

The authors of [14] utilized a Support Vector Machine (1-layer perceptron) for classification and obtained an accuracy (*ACC*) of 77% on the entire dataset using 52 features, and an accuracy of 80% on a smaller set of 36 features. By comparison, our data augmentation method for $d = 100$ achieved an accuracy of 97.5% on training dataset, but dropped dramatically to 73% on holdout data, and 72% on holdout dataset with randomly permuted nodes. On the other hand, both the kernels method and the sum-pooling method with $d = 50$ achieved an accuracy of around 79% on training dataset, while dropping accuracy performance by only 2% to around 77% on holdout data (as well as holdout data with nodes permuted).

For $d = 1$, data augmentation performed the best on the training set with an area under the receiver operating characteristic (*AUC*) of 0.896, followed closely by the identity method with an *AUC* of 0.886. On the permuted holdout set however, sort-pooling performed the best with an *AUC* of 0.803.

Table 1
Accuracy *ACC*(%) for enzyme/non-enzyme classification of the seven algorithms on PROTEINS_FULL dataset after 300 epochs for embedding dimensions $d = 1$ (top), $d = 10$ (second), $d = 50$ (third), and $d = 100$ (bottom).

d = 1	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	76	72	80	81.6	76.2	78	72.4
Holdout	74	74	72.5	76.5	70.5	74.5	72
Holdout Perm	74	74	67.5	75	70.5	74.5	72

d = 10	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	84.5	78.2	87	90.6	77.8	85.2	72.5
Holdout	74	75.5	73	76	75	71	74.5
Holdout Perm	74	75.5	62.5	73.5	75	71	74.5

d = 50	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	83.1	78.8	91	96	79.2	83.7	76.7
Holdout	71.5	76.5	72.5	71	77	71	76
Holdout Perm	71.5	76.5	69.5	72	77	71	76

d = 100	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	88	77	97.5	97.5	78.1	87.3	76.6
Holdout	71	74.5	72.5	73	75.5	69.5	74.5
Holdout Perm	71	74.5	68.5	72	75.5	69.5	74.5

Table 2
Area under the receiver operating characteristic curve (*AUC*) for enzyme/non-enzyme classification of the seven algorithms on PROTEINS_FULL dataset after 300 epochs for embedding dimensions $d = 1$ (top), $d = 10$ (second), $d = 50$ (third), and $d = 100$ (bottom).

d = 1	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	0.846	0.758	0.886	0.896	0.818	0.858	0.778
Holdout	0.794	0.775	0.766	0.796	0.777	0.803	0.788
Holdout Perm	0.794	0.775	0.747	0.785	0.777	0.803	0.788

d = 10	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	0.913	0.849	0.941	0.970	0.842	0.930	0.787
Holdout	0.820	0.817	0.782	0.796	0.821	0.798	0.779
Holdout Perm	0.820	0.817	0.668	0.784	0.821	0.798	0.779

d = 50	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	0.922	0.847	0.965	0.994	0.856	0.920	0.820
Holdout	0.791	0.818	0.775	0.768	0.821	0.791	0.777
Holdout Perm	0.791	0.818	0.716	0.768	0.821	0.791	0.777

d = 100	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	0.949	0.832	0.997	0.997	0.849	0.948	0.842
Holdout	0.754	0.801	0.766	0.775	0.817	0.784	0.776
Holdout Perm	0.754	0.801	0.708	0.775	0.817	0.784	0.776

For $d = 10$, sum-pooling, ordering, and kernels performed well on the permuted holdout set with *AUC*'s of 0.821, 0.820, and 0.818 respectively. The high performance of the identity method, data augmentation, and sort-pooling on the training set did not translate to the permuted holdout set at $d = 10$. By $d = 100$, sum-pooling still performed the best on the permuted holdout set with an *AUC* of 0.817. This was followed by the kernels method which achieved an *AUC* of 0.801 on the permuted holdout set.

For experiments where $d > 1$, the identity method and data augmentation show a notable drop in performance from the training set to the holdout set. This trend is also, to a lesser extent, visible in the sort pooling and ordering methods. In the holdout permuted set we see significant oscillations in the performance of both the identity and data augmentation methods.

4.2. Graph regression

4.2.1. Methodology

For our experiments in graph regression we consider the QM9 dataset [39]. This dataset consists of 134 K molecules represented as graphs, where the nodes represent atoms and edges represent the bonds between them.

Each graph has between 3 and 29 nodes, $3 \leq n \leq 29$. Each node has 11 features, $r = 11$. We hold out 20 thousand of these molecules for evaluation purposes. The dataset includes 19 quantitative features for each molecule.

In our study, following [17], we attempt to predict the electron energy gap (units eV) ($\Delta\epsilon$ in [17]), whose chemical accuracy is $0.043eV$ and whose prediction performance of any machine learning technique is worse than for any other feature, cf [20].

