



## PAPER

## OPEN ACCESS

RECEIVED  
1 April 2021REVISED  
28 June 2021ACCEPTED FOR PUBLICATION  
14 July 2021PUBLISHED  
6 August 2021

Original Content from  
this work may be used  
under the terms of the  
[Creative Commons  
Attribution 4.0 licence](#).

Any further distribution  
of this work must  
maintain attribution to  
the author(s) and the title  
of the work, journal  
citation and DOI.



# Symmetric and antisymmetric kernels for machine learning problems in quantum physics and chemistry

Stefan Klus<sup>1</sup> , Patrick Gelß<sup>2</sup> , Feliks Nüske<sup>3</sup> and Frank Noé<sup>2,4,5</sup> <sup>1</sup> Department of Mathematics, University of Surrey, Guildford, United Kingdom<sup>2</sup> Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany<sup>3</sup> Department of Mathematics, Paderborn University, Paderborn, Germany<sup>4</sup> Department of Physics, Freie Universität Berlin, Berlin, Germany<sup>5</sup> Department of Chemistry, Rice University, Houston, TX, United States of AmericaE-mail: [p.gelss@fu-berlin.de](mailto:p.gelss@fu-berlin.de)**Keywords:** symmetry and antisymmetry, reproducing kernel Hilbert spaces, quantum physics, quantum chemistry

## Abstract

We derive symmetric and antisymmetric kernels by symmetrizing and antisymmetrizing conventional kernels and analyze their properties. In particular, we compute the feature space dimensions of the resulting polynomial kernels, prove that the reproducing kernel Hilbert spaces induced by symmetric and antisymmetric Gaussian kernels are dense in the space of symmetric and antisymmetric functions, and propose a Slater determinant representation of the antisymmetric Gaussian kernel, which allows for an efficient evaluation even if the state space is high-dimensional. Furthermore, we show that by exploiting symmetries or antisymmetries the size of the training data set can be significantly reduced. The results are illustrated with guiding examples and simple quantum physics and chemistry applications.

## 1. Introduction

Kernel methods and neural networks are two of the most prevalent and versatile machine learning techniques. While various recent publications focus on invariant or equivariant deep learning algorithms, our goal is to derive kernel-based methods that exploit symmetries. Symmetries play an important role in many research areas such as physics and chemistry [1–3], but also point cloud classification problems [4] or problems defined on sets [5] are naturally permutation-invariant. One of the most prominent applications is in quantum physics. Systems of bosons require symmetric wave functions, whereas systems of fermions are represented by antisymmetric wave functions. Exploiting such symmetries of the underlying system is a popular and powerful approach that has been used to improve the performance of kernel-based methods as well as deep-learning algorithms. The goal is to obtain more accurate representations without increasing the number of training data points—resulting in more efficient learning algorithms—and to ensure that symmetry constraints are satisfied. In [1] and [2], for instance, neural networks and kernel approaches that take into account symmetries of molecules are constructed. These methods are then used for learning potential energy surfaces. An approach for constructing potential energy surfaces based on Gaussian processes combined with permutation-invariant kernels can be found in [6]. Gaussian processes that exploit symmetries by summing over permutations of identical atoms are also utilized in [7] to improve the accuracy of density functional theory descriptions. Moreover, the so-called SOAP (smooth overlap of atomic positions) kernel [8] is a popular framework to design translation-, rotation-, and permutation-invariant descriptors of molecules. In [9], general invariant kernels (capturing discrete and continuous transformations) for pattern analysis are defined and analyzed. Recently, neural network architectures for antisymmetric wavefunctions have been proposed [10–14] that typically operate by applying Slater determinants to the outputs. The neural networks optimize the basis functions entering the Slater determinants through a deep learning variant of a technique called *backflow*. Backflow is a method to modify the basis functions used in quantum Monte Carlo as trial wavefunctions [15]. Neural network approaches such as *FermiNet* [11] and *PauliNet* [12] achieve extremely high accuracy with relatively few Slater

determinants compared to standard quantum chemistry methods that build Slater determinants with fixed basis functions. Kernels, on the other hand, accomplish this by mapping the data to potentially infinite-dimensional feature spaces. Any continuous antisymmetric function can be approximated by antisymmetrized universal kernels. The universal approximation of symmetric and anti-symmetric functions is also studied in [16].

In this work, we develop kernels that are intrinsically symmetric or antisymmetric. Although we focus mostly on physics and chemistry applications in what follows, the derived kernels can be used in the same way in other kernel-based supervised or unsupervised learning algorithms such as kernel principal component analysis (kernel PCA) [17], kernel canonical correlation analysis (kernel CCA) [18], or support vector machines (SVMs) [19]. The main contributions are:

- We derive symmetric and antisymmetric kernels based on conventional kernels such as polynomial and Gaussian kernels and show that certain kernels can be expressed as Slater permanents or determinants.
- We analyze the feature spaces and approximation properties of such kernels.
- We demonstrate that these techniques improve the efficiency of kernel-based methods for problems exhibiting symmetries or antisymmetries.
- We apply kernel-based methods for solving the time-independent Schrödinger equation to simple quantum mechanics problems. Furthermore, we predict the boiling points of molecules using kernel ridge regression.

In section 2, we first introduce kernels, reproducing kernel Hilbert spaces, and kernel-based methods for solving the time-independent Schrödinger equation. Antisymmetric kernels will be derived in section 3 and symmetric kernels in section 4. These two sections contain the main theoretical results, in particular the analysis of the properties of the resulting polynomial and Gaussian kernels. Numerical results will be presented in section 5. We conclude the paper with a list of open problems and future research.

## 2. Kernels and kernel-based methods

We will briefly recapitulate the properties of kernels and introduce the induced reproducing kernel Hilbert spaces. Additionally, we will present a kernel-based method for solving the time-independent Schrödinger equation.

### 2.1. Reproducing kernel Hilbert spaces

A kernel can be regarded as a similarity measure. We will focus on real-valued kernels, but the definitions can be easily extended to complex domains.

**Definition 2.1** (Kernel [19]). *Given a non-empty set  $\mathbb{X}$ , a function  $k : \mathbb{X} \times \mathbb{X}$  is called kernel if there exists a Hilbert space  $\mathbb{H}$  and a feature map  $\phi : \mathbb{X} \rightarrow \mathbb{H}$  such that*

$$k(x, x') = \langle \phi(x), \phi(x') \rangle.$$

For a given kernel  $k$ , the so-called *Gram matrix*  $G \in \mathbb{R}^{m \times m}$  associated with a data set  $\{x^{(i)}\}_{i=1}^m \subset \mathbb{X}$  is defined by  $G_{ij} = k(x^{(i)}, x^{(j)})$ .

**Definition 2.2** (Positive definiteness [19]). *A function  $k : \mathbb{X} \times \mathbb{X}$  is called positive definite if for all  $m$ , all vectors  $c = [c_1, \dots, c_m]^\top \in \mathbb{R}^m$ , and all subsets  $\{x^{(i)}\}_{i=1}^m \subset \mathbb{X}$  it holds that*

$$c^\top G c = \sum_{i=1}^m \sum_{j=1}^m c_i c_j k(x^{(i)}, x^{(j)}) \geq 0.$$

Strictly positive definite means that  $c^\top G c = 0$  for mutually distinct data points only if  $c = 0$ . It can be shown that a function  $k : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$  is a kernel if and only if it is symmetric, i.e.  $k(x, x') = k(x', x)$ , and positive definite (s.p.d. in what follows to avoid confusion between different notions of symmetry), see [19]. Such kernels induce so-called reproducing kernel Hilbert spaces.

**Definition 2.3** (RKHS [19, 20]). *Let  $\mathbb{X}$  be a non-empty set. A space  $\mathbb{H}$  of functions  $f : \mathbb{X} \rightarrow \mathbb{R}$  is called reproducing kernel Hilbert space (RKHS) with inner product  $\langle \cdot, \cdot \rangle_{\mathbb{H}}$  if a kernel  $k$  exists such that*

- $f(x) = \langle f, k(x, \cdot) \rangle_{\mathbb{H}}$  for all  $f \in \mathbb{H}$ , and
- $\mathbb{H} = \overline{\text{span}\{k(x, \cdot) \mid x \in \mathbb{X}\}}$ .

The first requirement is called the *reproducing property*. For  $f = k(x, \cdot)$ , this results in  $k(x, x') = \langle k(x, \cdot), k(x', \cdot) \rangle_{\mathbb{H}}$  so that we can define the so-called *canonical feature map* by  $\phi(x) = k(x, \cdot)$ . Additionally, for a data set  $\{x^{(i)}\}_{i=1}^m$ , we define  $\Phi = [\phi(x_1), \dots, \phi(x_m)]$  so that  $G = \Phi^T \Phi$ . For more details on kernels and reproducing kernel Hilbert spaces, we refer to [19, 20]. It was shown in [21, 22] that not only function evaluations but also derivative evaluations can be represented as inner products in the RKHS  $\mathbb{H}$ , provided the kernel is sufficiently smooth. Let now  $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}_0^d$  be a multi-index. We define  $|\alpha| = \sum_{i=1}^d \alpha_i$  as usual and, for a fixed  $r \in \mathbb{N}_0$ , the index set  $I_r = \{\alpha \in \mathbb{N}_0^d : |\alpha| \leq r\}$ . Given a function  $f: \mathbb{X} \rightarrow \mathbb{R}$ , the partial derivative of  $f$  with respect to  $\alpha$  is defined by

$$\mathcal{D}^\alpha f = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} f.$$

**Theorem 2.4** ([21, 22]). *Let  $r \in \mathbb{N}_0$  be a non-negative number,  $k \in C^{2r}(\mathbb{X} \times \mathbb{X})$  a kernel, and  $\mathbb{H}$  the induced RKHS. Then:*

- (a)  $\mathcal{D}^\alpha k(x, \cdot) \in \mathbb{H}$  for any  $x \in \mathbb{X}$  and  $\alpha \in I_r$ .
- (b)  $\langle \mathcal{D}^\alpha f \rangle(x) = \langle \mathcal{D}^\alpha k(x, \cdot), f \rangle_{\mathbb{H}}$  for any  $x \in \mathbb{X}$ ,  $f \in \mathbb{H}$ , and  $\alpha \in I_r$ .

In (i) and (ii), the derivative  $\mathcal{D}^\alpha$  is understood as acting on the first argument of the kernel  $k$ .

We will need this property later for the approximation of differential operators. Another question is how rich these Hilbert spaces  $\mathbb{H}$  induced by a kernel  $k$  are.

**Definition 2.5** (Universal kernel [23]). *Let  $\mathbb{X}$  be compact and  $C(\mathbb{X})$  the space of all continuous functions mapping from  $\mathbb{X}$  to  $\mathbb{R}$  equipped with  $\|\cdot\|_\infty$ . A kernel  $k$  is called universal if the induced RKHS  $\mathbb{H}$  is dense in  $C(\mathbb{X})$ .*

That is, for a function  $f \in C(\mathbb{X})$ , we can find a function  $g \in \mathbb{H}$  such that  $\|g - f\|_\infty < \varepsilon$  for any  $\varepsilon > 0$ . The Gaussian kernel

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right),$$

for instance, is universal, while the polynomial kernel

$$k(x, x') = (c + x^T x')^q$$

is not. We will analyze the properties of these kernels and their symmetrized and antisymmetrized counterparts in more detail below. Various other notions of universality and the relationships between universal and characteristic kernels are discussed in [24]. In what follows, we will omit the subscript  $\mathbb{H}$  if it is clear which inner product or norm we are referring to.

### 2.2. Kernel-based solution of the Schrödinger equation

In [25], we proposed a kernel-based method for the solution of the time-independent Schrödinger equation and the approximation of other differential operators such as the generator of the Koopman operator. We will restrict ourselves to the Schrödinger equation. Let  $V$  be a potential and  $\mathcal{H} = -\frac{\hbar^2}{2\mathbf{m}}\Delta + V$  the Hamiltonian, where  $\hbar$  is the reduced Planck constant and  $\mathbf{m}$  the mass, then the time-independent Schrödinger equation is defined by

$$\mathcal{H}\psi = E\psi.$$

That is, we want to compute eigenfunctions  $\psi$  and the associated eigenvalues  $E$ , which correspond to energies of the system. We define

$$\mathcal{C}\phi(x) = -\frac{\hbar^2}{2\mathbf{m}} \sum_{l=1}^d \mathcal{D}^{2e_l} \phi(x) + V(x)\phi(x),$$

where  $e_l$  is the  $l$ th unit vector, and operators

$$\mathcal{C}_{00} = \int \phi(x) \otimes \phi(x) d\mu(x) \quad \text{and} \quad \mathcal{C}_{01} = \int \phi(x) \otimes \mathcal{C}\phi(x) d\mu(x).$$

Here,  $C_{00}$  is the standard covariance operator (see [26]) and  $C_{01}$  contains the action of the Schrödinger operator. Since these integrals typically cannot be computed in practice, we estimate them using  $\mu$ -distributed training data  $\{x^{(i)}\}_{i=1}^m$ , resulting in the empirical operators

$$\widehat{C}_{00} = \frac{1}{m} \sum_{i=1}^m \phi(x^{(i)}) \otimes \phi(x^{(i)}) \quad \text{and} \quad \widehat{C}_{01} = \frac{1}{m} \sum_{i=1}^m \phi(x^{(i)}) \otimes d\phi(x^{(i)}).$$

Assuming that the eigenfunctions can be represented as  $\widehat{\psi} = \Phi u$ , i.e. they are contained in the space spanned by the functions  $\{\phi(x^{(i)})\}_{i=1}^m$ , we obtain a matrix eigenvalue problem

$$G_{10}u = \widehat{E}G_{00}u,$$

where the entries of the (generalized) Gram matrices  $G_{00}, G_{10} \in \mathbb{R}^{m \times m}$  are defined by

$$[G_{00}]_{ij} = \left[ \phi(x^{(i)}) \right] (x^{(j)}) = k(x^{(i)}, x^{(j)})$$

and

$$[G_{10}]_{ij} = \left[ d\phi(x^{(i)}) \right] (x^{(j)}) = -\frac{\hbar^2}{2m} \sum_{l=1}^d \mathcal{D}^{2e_l} k(x_i, x_j) + V(x_i)k(x_i, x_j).$$

Eigenfunctions are then of the form

$$\widehat{\psi} = \Phi u = \sum_{i=1}^m u_i k(x^{(i)}, \cdot).$$

A detailed derivation and numerical results for simple quantum mechanics problems—the quantum harmonic oscillator and the hydrogen atom—can be found in [25].

### 3. Antisymmetric kernels and their properties

In this section, we will introduce the notion of *antisymmetric kernels* and define antisymmetric counterparts of well-known kernels such as the polynomial kernel and the Gaussian kernel. Furthermore, we analyze the properties of the resulting reproducing kernel Hilbert spaces. Most results can then be carried over to the symmetric case, which will be studied in section 4.

#### 3.1. Antisymmetric kernels

Let  $\mathbb{X} \subset \mathbb{R}^d$  be the state space. Furthermore, let  $S_d$  be the symmetric group and  $\pi \in S_d$  a permutation. With a slight abuse of notation, we define  $\pi(x) = [x_{\pi(1)}, \dots, x_{\pi(d)}]^\top$  to be the vector  $x \in \mathbb{X}$  permuted by  $\pi$ . A function  $f: \mathbb{X} \rightarrow \mathbb{R}$  is called antisymmetric if

$$f(x) = \text{sgn}(\pi)f(\pi(x)),$$

where  $\text{sgn}(\pi)$  denotes the sign of the permutation  $\pi$ , which is 1 if the number of transpositions is even and  $-1$  if it is odd. We define the *antisymmetrization operator*  $\mathcal{A}$  by

$$(\mathcal{A}f)(x) = \frac{1}{d!} \sum_{\pi \in S_d} \text{sgn}(\pi)f(\pi(x)).$$

**Remark 3.1.** In the same way, we can consider state spaces of the form  $\mathbb{X} \subset \bigoplus_{i=1}^{d_x} \mathbb{R}^{d_y}$ . Functions would then be antisymmetric with respect to permutations of vectors in  $\mathbb{R}^{d_y}$ . That is, for  $x = [x_1, \dots, x_{d_x}]^\top$  with  $x_i \in \mathbb{R}^{d_y}$ , the permuted vector is then  $\pi(x) = [x_{\pi(1)}, \dots, x_{\pi(d_x)}]^\top$ . For typical quantum mechanics applications, for instance,  $d_y = 3$  (every particle has a position in a three-dimensional space) and  $d_x$  the number of fermions (or bosons in the symmetric case). The special case  $d_x = 2$  is considered in [27], where the spectral properties of symmetric and antisymmetric pairwise kernels are analyzed. Supervised learning problems with such pairwise kernels are discussed in [28].

Our goal is to define antisymmetric kernels for arbitrary  $d$ , which can then be used in kernel-based learning algorithms.

**Definition 3.2** (Antisymmetric kernel function). Let  $k : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$  be a kernel. We define an antisymmetric function  $k_a : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$  by

$$k_a(x, x') = \frac{1}{d!^2} \sum_{\pi \in S_d} \sum_{\pi' \in S_d} \text{sgn}(\pi) \text{sgn}(\pi') k(\pi(x), \pi'(x')).$$

Clearly, if  $k(x, x') = k(x', x)$ , then also  $k_a(x, x') = k_a(x', x)$ . Furthermore, for a fixed permutation  $\hat{\pi} \in S_d$ , it holds that

$$\begin{aligned} k_a(\hat{\pi}(x), x') &= \frac{1}{d!^2} \sum_{\pi \in S_d} \sum_{\pi' \in S_d} \text{sgn}(\pi) \text{sgn}(\pi') k(\pi(\hat{\pi}(x)), \pi'(x')) \\ &= \text{sgn}(\hat{\pi}) \frac{1}{d!^2} \sum_{\pi \in S_d} \sum_{\pi' \in S_d} \text{sgn}(\pi \circ \hat{\pi}) \text{sgn}(\pi') k([\pi \circ \hat{\pi}](x), \pi'(x')) \\ &= \text{sgn}(\hat{\pi}) \frac{1}{d!^2} \sum_{\pi \in S_d} \sum_{\pi' \in S_d} \text{sgn}(\pi) \text{sgn}(\pi') k(\pi(x), \pi'(x')) \\ &= \text{sgn}(\hat{\pi}) k_a(x, x'). \end{aligned} \tag{1}$$

Here, we used the fact that  $\text{sgn}(\hat{\pi}) = \text{sgn}(\hat{\pi}^{-1})$  and  $\text{sgn}(\pi \circ \hat{\pi}) = \text{sgn}(\pi) \text{sgn}(\hat{\pi})$ . Additionally, we utilized the property that for a function  $g : S_d \rightarrow \mathbb{R}$  it holds that  $\sum_{\pi \in S_d} g(\pi) = \sum_{\pi \in S_d} g(\pi \circ \hat{\pi})$ , which corresponds to a reordering of the summands. Thus,  $k_a$  is antisymmetric in both arguments. From (1) it directly follows that  $k_a(x, x') = 0$  if at least two entries of  $x$  or  $x'$  are equal<sup>6</sup>.

**Lemma 3.3.** The function  $k_a$  defines an s.p.d. kernel.

*Proof.* We have

$$\begin{aligned} k_a(x, x') &= \frac{1}{d!^2} \sum_{\pi \in S_d} \sum_{\pi' \in S_d} \text{sgn}(\pi) \text{sgn}(\pi') k(\pi(x), \pi'(x')) \\ &= \frac{1}{d!^2} \sum_{\pi \in S_d} \sum_{\pi' \in S_d} \text{sgn}(\pi) \text{sgn}(\pi') \langle \phi(\pi(x)), \phi(\pi'(x')) \rangle \\ &= \left\langle \frac{1}{d!} \sum_{\pi \in S_d} \text{sgn}(\pi) \phi(\pi(x)), \frac{1}{d!} \sum_{\pi' \in S_d} \text{sgn}(\pi') \phi(\pi'(x')) \right\rangle \\ &= \langle \phi_a(x), \phi_a(x') \rangle, \end{aligned}$$

where  $\phi_a(x) = \frac{1}{d!} \sum_{\pi \in S_d} \text{sgn}(\pi) \phi(\pi(x))$ . That is,  $k_a$  is a kernel. Symmetry was shown above. To see that the function is positive definite, let  $c = [c_1, \dots, c_m]^T \in \mathbb{R}^m$  be a coefficient vector and  $\{x^{(i)}\}_{i=1}^m \subset \mathbb{X}$ . Then

$$\begin{aligned} c^T G_a c &= \sum_{i=1}^m \sum_{j=1}^m c_i c_j k_a(x^{(i)}, x^{(j)}) = \sum_{i=1}^m \sum_{j=1}^m c_i c_j \langle \phi_a(x^{(i)}), \phi_a(x^{(j)}) \rangle \\ &= \left\langle \sum_{i=1}^m c_i \phi_a(x^{(i)}), \sum_{j=1}^m c_j \phi_a(x^{(j)}) \right\rangle = \left\| \sum_{i=1}^m c_i \phi_a(x^{(i)}) \right\|^2 \geq 0. \end{aligned}$$