Table 3
Average precision (*AP*) for enzyme/non-enzyme classification of the seven algorithms on PROTEINS_FULL dataset after 300 epochs for embedding dimensions $d = 1$ (top), $d = 10$ (second), $d = 50$ (third), and $d = 100$ (bottom).

d = 1	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	0.788	0.709	0.844	0.857	0.754	0.811	0.707
Holdout	0.720	0.698	0.692	0.725	0.636	0.680	0.708
Holdout Perm	0.720	0.698	0.622	0.710	0.636	0.680	0.708

d = 10	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	0.890	0.804	0.922	0.961	0.797	0.904	0.722
Holdout	0.738	0.749	0.631	0.646	0.753	0.693	0.693
Holdout Perm	0.738	0.749	0.497	0.664	0.753	0.693	0.693

d = 50	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	0.899	0.797	0.950	0.991	0.814	0.891	0.757
Holdout	0.700	0.738	0.627	0.589	0.750	0.676	0.666
Holdout Perm	0.700	0.738	0.520	0.600	0.750	0.676	0.666

d = 100	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	0.933	0.777	0.995	0.995	0.806	0.927	0.782
Holdout	0.627	0.729	0.601	0.622	0.747	0.637	0.704
Holdout Perm	0.627	0.729	0.529	0.656	0.747	0.637	0.704

The best existing estimator for this task is enn-s2s-ens5 from [20] and has a mean absolute error (*MAE*) of $0.0529eV$, which is a factor 1.23 larger than the chemical accuracy. We run the end to end model with three GCN layers in Γ, each with 50 hidden units. η consists of three multi-layer perceptron layers, each with 150 hidden units. We use rectified linear units for the nonlinear activation function. Finally, we vary d, the size of the node embeddings output by Γ. We set d equal to 1, 10, 50 and 100.

For each method and embedding size we train for 300 epochs. Again, the data augmentation method will have experienced (effectively) five times as many training data as other methods due to the (implicitly) increased size of its training set. We use a batch size of 128 graphs. The loss function minimized during training is the mean square error (MSE) between the ground truth and the network output.

$$MSE = \frac{1}{B}\sum_{t=1}^{B} |\Delta\varepsilon_t - \eta(\phi(X^{(t)}))|^2 \tag{4.9}$$

where $B = 128$ is the batch size of 128 graphs and $\Delta\varepsilon_t$ is the electron energy gap of the t^{th} graph (molecule). The performance metric is Mean Absolute Error (*MAE*)

$$MAE = \frac{1}{B}\sum_{t=1}^{B} |\Delta\varepsilon_t - \eta(\phi(X^{(t)}))|. \tag{4.10}$$

We track the mean absolute error through the course of training. We look at this performance metric on the training set, the holdout set, and a random node permutation of the holdout set.

The Supplementary Material section contains plots of the loss function (MSE) and of the performance metric (*MAE*) tracked during training: Figures A.10, A.11, A.12, and A.13.

4.2.2. Discussion

Numerical results at the end of training (after 300 epochs) are included in Table 4. From the results we see that the ordering method performed best for $d = 100$ followed closely by the data augmentation method, while both the ordering method and the kernels method performed well for $d = 10$, though both fell slightly short of data augmentation which performed marginally better on both the training data and the holdout data, though with significantly more training iterations. For $d = 1$, the kernels method failed to train adequately. The identity mapping performed relatively well on training data (for $d = 100$ it achieved the smallest MAE among all methods and all parameters) and even the holdout data, however it lost its performance on the permuted holdout data. The identity mapping's failure to generalize across permutations of the holdout set is likely exacerbated by the fact that the QM9 data as presented to the network comes ordered in its node positions from heaviest atom to lightest. Data augmentation notably kept its performance despite this due to training on many permutations of the data.

For $d = 100$, our ordering method achieved a *MAE* of $0.155eV$ on training dataset and $0.187eV$ on holdout dataset, which are 3.6 and 4.35 times larger than the chemical accuracy ($0.043eV$), respectively. This is worse than the enn-s2s-ens5 method in [20] (as of 2022 best method) that achieved a *MAE* 0.0529 (eV), 1.23 larger than the chemical accuracy, but better than the Coulomb Matrix (CM) representation in [41] that achieved a *MAE* 5.32 larger than the chemical accuracy whose features were optimized for this task.

A key message of these experiments is that using a permutation-invariant embedding Φ induces better performance on the holdout set compared to the control setting (identity).