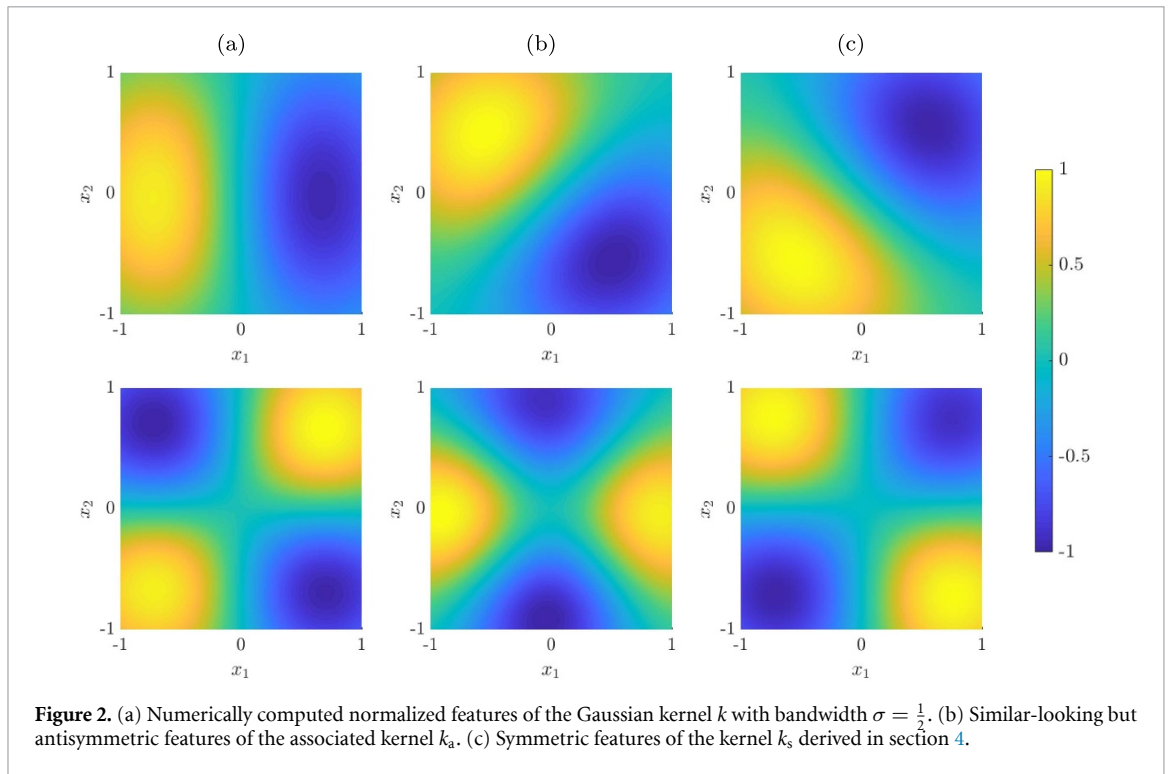
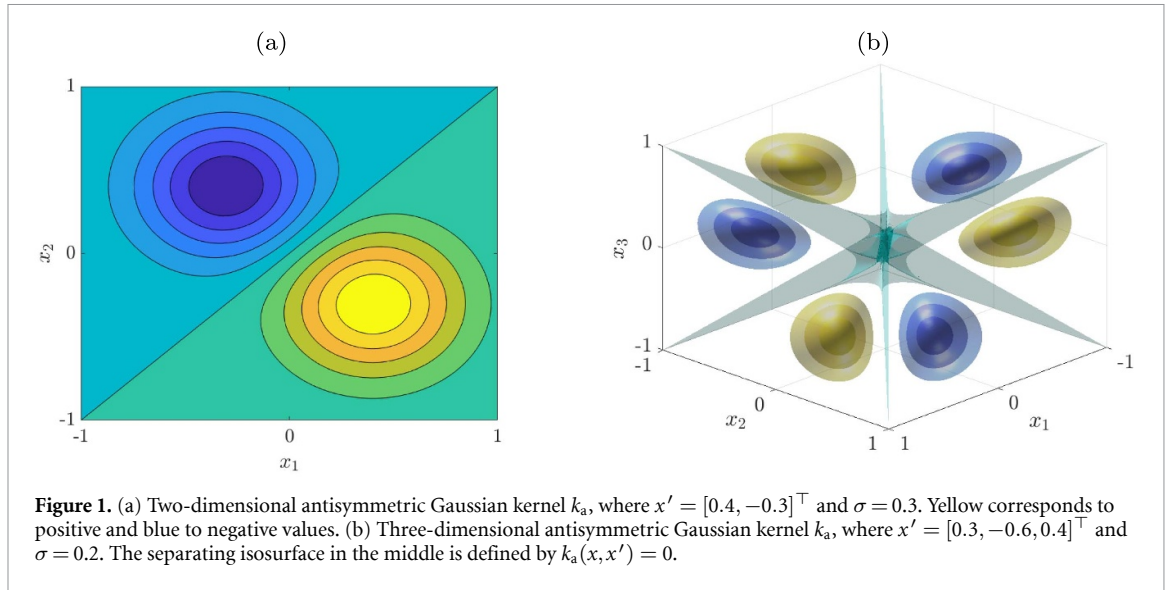
□

The antisymmetrized two- and three-dimensional Gaussian kernels are visualized in figure 1. The feature space mapping of the antisymmetric kernel  $k_a$  is the antisymmetrization operator  $\mathcal{A}$  applied to the feature space mapping of the kernel  $k$ .

**Example 3.4.** For  $\mathbb{X} \subset \mathbb{R}^2$ , the feature space of the quadratic kernel  $k(x, x') = (1 + x^T x')^2$  is spanned by  $\{1, x_1, x_2, x_1^2, x_1 x_2, x_2^2\}$  and thus six-dimensional. The feature space of the antisymmetrized kernel  $k_a$  is spanned by the two antisymmetric functions  $\{x_1 - x_2, x_1^2 - x_2^2\}$ . This illustrates that the feature space is significantly reduced. △

Polynomial kernels of arbitrary degree  $p$  for  $d$ -dimensional spaces will be discussed in more detail in section 3.2.

<sup>6</sup> Assume w.l.o.g. that  $x_i = x_j$  for some indices  $i \neq j$ . Let  $\hat{\pi}$  be the permutation which only swaps the positions  $i$  and  $j$ , then it holds that  $k_a(x, x') = k_a(\hat{\pi}(x), x') = \text{sgn}(\hat{\pi}) k_a(x, x') = -k_a(x, x')$ .



**Remark 3.5.** The Mercer features of a kernel  $k$  are defined by the eigenfunctions of the integral operator

$$(\mathcal{T}_k f)(x) = \int k(x, x') f(x') d\mu(x')$$

multiplied by the square root of the associated eigenvalues  $\lambda$ , see [19]. The Mercer features of an antisymmetric kernel  $k_a$  are automatically antisymmetric. This can be seen as follows: let  $\varphi$  be an eigenfunction of  $\mathcal{T}_{k_a}$  with corresponding eigenvalue  $\lambda$ , then

$$\lambda \varphi(\pi(x)) = \int k_a(\pi(x), x') \varphi(x') d\mu(x') = \text{sgn}(\pi) \int k_a(x, x') \varphi(x') d\mu(x') = \lambda \text{sgn}(\pi) \varphi(x).$$

Mercer features of the Gaussian kernel and its antisymmetric and symmetric (see section 4) counterparts—computed by a spectral decomposition of the covariance operator, cf [29]—are shown in figure 2.

**Definition 3.6** (Permutation invariance). We call a kernel permutation-invariant if

$$k(x, x') = k(\pi(x), \pi(x'))$$

for all permutations  $\pi \in S_d$ .

The Gaussian kernel and the polynomial kernel are permutation-invariant since the standard inner product and induced norm are permutation-invariant, i.e.  $\langle xx' \rangle = \langle \pi(x)\pi(x') \rangle$  for a permutation  $\pi \in S_d$ . The antisymmetric kernel  $k_a$  is permutation-invariant by construction. While many kernels used in practice are naturally permutation-invariant, an open question is whether this assumption limits the expressivity of the induced function space. We will analyze the properties of the Gaussian kernel in section 3.3. The permutation-invariance allows us to simplify the representation of the antisymmetric kernel.

**Lemma 3.7.** *Given a permutation-invariant kernel  $k$ , it holds that*

$$k_a(x, x') = \frac{1}{d!} \sum_{\pi \in S_d} \text{sgn}(\pi)k(\pi(x), x') = \frac{1}{d!} \sum_{\pi \in S_d} \text{sgn}(\pi)k(x, \pi(x')).$$

*Proof.* We obtain

$$\begin{aligned} k_a(x, x') &= \frac{1}{d!^2} \sum_{\pi \in S_d} \sum_{\pi' \in S_d} \text{sgn}(\pi)\text{sgn}(\pi')k(\pi(x), \pi'(x')) \\ &= \frac{1}{d!^2} \sum_{\pi \in S_d} \sum_{\pi' \in S_d} \text{sgn}(\pi)\text{sgn}(\pi')k([\pi'^{-1} \circ \pi](x), x') \\ &= \frac{1}{d!^2} \sum_{\pi \in S_d} \sum_{\pi' \in S_d} \text{sgn}(\pi'^{-1} \circ \pi)k([\pi'^{-1} \circ \pi](x), x') \\ &= \frac{1}{d!} \sum_{\pi \in S_d} \text{sgn}(\pi)k(\pi(x), x') \end{aligned}$$

since all permutations occur  $d!$  times. In the third line, we used the same properties of permutations as above. The proof for the second representation is analogous.  $\square$

For the sake of simplicity, assume now that the kernel  $k$  is permutation-invariant. We want to show that for a universal kernel  $k$ , the reproducing kernel Hilbert space induced by the corresponding antisymmetric kernel  $k_a$  is dense in the space of antisymmetric functions.

**Proposition 3.8.** *Let  $\mathbb{X}$  be bounded. Given a universal, permutation-invariant, continuous kernel  $k$ , the space  $\mathbb{H}_a$  induced by  $k_a$  is dense in the space of continuous antisymmetric functions given by  $C_a(\mathbb{X}) = \{f \in C(\mathbb{X}) \mid f \text{ is antisymmetric}\}$ .*

*Proof.* Let  $f$  be antisymmetric. It follows that  $f(x) = \text{sgn}(\pi)f(\pi(x))$  for all  $\pi \in S_d$  and thus

$$f(x) = \frac{1}{d!} \sum_{\pi \in S_d} \text{sgn}(\pi)f(\pi(x)).$$

Since  $k$  is assumed to be universal, we can find coefficients  $\alpha_i \in \mathbb{R}$  and vectors  $x^{(i)} \in \mathbb{X}$  such that  $\|\sum_{i=1}^n \alpha_i k(\cdot, x^{(i)}) - f\|_\infty < \varepsilon$ . Then

$$\begin{aligned} \left\| \sum_{i=1}^n \alpha_i k_a(\cdot, x^{(i)}) - f \right\|_\infty &= \left\| \sum_{i=1}^n \frac{\alpha_i}{d!} \sum_{\pi \in S_d} \text{sgn}(\pi)k(\pi(\cdot), x^{(i)}) - \frac{1}{d!} \sum_{\pi \in S_d} \text{sgn}(\pi)f(\pi(\cdot)) \right\|_\infty \\ &= \left\| \frac{1}{d!} \sum_{\pi \in S_d} \text{sgn}(\pi) \underbrace{\left[ \sum_{i=1}^n \alpha_i k(\pi(\cdot), x^{(i)}) - f(\pi(\cdot)) \right]}_{< \varepsilon \forall \pi} \right\|_\infty < \varepsilon. \end{aligned} \quad \square$$

Continuous antisymmetric functions can be approximated arbitrarily well by universal antisymmetric kernels such as the Gaussian kernel. Although we used the same number of data points for the approximation in the proof (i.e.  $n$  points for the expansion in terms of  $k$  and also  $k_a$ ), fewer data points are required in practice if we employ the antisymmetric kernel, see example 3.14.

### 3.2. Antisymmetric polynomial kernels

We have seen in example 3.4 that the feature space dimension of the polynomial kernel of order two for  $\mathbb{X} \subset \mathbb{R}^2$  is reduced from six to two by the antisymmetrization. Let  $q = (q_1, \dots, q_d) \in \mathbb{N}_0^d$  be a multi-index and

$|q| = \sum_{i=1}^d q_i$ . We define  $x^q = \prod_{i=1}^d x_i^{q_i}$ . For a  $d$ -dimensional state space  $\mathbb{X}$ , the polynomial kernel of order  $p$  is then given by

$$k(x, x') = (c + x^\top x')^p = \sum_{0 \leq |q| \leq p} (\sqrt{a_q} x^q) (\sqrt{a_q} x'^q),$$

where

$$a_q = \binom{p}{q} \frac{c^{q_0}}{q_0!}$$

and  $q_0 = p - |q|$ , cf [30]. The multinomial coefficients are defined by

$$\binom{p}{q} = \frac{p!}{q_1! \dots q_d!}.$$

Thus, the feature space is spanned by the monomials  $\{x^q \mid 0 \leq |q| \leq p\}$  and the dimension of the feature space is  $n_\phi = \binom{p+d}{d}$ , see, e.g. [31].

We now want to find the feature space of the corresponding antisymmetric kernel  $k_a$ . Given a multi-index  $q$ , assume that there exist two entries  $q_i$  and  $q_j$  with  $q_i = q_j$ . Since the transposition  $(i, j)$  leaves the multi-index (and thus  $x^q$ ) unchanged, this monomial will be eliminated by the antisymmetrization operator. It follows that the monomials must have distinct indices. In fact, the nonzero images of monomials under antisymmetrization are of the form

$$f_\mu(x) = \begin{vmatrix} x_1^{\delta_1+\mu_1} & x_1^{\delta_2+\mu_2} & \dots & x_1^{\delta_d+\mu_d} \\ x_2^{\delta_1+\mu_1} & x_2^{\delta_2+\mu_2} & \dots & x_2^{\delta_d+\mu_d} \\ \vdots & \vdots & \ddots & \vdots \\ x_d^{\delta_1+\mu_1} & x_d^{\delta_2+\mu_2} & \dots & x_d^{\delta_d+\mu_d} \end{vmatrix}, \tag{2}$$

where  $\delta = (d-1, d-2, \dots, 0)$  and  $\mu = (\mu_1, \dots, \mu_d)$  with  $\mu_1 \geq \mu_2 \geq \dots \geq \mu_d \geq 0$  is a partition of a positive integer<sup>7</sup>, see [32]. The degrees of the terms of this antisymmetric polynomial are  $|\mu| + \binom{d}{2}$ . Since we need all monomials of order  $0 \leq |q| \leq p$ , we have to consider the partitions  $\mu$  of  $0 \leq p_r \leq p - \binom{d}{2}$ .

This representation uses the fact that multi-indices corresponding to antisymmetric polynomials can be written as  $q = \delta + \mu$ , where  $\delta$  is defined as above and  $\mu$  a partition. It follows that an antisymmetric polynomial must be at least of order  $\binom{d}{2}$ . Equation (2) can be regarded as a Slater determinant (introduced below) for a specific set of functions. We also obtain the Vandermonde determinant (up to the sign) as a special case where  $\mu = 0$ .

**Definition 3.9** (Partition function). *Let  $s_\ell(n)$  be the function that counts the partitions of  $n$  into exactly  $\ell$  parts.*

A closed-form expressions for  $s_\ell(n)$  is not known, but it can be expressed in terms of generating functions or computed using the recurrence relation

$$s_\ell(n) = s_\ell(n - \ell) + s_{\ell-1}(n - 1),$$

where we define  $s_\ell(n) = 1$  if  $n = 0$  and  $\ell = 0$  and  $s_\ell(n) = 0$  if  $n \leq 0$  or  $\ell \leq 0$  (but not  $n = \ell = 0$ ), see [33] for more details about partitions and partition functions.

**Lemma 3.10.** *The dimension of the feature space generated by the antisymmetrized polynomial kernel of order  $p$  is*

$$n_{\phi_a} = \sum_{p_r=0}^{p-\binom{d}{2}} \sum_{j=0}^d s_j(p_r).$$

*Proof.* Since  $\binom{d}{2}$  of the  $|q|$  exponents are already spoken for, we can use only the remaining  $|q| - \binom{d}{2}$  to generate partitions  $\mu$ , with  $0 \leq |q| \leq p$ . All these numbers can be decomposed into at most  $d$  parts since we have only  $d$  variables. If the number of components is smaller than  $d$ , we simply add zeros. □

<sup>7</sup> A partition of a positive integer  $n$  is a decomposition into positive integers so that the sum is  $n$ . The order of the summands does not matter, i.e.  $6 = 3 + 2 + 1$  and  $6 = 1 + 2 + 3$  are the same partition. We sort partitions in non-increasing order, e.g.  $\mu = (3, 2, 1)$  is a partition of 6 into three parts.



**Table 1.** Dimensions of the feature spaces spanned by the polynomial kernel  $k$  and its antisymmetric counterpart  $k_a$ . Here,  $d$  is the dimension of the state space and  $p$  the degree of the polynomial kernel.

$d \backslash p$	2		3		4		5		6		7		8	
	$n_\phi$	$n_{\phi_a}$	$n_\phi$	$n_{\phi_a}$	$n_\phi$	$n_{\phi_a}$	$n_\phi$	$n_{\phi_a}$	$n_\phi$	$n_{\phi_a}$	$n_\phi$	$n_{\phi_a}$	$n_\phi$	$n_{\phi_a}$
2	6	2	10	4	15	6	21	9	28	12	36	16	45	20
3	10	0	20	1	35	2	56	4	84	7	120	11	165	16
4	15	0	35	0	70	0	126	0	210	1	330	2	495	4

**Example 3.11.** For  $d = 3$  and  $p = 6$ , the base case is  $\delta = (2, 1, 0)$ , and we can generate partitions for

$$\begin{aligned}
 p_r = 0 : \mu &= (0, 0, 0), & q &= (2, 1, 0), \\
 p_r = 1 : \mu &= (1, 0, 0), & q &= (3, 1, 0), \\
 p_r = 2 : \mu &= (2, 0, 0), (1, 1, 0), & q &= (4, 1, 0), (3, 2, 0), \\
 p_r = 3 : \mu &= (3, 0, 0), (2, 1, 0), (1, 1, 1), & q &= (5, 1, 0), (4, 2, 0), (3, 2, 1),
 \end{aligned}$$

resulting in seven antisymmetric polynomials. △

The sizes of the feature spaces of the polynomial kernels  $k$  and  $k_a$  for different dimensions  $d$  and degrees  $p$  are summarized in table 1. This shows that antisymmetric polynomial kernels might not be feasible for higher-dimensional problems. For  $d = 10$ , for example, the lowest degree of the monomials is already 45.

### 3.3. Antisymmetric Gaussian kernels

We will now analyze the properties of the Gaussian kernel. We have shown in Proposition 3.8 that the space spanned by the antisymmetric Gaussian kernel is dense in the space of continuous antisymmetric functions. For the Gaussian kernel, the expression obtained in lemma 3.7 can be simplified even further.

**Lemma 3.12.** Let  $k$  be the Gaussian kernel with bandwidth  $\sigma$ , then

$$k_a(x, x') = \frac{1}{d!} \begin{vmatrix} e^{-\frac{(x_1-x'_1)^2}{2\sigma^2}} & \dots & e^{-\frac{(x_1-x'_d)^2}{2\sigma^2}} \\ \vdots & \ddots & \vdots \\ e^{-\frac{(x_d-x'_1)^2}{2\sigma^2}} & \dots & e^{-\frac{(x_d-x'_d)^2}{2\sigma^2}} \end{vmatrix}.$$

*Proof.* Applying Leibniz' formula

$$\det(A) = \sum_{\pi \in S_d} \text{sgn}(\pi) \prod_{i=1}^d a_{\pi(i),i},$$

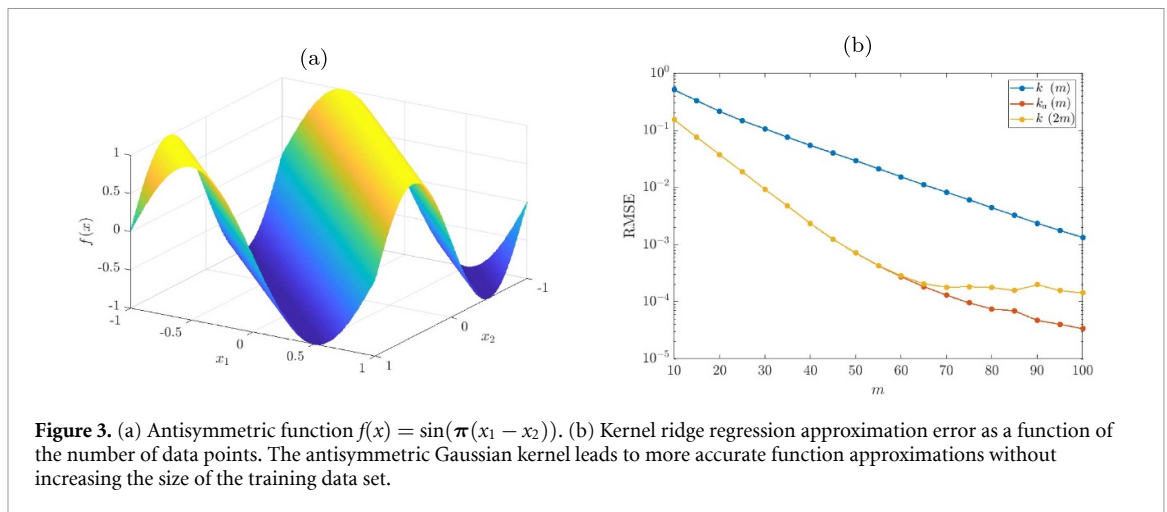
we have

$$\begin{aligned}
 \begin{vmatrix} e^{-\frac{(x_1-x'_1)^2}{2\sigma^2}} & \dots & e^{-\frac{(x_1-x'_d)^2}{2\sigma^2}} \\ \vdots & \ddots & \vdots \\ e^{-\frac{(x_d-x'_1)^2}{2\sigma^2}} & \dots & e^{-\frac{(x_d-x'_d)^2}{2\sigma^2}} \end{vmatrix} &= \sum_{\pi \in S_d} \text{sgn}(\pi) \prod_{i=1}^d e^{-\frac{(x_{\pi(i)}-x'_i)^2}{2\sigma^2}} \\
 &= \sum_{\pi \in S_d} \text{sgn}(\pi) e^{-\frac{\sum_{i=1}^d (x_{\pi(i)}-x'_i)^2}{2\sigma^2}} \\
 &= \sum_{\pi \in S_d} \text{sgn}(\pi) e^{-\frac{\|\pi(x)-x'\|^2}{2\sigma^2}} \\
 &= \sum_{\pi \in S_d} \text{sgn}(\pi) k(\pi(x), x').
 \end{aligned}$$

Lemma 3.7 then yields the desired result. □

This decomposition is akin to the well-known Slater determinant (see, e.g. [34]), which defines an antisymmetric wave function by

$$\psi_a(x) = \frac{1}{\sqrt{d!}} \begin{vmatrix} \psi_1(x_1) & \dots & \psi_d(x_1) \\ \vdots & \ddots & \vdots \\ \psi_1(x_d) & \dots & \psi_d(x_d) \end{vmatrix}.$$



Notice that here the normalization factor is chosen in such a way that, provided the wave functions  $\psi_i$ ,  $i = 1, \dots, d$ , are normalized and orthogonal to each other,  $\psi_a$  is normalized as well.

**Remark 3.13.** We can define a more general class of antisymmetric kernels. Let  $f: \mathbb{R} \rightarrow \mathbb{R}$  be a function, then

$$k_a(x, x') = \frac{1}{d!} \begin{vmatrix} f(|x_1 - x'_1|) & \dots & f(|x_1 - x'_d|) \\ \vdots & \ddots & \vdots \\ f(|x_d - x'_1|) & \dots & f(|x_d - x'_d|) \end{vmatrix}$$

defines an antisymmetric kernel. We call such a function  $k_a$  a Slater kernel. The Gaussian kernel can be obtained by setting  $f(r) = e^{-\frac{r^2}{2\sigma^2}}$  and the Laplacian kernel—using the 1-norm—by setting  $f(r) = e^{-\frac{r}{\sigma}}$ . Alternatively, kernels based on generalized Slater determinants could be constructed or by concatenating creation and annihilation operators, see also [11–13].

The advantage of the Slater determinant formulation is that we can compute it efficiently using matrix decomposition techniques, without having to iterate over all permutations, which would be clearly infeasible for higher-dimensional problems.