Table 4
Mean Absolute Error (*MAE*) for regression of the electron energy gap $\Delta\epsilon = LUMO - HOMO$ (eV) of the seven algorithms on QM9 dataset after 300 epochs for embedding dimensions $d = 1$ (top), $d = 10$ (second), $d = 50$ (third), and $d = 100$ (bottom). For each dataset, the method with best peformance is highlighted.

d = 1	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	0.302	0.867	0.320	**0.281**	0.349	0.309	0.389
Holdout	0.304	0.868	0.331	**0.285**	0.344	0.313	0.385
Holdout Perm	0.304	0.868	2.433	**0.298**	0.344	0.313	0.385

d = 10	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	0.220	0.219	0.182	**0.175**	0.214	0.226	0.282
Holdout	0.232	0.222	0.244	**0.208**	0.223	0.278	0.287
Holdout Perm	0.232	0.222	1.099	**0.216**	0.223	0.278	0.287

d = 50	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	**0.163**	0.257	**0.163**	0.172	0.182	0.166	0.196
Holdout	**0.191**	0.258	0.234	0.212	0.204	0.227	0.211
Holdout Perm	**0.191**	0.258	1.607	0.219	0.204	0.277	0.211

d = 100	ordering	kernels	identity	data augment	sum-pooling	sort-pooling	set-2-set
Training	0.155	0.269	**0.139**	0.164	0.178	0.199	0.173
Holdout	**0.187**	0.267	0.227	0.206	0.201	0.239	0.201
Holdout Perm	**0.187**	0.267	1.086	0.213	0.201	0.239	0.201

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.acha.2025.101798.

References

[1] B. Alexeev, J. Cahill, Dustin G. Mixon, Full spark frames, J. Fourier Anal. Appl. 18 (2012) 1167–1194.
[2] A.S. Bandeira, J. Cahill, D. Mixon, A.A. Nelson, Saving phase: injectivity and stability for phase retrieval, Appl. Comput. Harmon. Anal. 37 (1) (2014) 106–125.
[3] A. Auslender, M. Teboulle, Asymptotic Cones and Functions in Optimization and Variational Inequalities, Springer, 2003.
[4] R. Balan, Frames and phaseless reconstruction, in: K. Okoudjou (Ed.), Finite Frame Theory: A Complete Introduction to Overcompleteness, AMS Short Course at the Joint Mathematics Meetings, San Antonio, January 2015, in: Proceedings of Symposia in Applied Mathematics, vol. 73, 2016, pp. 175–199.
[5] R. Balan, Y. Wang, Invertibility and robustness of phaseless reconstruction, Appl. Comput. Harmon. Anal. 38 (3) (2015) 469–488.
[6] R. Balan, D. Zou, On Lipschitz analysis and Lipschitz synthesis for the phase retrieval problem, Linear Algebra Appl. 496 (2016) 152–181.
[7] A. Beck, A. Ben-Tal, M. Teboulle, Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares, SIAM J. Matrix Anal. Appl. 28 (2) (2006) 425–445.
[8] A. Beck, M. Teboulle, On minimizing quadratically constrained ratio of two quadratic functions, J. Convex Anal. 17 (3,4) (2010) 789–804.
[9] B.G. Bodmann, N. Hammen, Stable phase retrieval with low-redundancy frames, Adv. Comput. Math. 41 (2015) 317–331.
[10] Jacek Bochnak, Michel Coste, Marie-Françoise Roy, Real Algebraic Geometry, vol. 36, Springer Science & Business Media, 2013.
[11] Jameson Cahill, Joseph W. Iverson, Dustin G. Mixon, Daniel Packer, Group-invariant max filtering, Found. Comput. Math. (2024) 1–38.
[12] Cédric Villani, Topics in Optimal Transportation, American Mathematical Society, 2003.
[13] Zhengdao Chen, Soledad Villar, Lei Chen, Joan Bruna, On the equivalence between graph isomorphism testing and function approximation with GNNs, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, in: Curran Associates, vol. 32, Inc., 2019.
[14] Paul D. Dobson, Andrew J. Doig, Distinguishing enzyme structures from non-enzymes without alignments, J. Mol. Biol. 330 (4) (2003) 771–783.
[15] Emilie Dufresne, Separating invariants and finite reflection groups, Adv. Math. 221 (6) (2009) 1979–1989.
[16] Nadav Dym, Steven J. Gortler, Low dimensional invariant embeddings for universal geometric learning, Found. Comput. Math. (2025) 375–415.
[17] Felix A. Faber, Bing Hutchison, Luke Huang, Justin Gilmer, Samuel S. Schoenholz, George E. Dahl, Oriol Vinyals, Steven Kearnes, Patrick F. Riley, O. Anatole von Lilienfeld, Prediction errors of molecular machine learning models lower than hybrid DFT error, J. Chem. Theory Comput. (13) (2017) 5255–5264.
[18] Tom Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (2006) 861–874.
[19] Floris Geerts, Juan L. Reutter, Expressiveness and approximation properties of graph neural networks, in: International Conference on Learning Representations, 2022.
[20] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, George E. Dahl, Neural message passing for quantum chemistry, in: Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, 2017, pp. 1263–1272, JMLR.org.
[21] G. Kemper, H. Derksen, Computational Invariant Theory, Springer, 2002.
[22] Morris Hirsch, Differential Topology, Springer, 1994.
[23] A.C. Hip J. Cahill, A. Contreras, Complete set of translation invariant measurements with Lipschitz bounds, Appl. Comput. Harmon. Anal. 49 (2) (2020) 521–539.
[24] Nicolas Keriven, Gabriel Peyré, Universal invariant and equivariant graph neural networks, Adv. Neural Inf. Process. Syst. 32 (2019) 7092–7101.
[25] Kristian Kersting, Nils M. Kriege, Christopher Morris, Petra Mutzel, Marion Neumann, Benchmark data sets for graph kernels, http://graphkernels.cs.tu-dortmund.de, 2016.
[26] Thomas N. Kipf, Max Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations (ICLR), 2017.
[27] W. Li, W. Liao, Stable super-resolution limit and smallest singular value of restricted Fourier matrices, Appl. Comput. Harmon. Anal. 51 (2021) 118–156.
[28] Yujia Li, Daniel Tarlow, Marc Brockschmidt, Richard Zemel, Gated graph sequence neural networks, arXiv e-prints, arXiv:1511.05493, 2015.
[29] D.G. Schaeffer, M. Golubitsky, I. Stewart, Singularities and Groups in Bifurcation Theory, vol. 2, Springer, 1988.
[30] Romanos Diogenes Malikiosis, Vignon Oussa, Full spark frames in the orbit of a representation, Appl. Comput. Harmon. Anal. 49 (3) (2020) 791–814.
[31] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, Yaron Lipman, Invariant and equivariant graph networks, in: International Conference on Learning Representations, 2019.