**Example 3.14.** In order to illustrate the difference between a standard Gaussian kernel  $k$  and its antisymmetrized counterpart  $k_a$ , we define an antisymmetric function  $f: \mathbb{R}^2 \rightarrow \mathbb{R}$  by  $f(x) = \sin(\pi(x_1 - x_2))$  and apply kernel ridge regression (see, e.g. [31]) to randomly sampled data points<sup>8</sup>. That is, we generate  $m$  data points  $x^{(i)}$  in  $\mathbb{X} = [-1, 1] \times [-1, 1]$  and compute  $y^{(i)} = f(x^{(i)})$ . We then try to recover  $f$  from the training data  $\{(x^{(i)}, y^{(i)})\}_{i=1}^m$ . Additionally, we define an augmented data set of size  $2m$  by adding the antisymmetrized data set, i.e.  $\{(x^{(i)}, y^{(i)})\}_{i=1}^m \cup \{(\pi(x^{(i)}), -y^{(i)})\}_{i=1}^m$ , where  $\pi = (1, 2)$  in cycle notation. The bandwidth of the kernel is set to  $\sigma = \frac{1}{2}$ . The results are shown in figure 3. We measure the root-mean-square error (RMSE)—averaged over 5000 runs—in the midpoints of a regular  $30 \times 30$  box discretization of the domain. Kernel ridge regression using  $k_a$  results in more accurate function approximations and is, for small  $m$ , numerically equivalent to kernel ridge regression using  $k$  applied to the augmented data set of size  $2m$ . For larger values of  $m$ , doubling the size of the data set leads to ill-conditioned matrices and increased numerical errors<sup>9</sup>.  $\triangle$

The example shows that the antisymmetrized kernel is indeed advantageous, it enables a more accurate representation without increasing the size of the data set. For higher-dimensional problems, this effect will be even more pronounced. To obtain the same accuracy for a three-dimensional antisymmetric function, we would already need  $6m$  data points. The kernel evaluations, on the other hand, become more expensive, but are easily parallelizable. The bottleneck of kernel-based methods is often the size of the training data set, which enters in a cubic way (since a generally dense system of linear equations has to be solved, or, if we are interested in eigenfunctions of operators associated with dynamical systems, a generalized eigenvalue problem).

<sup>8</sup> We use a bold  $\pi$  for the mathematical constant to avoid confusion with permutations  $\pi$ .

<sup>9</sup> This could be mitigated by decreasing the bandwidth or by regularization techniques.

### 3.4. Derivatives of antisymmetric kernels

For the approximation of differential operators, we will also need partial derivatives of the kernel  $k_a$ . Since  $k_a$  just comprises alternating sums of kernel functions  $k$ , we can compute derivatives of  $k_a$  by summing over derivatives of  $k$ . For polynomial and Gaussian kernels, the derivatives of  $k$  can be found in [25]. Alternatively, the partial derivatives of the antisymmetric Gaussian kernel can be computed via Slater determinants.

**Example 3.15.** For the antisymmetric Gaussian kernel, let  $K^{e_l} \in \mathbb{R}^{d \times d}$  be the matrix with entries

$$K_{ij}^{e_l} = \begin{cases} -\frac{1}{\sigma^2}(x_i - x'_j)e^{-\frac{(x_i - x'_j)^2}{2\sigma^2}}, & i = l, \\ e^{-\frac{(x_i - x'_j)^2}{2\sigma^2}}, & i \neq l. \end{cases}$$

Then

$$\mathcal{D}^{e_l} k_a(x, x') = \frac{1}{d!} \det(K^{e_l}).$$

Similar formulas can be derived for the second-order derivatives. △

## 4. Symmetric kernels and their properties

Although we focused on antisymmetric functions so far, symmetric functions also play an important role in quantum physics. Other typical applications include point clouds, sets, and graphs, where the numbering of points, elements, or vertices should not impair the learning algorithms. Some of the above results can be easily carried over to the symmetric case. The special case  $d_x = 2$  is analyzed in [27]. Similar symmetrized kernels are also constructed in [6]. We focus on the analysis of the induced functions spaces.

### 4.1. Symmetric kernels

We call a function  $f: \mathbb{X} \rightarrow \mathbb{R}$  symmetric if

$$f(x) = f(\pi(x))$$

for all permutations  $\pi \in S_d$  and define the *symmetrization operator*

$$(Sf)(x) = \frac{1}{d!} \sum_{\pi \in S_d} f(\pi(x)).$$

**Definition 4.1** (Symmetric kernel function). Let  $k: \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$  be a kernel. We then define a symmetric function  $k_s: \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$  by

$$k_s(x, x') = \frac{1}{d!^2} \sum_{\pi \in S_d} \sum_{\pi' \in S_d} k(\pi(x), \pi'(x')).$$

We simply omitted the signs of the permutations here. As before, if  $k(x, x') = k(x', x)$ , then also  $k_s(x, x') = k_s(x', x)$ . The function  $k_s$  is permutation-symmetric in both arguments. Note that the definition of permutation-symmetry is different from permutation-invariance, which was defined by  $k(x, x') = k(\pi(x), \pi(x'))$ . Permutation-symmetric kernels are, however, automatically permutation-invariant. We briefly restate the above results for symmetric functions, the proofs are analogous to their counterparts for antisymmetric functions.

**Lemma 4.2.** The function  $k_s$  defines an s.p.d. kernel.

**Example 4.3.** For  $\mathbb{X} \subset \mathbb{R}^2$ , the feature space of the symmetrized polynomial kernel of order 2 is spanned by the symmetric functions  $\{1, x_1 + x_2, x_1^2 + x_2^2, x_1 x_2\}$ . △

More general results for polynomial kernels will be derived in section 4.2. Eigenfunctions of the integral operator associated with  $k_s$  are symmetric. Mercer features of the symmetrized Gaussian kernel for  $d = 2$  are shown in figure 2.

**Lemma 4.4.** Given a permutation-invariant kernel  $k$ , it holds that

$$k_s(x, x') = \frac{1}{d!} \sum_{\pi \in S_d} k(\pi(x), x') = \frac{1}{d!} \sum_{\pi \in S_d} k(x, \pi(x')).$$

**Table 2.** Dimensions of the feature spaces spanned by the polynomial kernel  $k$  and its symmetric counterpart  $k_s$ . Here,  $d$  is again the dimension of the state space and  $p$  the degree, cf table 1.

$d \backslash p$	2		3		4		5		6		7		8	
	$n_\phi$	$n_{\phi_s}$	$n_\phi$	$n_{\phi_s}$	$n_\phi$	$n_{\phi_s}$	$n_\phi$	$n_{\phi_s}$	$n_\phi$	$n_{\phi_s}$	$n_\phi$	$n_{\phi_s}$	$n_\phi$	$n_{\phi_s}$
2	6	4	10	6	15	9	21	12	28	16	36	20	45	25
3	10	4	20	7	35	11	56	16	84	23	120	31	165	41
4	15	4	35	7	70	12	126	18	210	27	330	38	495	53

Analogously, continuous symmetric functions can be approximated arbitrarily well by symmetric universal kernels.

**Proposition 4.5.** *Let  $\mathbb{X}$  be bounded. Given a universal, permutation-invariant, continuous kernel  $k$ , the space  $\mathbb{H}_s$  induced by  $k_s$  is dense in the space of continuous symmetric functions given by  $C_s(\mathbb{X}) = \{f \in C(\mathbb{X}) \mid f \text{ is symmetric}\}$ .*

### 4.2. Symmetric polynomial kernels

Let us compute the dimensions of the feature spaces spanned by symmetrized polynomial kernels.

**Lemma 4.6.** *The dimension of the feature space generated by the symmetrized polynomial kernel of order  $p$  is*

$$n_{\phi_s} = \sum_{p_r=0}^p \sum_{j=0}^d s_j(p_r).$$

*Proof.* Let  $\pi$  be a permutation, then the multi-indices  $q$  and  $\pi(q)$  generate the same feature space function when we apply the symmetrization operator  $\mathcal{S}$  to the corresponding monomials  $x^q$  and  $x^{\pi(q)}$ . We thus have to consider only partitions  $\mu$  of the integers  $0 \leq |q| \leq p$  since the ordering of the multi-indices does not matter.  $\square$

This case is similar to the antisymmetric case, with the difference that we require partitions of integers up to  $p$  instead of  $p - \binom{d}{2}$ . Table 2 lists the dimensions of the feature spaces spanned by the polynomial kernel  $k$  and its symmetric version  $k_s$  for different combinations of  $d$  and  $p$ . Compared to the standard polynomial kernel, the number of features is significantly lower, but higher than the number of features generated by the antisymmetric polynomial kernel.

### 4.3. Symmetric Gaussian kernels

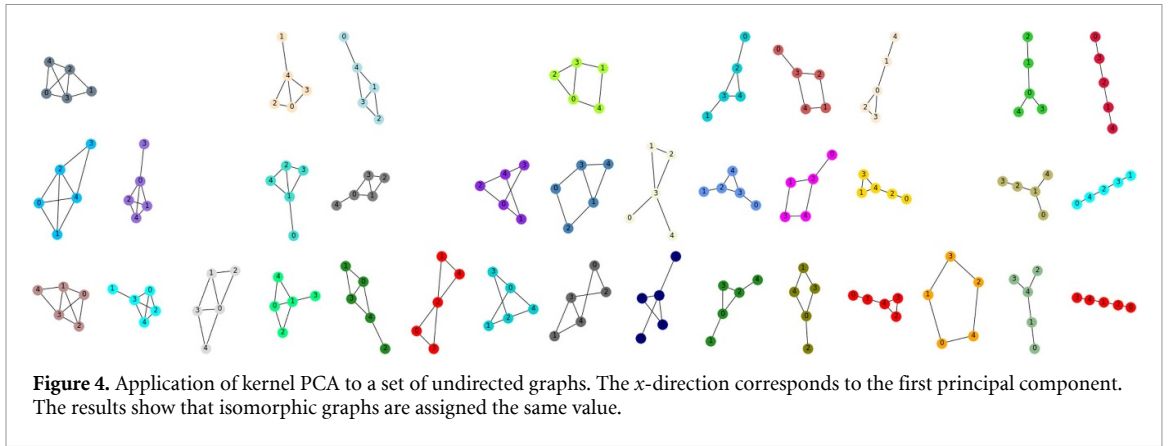
The symmetric kernel cannot be expressed as a Slater determinant anymore, but we can utilize a related concept. The *permanent* of a matrix  $A \in \mathbb{R}^{d \times d}$  is defined by

$$\text{per}(A) = \sum_{\pi \in \mathcal{S}_d} \prod_{i=1}^d a_{\pi(i),i} =: \begin{vmatrix} a_{11} & \dots & a_{1d} \\ \vdots & \ddots & \vdots \\ a_{d1} & \dots & a_{dd} \end{vmatrix}.$$

While for  $d = 2$  the permanent can be written as a determinant (by flipping the sign of  $a_{12}$  or  $a_{21}$ ), this is not possible anymore for  $d \geq 3$  [35]. No polynomial-time algorithm for the computation of the permanent is known, but there are efficient approximation schemes for matrices with non-negative entries [36].

**Lemma 4.7.** *Let  $k$  be the Gaussian kernel with bandwidth  $\sigma$ , then*

$$k_s(x, x') = \frac{1}{d!} \begin{vmatrix} e^{-\frac{(x_1-x'_1)^2}{2\sigma^2}} & \dots & e^{-\frac{(x_1-x'_d)^2}{2\sigma^2}} \\ \vdots & \ddots & \vdots \\ e^{-\frac{(x_d-x'_1)^2}{2\sigma^2}} & \dots & e^{-\frac{(x_d-x'_d)^2}{2\sigma^2}} \end{vmatrix}.$$



*Proof.* The proof is analogous to the one for lemma 3.12. Using the definition of the permanent, we obtain

$$\begin{aligned}
 \begin{vmatrix} e^{-\frac{(x_1-x'_1)^2}{2\sigma^2}} & \dots & e^{-\frac{(x_1-x'_d)^2}{2\sigma^2}} \\ \vdots & \ddots & \vdots \\ e^{-\frac{(x_d-x'_1)^2}{2\sigma^2}} & \dots & e^{-\frac{(x_d-x'_d)^2}{2\sigma^2}} \end{vmatrix} &= \sum_{\pi \in S_d} \prod_{i=1}^d e^{-\frac{(x_{\pi(i)}-x'_i)^2}{2\sigma^2}} \\ &= \sum_{\pi \in S_d} e^{-\frac{\sum_{i=1}^d (x_{\pi(i)}-x'_i)^2}{2\sigma^2}} \\ &= \sum_{\pi \in S_d} e^{-\frac{\|\pi(x)-x'\|^2}{2\sigma^2}} \\ &= \sum_{\pi \in S_d} k(\pi(x), x').
 \end{aligned}$$

The result then follows from lemma 4.4. □

**Example 4.8.** Assume we have a set of undirected graphs that we would like to classify or categorize. The results should not depend on the vertex labels and thus be identical for isomorphic graphs. Let  $A, A' \in \mathbb{R}^{d \times d}$  be the adjacency matrices of the graphs  $G$  and  $G'$ , respectively. We define a Gaussian kernel for graphs by

$$k(G, G') = \exp\left(-\frac{\|A - A'\|_F^2}{2\sigma^2}\right),$$

where  $\|\cdot\|_F$  denotes the Frobenius norm, and make it symmetric as described above. The only difference here is that we have to define  $\pi(A) = (a_{\pi(i), \pi(j)})_{i,j=1}^d$  to permute rows and columns simultaneously. The kernel function  $k_s(G, G')$  can then be expressed in terms of so-called hyperpermanents. We have

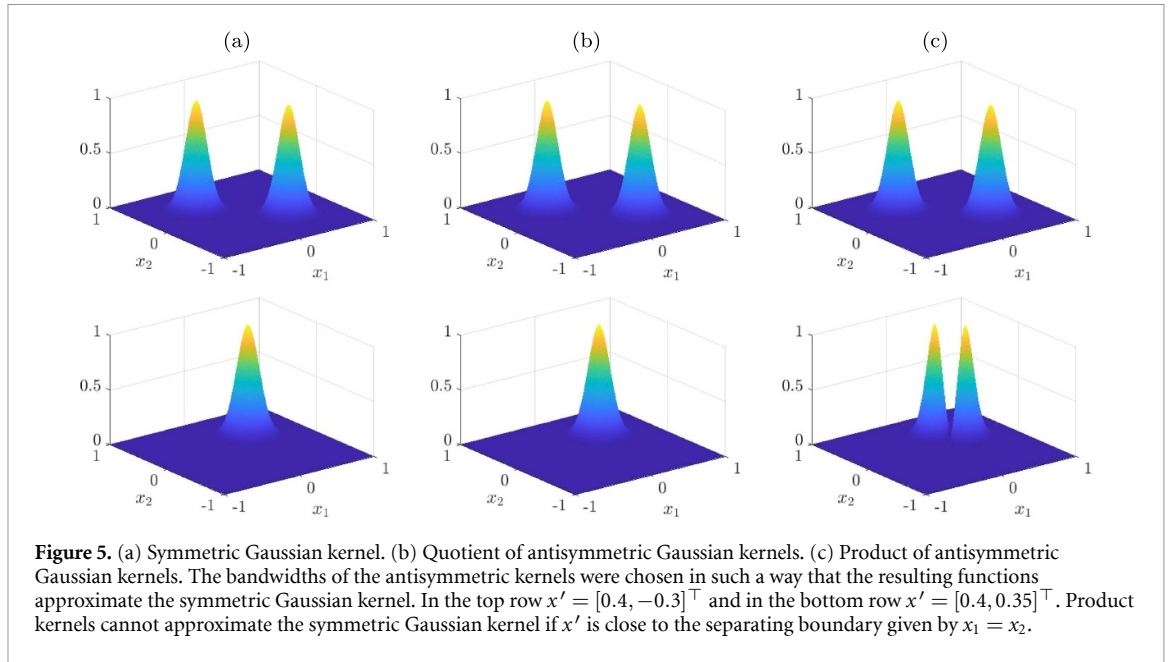
$$k_s(G, G') = \sum_{\pi \in S_d} \exp\left(-\frac{\|\pi(A) - A'\|_F^2}{2\sigma^2}\right) = \sum_{\pi \in S_d} \prod_{i=1}^d \prod_{j=1}^d \exp\left(-\frac{(a_{\pi(i), \pi(j)} - a'_{i,j})^2}{2\sigma^2}\right).$$

The derivation of a formula for the Laplace expansion of hyperpermanents can be found in appendix A.1. For the considered example, we set  $\sigma = 1$  and randomly generate a set of 100 undirected connected graphs of size  $d = 5$ . We then apply kernel PCA, see [17], using the symmetric kernel  $k_s$ . Sorting the graphs according to the first principal component, we obtain the ordering shown in figure 4 (only a subset of the graphs is displayed). Isomorphic graphs are grouped into the same category. △

Other learning algorithms such as kernel  $k$ -means, kernel ridge regression, or support vector machines can be used in the same way, enabling us to cluster, make predictions for, or classify data where the order of elements is irrelevant.

#### 4.4. Product or quotient representations of symmetric kernels

The aim now is to express a symmetric kernel not as a Slater permanent but as a product or quotient of antisymmetric functions. As shown in section 3.1, an antisymmetric kernel is zero for all  $x$  for which a (non-trivial) permutation  $\pi$  exists such that  $\pi(x) = x$ . Therefore, products of antisymmetric kernels are zero



for such  $x$  as well, see also figure 5. We thus mainly restrict ourselves to quotients. Let  $k_a^{(1)}$  and  $k_a^{(2)}$  be two permutation-invariant antisymmetric kernels and  $k_a^{(2)}(x, x') \neq 0$ . We define

$$k_s(x, x') = \frac{k_a^{(1)}(x, x')}{k_a^{(2)}(x, x')}.$$

Then

$$k_s(\pi(x), x') = \frac{k_a^{(1)}(\pi(x), x')}{k_a^{(2)}(\pi(x), x')} = \frac{\text{sgn}(\pi)k_a^{(1)}(x, x')}{\text{sgn}(\pi)k_a^{(2)}(x, x')} = k_s(x, x').$$

**Remark 4.9.** If the numerator and denominator can be written as determinants, i.e.  $k_a^{(1)}(x, x') = \det(K_1)$  and  $k_a^{(2)}(x, x') = \det(K_2)$ , we obtain

$$k_s(x, x') = \frac{\det(K_1)}{\det(K_2)} = \det(K_1) \det(K_2^{-1}) = \det(K_1 K_2^{-1}).$$

**Example 4.8.** Suppose  $d = 2$ . Let  $k^{(1)}$  and  $k^{(2)}$  be two Gaussian kernels with bandwidths  $\sigma_1$  and  $\sigma_2$ , respectively, where  $\sigma_1 < \sigma_2$ . If the bandwidths are sufficiently small, either  $k^{(1)}(x, x')$  and  $k^{(2)}(x, x')$  or  $k^{(1)}(\pi(x), x')$  and  $k^{(2)}(\pi(x), x')$  will be close to zero (unless  $\pi(x) = x$ ), where  $\pi = (1, 2)$  in cycle notation. Assume w.l.o.g. the latter holds, then

$$k_s(x, x') = \frac{k_a^{(1)}(x, x')}{k_a^{(2)}(x, x')} = \frac{k^{(1)}(x, x') - k^{(1)}(\pi(x), x')}{k^{(2)}(x, x') - k^{(2)}(\pi(x), x')} \approx \frac{k^{(1)}(x, x')}{k^{(2)}(x, x')},$$

which is a Gaussian with bandwidth  $\sigma$  satisfying  $\frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}$ . This is illustrated in figure 5. Furthermore, the limit of  $k_s(x, x')$  as  $x_2 \rightarrow x_1$  exists and is given by

$$k_s(x, x') = \frac{\sigma_2^2 k^{(1)}(x, x')}{\sigma_2^1 k^{(2)}(x, x')},$$

with  $x = [x_1, x_1]^T$ , see appendix B. △

The symmetric Gaussian kernel can be approximated by a quotient of antisymmetric Gaussian kernels, which can be evaluated in  $\mathcal{O}(d^3)$ , thus avoiding the non-polynomial complexity of the permanent. The question whether such kernels are universal is beyond the scope of this work.

## 5. Applications

In addition to the guiding examples presented above, we will illustrate the efficacy of the derived kernels with the aid of quantum physics and chemistry problems.

### 5.1. Particles in a one-dimensional box

Let us first consider a simple one-dimensional two-particle system. We define a potential  $V$  by

$$V(x) = \begin{cases} 0, & 0 \leq x \leq L, \\ \infty, & \text{otherwise.} \end{cases}$$

Furthermore, we assume that the two particles do not interact and obtain the Schrödinger equation

$$-\frac{\hbar^2}{2\mathbf{m}}\Delta\psi(x_1, x_2) = E\psi(x_1, x_2) \tag{3}$$

for  $0 \leq x_1, x_2 \leq L$ . By separating the two variables, we obtain the classical particle in a box problem, with eigenvalues  $E_\ell = \frac{\hbar^2 \pi^2 \ell^2}{2\mathbf{m}L^2}$  and eigenfunctions  $\psi_\ell(x) = \sqrt{\frac{2}{L}} \sin\left(\frac{\pi\ell x}{L}\right)$ , for  $\ell = 1, 2, 3, \dots$ , see, for instance, [37]. For the two-particle system, the eigenvalues are hence of the form

$$E_{\ell_1, \ell_2} = E_{\ell_1} + E_{\ell_2} = \frac{\hbar^2 \pi^2 (\ell_1^2 + \ell_2^2)}{2\mathbf{m}L^2}$$

and the eigenfunctions are

$$\psi_{\ell_1, \ell_2}(x_1, x_2) = \psi_{\ell_1}(x_1)\psi_{\ell_2}(x_2) = \frac{2}{L} \sin\left(\frac{\pi\ell_1 x_1}{L}\right) \sin\left(\frac{\pi\ell_2 x_2}{L}\right).$$

However, since the two particles are physically indistinguishable, the wave functions must satisfy  $|\psi_{\ell_1, \ell_2}(x_1, x_2)|^2 = |\psi_{\ell_1, \ell_2}(x_2, x_1)|^2$ , which implies that the functions are either symmetric (if the particles are bosons) or antisymmetric (if the particles are fermions). Let us assume that the two particles are electrons, i.e. fermions. We thus want to compute antisymmetric solutions of the time-independent Schrödinger equation by applying the approach introduced in section 2.2, see also [25]. In the same way, we could assume that the particles are bosons and compute symmetric solutions by replacing the antisymmetric kernel by a symmetric kernel.

We define  $\hbar = 1$ ,  $\mathbf{m} = 1$ , and  $L = \pi$ , choose the antisymmetric Gaussian kernel with bandwidth  $\sigma = 0.1$ , and generate  $m = 900$  uniformly sampled points in  $[0, L] \times [0, L]$ . Additionally, to ensure that the eigenfunctions are zero outside the box, we place 124 equidistantly distributed test points on the boundary and enforce  $\psi_{\ell_1, \ell_2}(x_1, x_2) = 0$  for these boundary points. We thus have to solve a constrained eigenvalue problem and use the algorithm described in [38]. The first three eigenfunctions  $\psi_{1,2}$ ,  $\psi_{1,3}$ , and  $\psi_{2,3}$  are shown in figure 6 and good approximations of the analytically computed eigenfunctions. The probability that the two electrons are in the same location is always zero. Furthermore, the results show that by increasing the number of data points we obtain more accurate and less noisy estimates of the true eigenvalues.