[32] Haggai Maron, Ethan Fetaya, Nimrod Segol, Yaron Lipman, On the universality of invariant networks, in: Kamalika Chaudhuri, Ruslan Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, 9–15 Jun 2019, in: Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 4363–4371.
[33] Caroline Moosmüller, Alexander Cloninger, Linear optimal transport embedding: provable Wasserstein classification for certain rigid transformations and perturbations, Inf. Inference 12 (1) (2023) 363–389.
[34] Rémi Peyre, Comparison between W_2 distance and $\dot{H}^{-1}$ norm, and localization of Wasserstein distance, ESAIM Control Optim. Calc. Var. 24 (4) (2018) 1489–1501.
[35] R.R. Phelps, Convex sets and nearest points, Proc. Am. Math. Soc. 8 (1957) 790–797.
[36] Omri Puny, Matan Atzmon, Edward J. Smith, Ishan Mishra, Aditya Grover, Heli Ben-Hamu, Yaron Lipman, Frame averaging for invariant and equivariant network design, in: International Conference on Learning Representations, 2022.
[37] Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas, Pointnet: deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 652–660.
[38] Ali Rahimi, Benjamin Recht, Random features for large-scale kernel machines, in: J. Platt, D. Koller, Y. Singer, S. Roweis (Eds.), Advances in Neural Information Processing Systems, in: Curran Associates, vol. 20, Inc., 2007.
[39] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, O. Anatole Von Lilienfeld, Quantum chemistry structures and properties of 134 kilo molecules, Sci. Data 1 (1) (2014) 1–7.
[40] H.L. Royden, P.M. Fitzpatrick, Real Analysis, 4th ed., Pearson Education, Inc., 2010.
[41] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, O. Anatole von Lilienfeld, Fast and accurate modeling of molecular atomization energies with machine learning, Phys. Rev. Lett. 108 (2012) 058301.
[42] Akiyoshi Sannai, Yuuki Takai, Matthieu Cordonnier, Universal approximations of permutation invariant/equivariant functions by deep neural networks, 2020.
[43] Oriol Vinyals, Samy Bengio, Manjunath Kudlur, Order matters: sequence to sequence for sets, in: International Conference on Learning Representations, 2016.
[44] B. Yu Weisfeiler, A.A. Leman, The reduction of a graph to canonical form and the algebra which appears therein, Nauchno-Technicheskaya Informatsia 2 (9) (1968) 12–16, English translation by G. Ryabov is available at https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf.
[45] J.H. Wells, L.R. Williams, Embeddings and Extensions in Analysis, Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 84, Springer-Verlag, 1975.
[46] Dmitry Yarotsky, Universal approximations of invariant maps by neural networks, Constr. Approx. 55 (1) (2022) 407–474.
[47] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R. Salakhutdinov, Alexander J. Smola, Deep sets, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, in: Curran Associates, vol. 30, Inc., 2017.
[48] Muhan Zhang, Zhicheng Cui, Marion Neumann, Yixin Chen, An end-to-end deep learning architecture for graph classification, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.