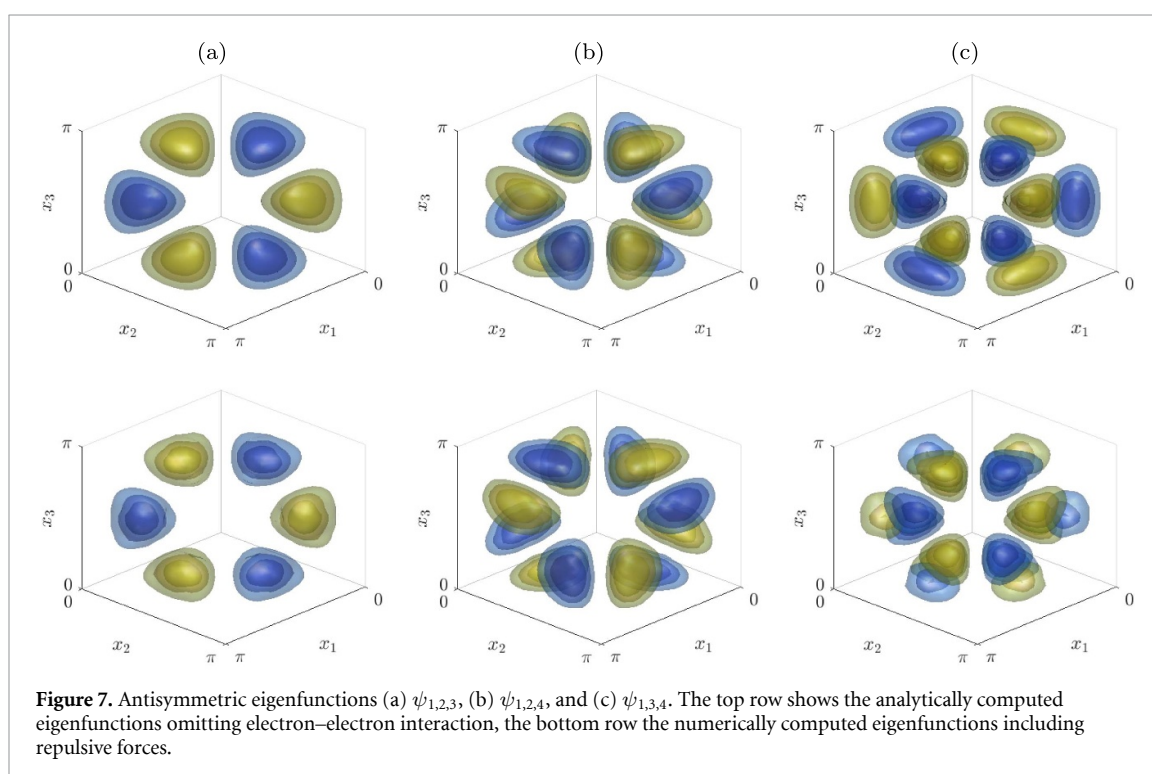
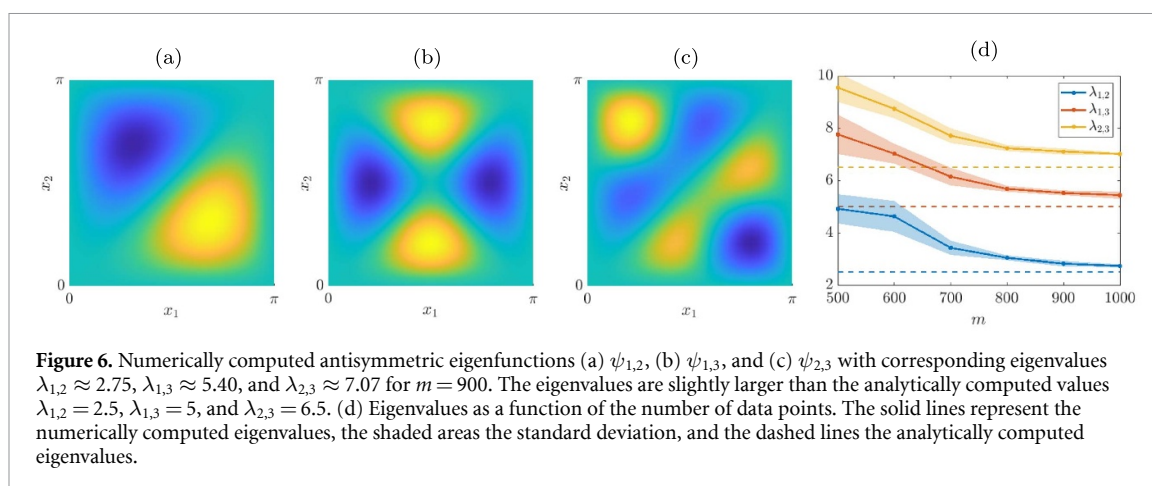
**Remark 5.1.** *We would like to point out that*

- *This example is just meant as an illustration of the concepts and not as a realistic physical model;*
- *Eigenfunctions with  $\ell_1 = \ell_2$  are symmetric and eliminated by the antisymmetrization operation;*
- *Approximations of the eigenfunctions can be obtained using far fewer points ( $m < 50$ ), but the eigenvalues will be considerably overestimated (kernels tailored to quantum mechanics applications might lead to better approximations);*
- *The antisymmetry assumption is encoded only in the kernel, not in the Schrödinger equation itself.*

This can be easily extended to the multi-particle case. We now add electron–electron interaction terms resulting in the Hamiltonian

$$\mathcal{H} = -\frac{\hbar^2}{2\mathbf{m}}\Delta + \sum_{i \neq j} \frac{1}{|x_i - x_j|}.$$

For  $d = 3$ , we randomly generate 3000 interior points and 600 boundary points to enforce Dirichlet boundary conditions. We choose a Gaussian kernel with bandwidth  $\sigma = 0.1$ , assemble the Gram matrices,



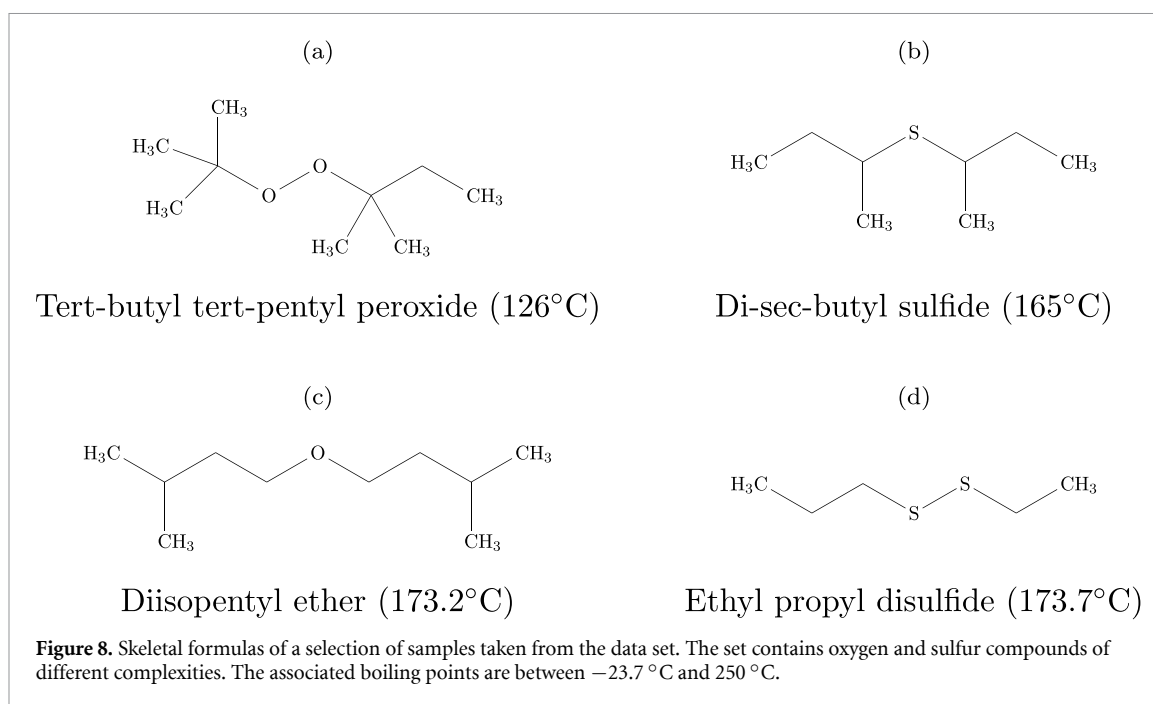
and again solve the resulting constrained eigenvalue problem. The results are shown in figure 7. For the sake of comparison, we also plot the corresponding eigenfunctions of the Schrödinger equation without the electron–electron interaction. It can be seen that the eigenfunctions for the separable case are similar to the eigenfunctions where the interaction terms are included. For this particular system, the interaction terms do not seem to have a drastic effect on the system’s low-lying energy states. In general, however, their effect on the electronic wavefunction can be significant. We also remark that energies and wavefunctions of the interacting system could in principle be approximated by perturbation techniques. However, due to the degeneracy of the antisymmetric states, such a perturbation analysis seems beyond the scope of this work.

## 5.2. Acyclic molecules

As a second example, we consider a data set of acyclic molecules [39]. The aim is to determine the boiling points of these molecules containing the elements C, H, O, and S. The data set<sup>10</sup> consists of 183 graphs  $G = (V, E)$  representing the molecular structures and the corresponding boiling points in degrees Celsius, see figure 8 for a few examples of molecules included in the data set. The number of vertices  $|V|$  varies between 3 and 11, where the hydrogen atoms of the molecules are neglected. Thus, in order to compare the graphs of

<sup>10</sup> The data set can be found at <https://brunl01.users.greyc.fr/CHEMISTRY/>.





different sizes, we expand all adjacency matrices to  $\mathbb{R}^{d \times d}$  with  $d = 11$  by appending rows and columns of zeros, representing artificial isolated nodes.

We define a symmetrized Laplacian kernel on graphs, cf example 4.8. Given the adjacency matrices  $A, A' \in \mathbb{R}^{d \times d}$  of the graphs  $G = (V, E)$  and  $G' = (V', E')$  as well as the kernel parameter  $\sigma > 0$ , we define the tensor  $T \in \mathbb{R}^{d \times d \times d \times d}$  by

$$t_{i,j,k,l} = \exp\left(-\frac{|a_{i,j} - a'_{k,l}|}{\sigma}\right)$$

for  $i \neq j$  and  $k \neq l$  and

$$t_{i,i,k,k} = \begin{cases} 1, & \text{if } V_i \text{ and } V'_k \text{ represent the same atom,} \\ \exp(-1/\sigma), & \text{otherwise.} \end{cases}$$

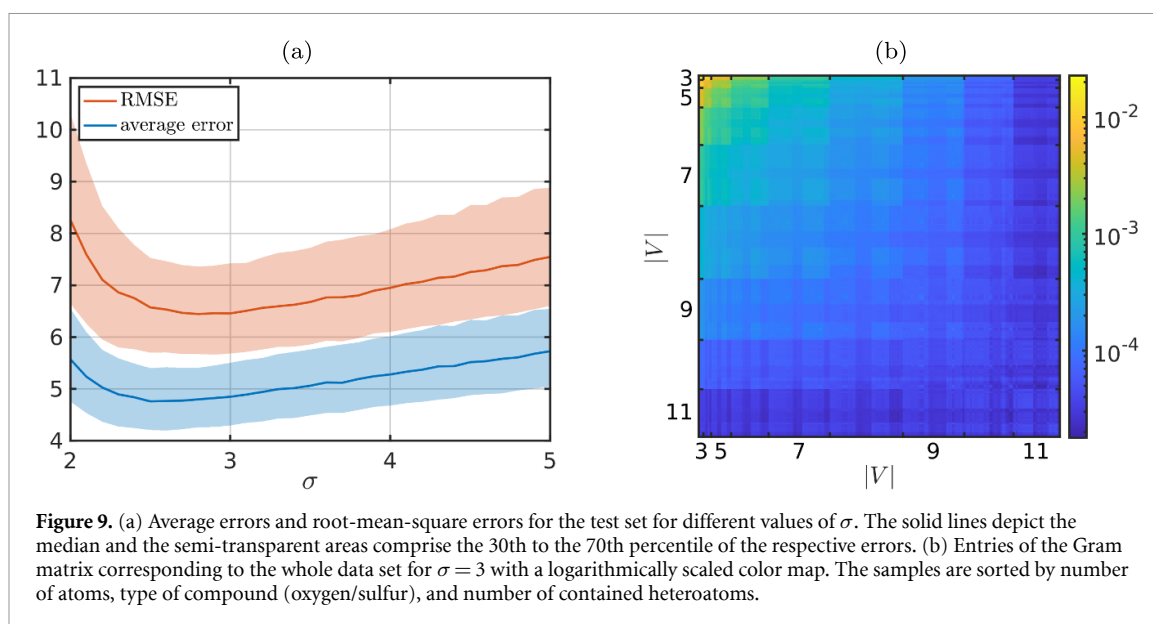
The latter definition ensures that we avoid unwanted effects of any ordinal labeling of the nodes. Using the hyperpermanent of  $T$ , the kernel evaluation  $k_s(G, G')$  can be written as

$$k_s(G, G') = \text{hper}(T) = \sum_{\pi \in S_d} \prod_{i=1}^d \prod_{j=1}^d t_{i,j,\pi(i),\pi(j)}. \quad (4)$$

Note that we do not consider entries  $t_{i,j,k,l}$  with either  $i = j, k \neq l$  or  $i \neq j, k = l$  since  $i = j \Leftrightarrow \pi(i) = \pi(j)$ . We refer to appendix A for different methods and simplifications for computing the hyperpermanent of  $T$ .

For kernel-based (ridge) regression (see, e.g. [40]), we extract 165 adjacency matrices ( $\approx 90\%$ ) and their corresponding boiling point temperatures from the data set as training samples, the other data pairs constitute the test set. That is, for any  $G$  in the test set, the regression function is given by  $f(G) = \Theta^\top K_{\text{train},G}$ , where the vector  $K_{\text{train},G} \in \mathbb{R}^{165}$  is the Gram matrix (or *kernel matrix*) corresponding to the training samples (rows) and the test sample  $G$  (column). The vector  $\Theta \in \mathbb{R}^{165}$  is the solution of  $K_{\text{train},\text{train}}\Theta = b$  with  $b$  being the vector of boiling points of the molecules in the training set. We then compute the average error as well as the root-mean-square error in the boiling points of the test set in order to evaluate the generalizability of the learned regression function. We repeat each experiment 10 000 times with randomly chosen training and test sets, the results for different kernel parameters  $\sigma$  are shown in figure 9(a).

The best results in terms of the average and the root-mean-square error are obtained for kernel parameters  $\sigma$  between 2.5 and 2.8, see table 3 for details. According to the data set's website, the best results so far for the boiling point prediction on 90% of the set as training data and 10% as test data are achieved by applying so-called *treelet kernels* which exploit all possible graph/tree patterns up to a given size [39]. In this case, the average error is listed as 4.87 and the root-mean-square error as 6.75. Both values are comparable with our results.



**Figure 9.** (a) Average errors and root-mean-square errors for the test set for different values of  $\sigma$ . The solid lines depict the median and the semi-transparent areas comprise the 30th to the 70th percentile of the respective errors. (b) Entries of the Gram matrix corresponding to the whole data set for  $\sigma = 3$  with a logarithmically scaled color map. The samples are sorted by number of atoms, type of compound (oxygen/sulfur), and number of contained heteroatoms.

**Table 3.** Mean and median of the average error and root-mean-square error over all repetitions for values of  $\sigma$  between 2.5 and 2.8.

$\sigma$	Average error		Root-mean-square error	
	mean	median	mean	median
0.25	4.90	4.76	6.85	6.57
0.26	4.91	4.77	6.87	6.53
0.27	4.92	4.77	6.88	6.47
0.28	4.94	4.80	6.91	6.44

As shown in figure 9(b), the entries of the Gram matrix tend to decrease for larger molecules. This effect can be explained by the expansion of the adjacency matrices and similarities of the molecules with small numbers of atoms. For instance, the first two compounds in the (ordered) data set are dimethyl ether ( $C_2H_6O$ ) and dimethyl sulfide ( $C_2H_6S$ ). Due to the expansion from  $|V| = 3$  to  $|V| = 11$ , the majority of the permutations in (4) do not affect the adjacency matrix of  $G'$ , cf appendix A.3. The block structure of the matrix arises from the ordering of the data set, i.e. each group of molecules with the same number of atoms is divided into subgroups of compounds containing one oxygen atom, two oxygen atoms, one sulfur atom, and two sulfur atoms.

## 6. Conclusion

We derived symmetric and antisymmetric kernels that can be used in kernel-based learning algorithms such as kernel PCA, kernel CCA, or support vector machines, but also to approximate symmetric or antisymmetric eigenfunctions of transfer operators or differential operators (e.g. the Koopman generator or Schrödinger operator). Potential applications range from point cloud analysis and graph classification to quantum physics and chemistry. Furthermore, we analyzed the induced reproducing kernel Hilbert spaces and resulting feature space dimensions. The effectiveness of the proposed kernels was demonstrated using guiding examples and simple benchmark problems.

The next step is now to apply kernel-based methods to more complex quantum systems. Such problems might require kernels tailored to the system at hand. By exploiting additional properties (sparsity, low-rank structure, weak coupling between subsystems), it could be possible to improve the performance of kernel-based methods. Furthermore, the kernel flow approach proposed in [41] could be extended to operator estimation problems. This would allow us to also learn the kernel from data.

Another topic for future research would be to consider other types of symmetries and to develop kernels that explicitly take these properties into account. While the antisymmetric kernel can be evaluated efficiently using matrix factorizations, this is not possible for the symmetric kernel, which requires the evaluation of a

matrix permanent. Utilizing efficient approximation schemes could speed up the generation of the required Gram matrices significantly. Alternatively, the product or quotient formulation of symmetric kernels could be exploited to facilitate the application of the proposed methods to higher-dimensional problems.

### Data availability statement

The data and code that support the findings of this study are openly available at <https://github.com/sklus/d3s/>.

### Acknowledgments

We would like to thank Jan Hermann for helpful discussions about quantum chemistry and the reviewers for their helpful comments and suggestions.

### Funding

P Gelß and F Noé have been partially funded by Deutsche Forschungsgemeinschaft (DFG) through grant CRC 1114 ‘Scaling Cascades in Complex Systems’ (Project ID: 23 522 1301, projects A04 and B06). F Noé also acknowledges funding from BMBF through the Berlin Institute for the Foundations of Learning and Data (BIFOLD), the European Commission (ERC CoG 772 230), and the Berlin Mathematics center MATH+ (AA2-8).

## Appendix A. Hyperpermanents

Given a tensor  $T \in \mathbb{R}^{d \times d \times d} = \mathbb{R}^{d \times d \times d \times d}$ , the hyperpermanent of  $T$  is given by

$$\text{hper}(T) = \sum_{\pi \in S_d} \prod_{i=1}^d \prod_{j=1}^d t_{\pi(i), \pi(j), i, j} = \sum_{\pi \in S_d} \prod_{i=1}^d \prod_{j=1}^d t_{i, j, \pi(i), \pi(j)}.$$

In example 4.8, we considered tensor entries of the form

$$t_{i, j, k, l} = \exp\left(-\frac{(a_{i, j} - a'_{k, l})^2}{2\sigma^2}\right) \tag{5}$$

for adjacency matrices  $A, A' \in \mathbb{R}^{d \times d}$  in order to construct the symmetrized Gaussian kernel for graphs. Another example for defining the entries of  $T$ , as described in section 5.2, is

$$t_{i, j, k, l} = \exp\left(-\frac{|a_{i, j} - a'_{k, l}|}{\sigma}\right), \tag{6}$$

which results in the symmetrized Laplacian kernel for graphs. For both choices, we set  $t_{i, i, k, k} = \exp(-1/2\sigma^2)$  and  $t_{i, i, k, k} = \exp(-1/\sigma)$ , respectively, if  $a_{i, i} \neq a'_{k, k}$  and  $t_{i, i, k, k} = \exp(0)$ , otherwise. In what follows, we will consider different techniques for computing the hyperpermanent of  $T$ .

#### A.1. Laplace expansion for the computation of hyperpermanents

Define  $T^{(\mu)} \in \mathbb{R}^{d \times d \times d}$  by

$$t_{i, j, k, l}^{(\mu)} = \begin{cases} t_{i, j, k, l} \cdot t_{i, 1, k, \mu} \cdot t_{1, j, \mu, l}, & \text{if } i = j \text{ and } k = l, \\ t_{i, j, k, l}, & \text{otherwise.} \end{cases}$$

Let  $\widehat{T}^{(\mu)} \in \mathbb{R}^{(d-1) \times (d-1) \times (d-1)}$  denote the tensor that results from  $T^{(\mu)}$  by removing all entries  $t_{i, j, k, l}^{(\mu)}$  with  $i = 1, j = 1, k = \mu, \text{ or } l = \mu$ . The hyperpermanent can then be written as

$$\begin{aligned}
 \text{hper}(T) &= \sum_{\pi \in S_d} \prod_{i=1}^d \prod_{j=1}^d t_{i,j,\pi(i),\pi(j)} \\
 &= \sum_{\pi \in S_d} t_{1,1,\pi(1),\pi(1)} \prod_{i=2}^d t_{i,1,\pi(i),\pi(1)} \prod_{j=2}^d t_{1,j,\pi(1),\pi(j)} \prod_{i=2}^d \prod_{j=2}^d t_{i,j,\pi(i),\pi(j)} \\
 &= \sum_{\mu=1}^d t_{1,1,\mu,\mu} \sum_{\substack{\pi \in S_d \\ \pi(1)=\mu}} \prod_{i=2}^d \prod_{j=2}^d t_{i,j,\pi(i),\pi(j)} (t_{i,1,\pi(i),\mu} t_{1,j,\mu,\pi(j)})^{\delta_{ij}} \\
 &= \sum_{\mu=1}^d t_{1,1,\mu,\mu} \text{hper}(\widehat{T}^{(\mu)}),
 \end{aligned}$$

where  $\delta_{ij}$  denotes the Kronecker delta. Note that  $i = j$  implies  $\pi(i) = \pi(j)$ .

**A.2. Hyperpermanents of pairwise symmetric tensors**

Suppose  $t_{i,j,k,l} = t_{j,i,k,l}$  and  $t_{i,j,k,l} = t_{i,j,l,k}$  for all  $i, j, k, l \in \{1, \dots, d\}$ . For instance, this is the case for tensors  $T$  containing elementwise evaluations of Gaussian and Laplacian kernels as given in (5) and (6), respectively. The hyperpermanent of  $T$  can then be written as

$$\begin{aligned}
 \text{hper}(T) &= \sum_{\pi \in S_d} \prod_{i=1}^d \prod_{j=1}^d t_{i,j,\pi(i),\pi(j)} \\
 &= \sum_{\pi \in S_d} \prod_{i=1}^d \left( t_{i,i,\pi(i),\pi(i)} \prod_{\substack{j=1 \\ j \neq i}}^d t_{i,j,\pi(i),\pi(j)} \right) \\
 &= \sum_{\pi \in S_d} \left( \prod_{i=1}^d t_{i,i,\pi(i),\pi(i)} \right) \left( \prod_{\substack{i=1 \\ j \neq i}}^d \prod_{j=1}^d t_{i,j,\pi(i),\pi(j)} \right) \tag{7} \\
 &= \sum_{\pi \in S_d} \left( \prod_{i=1}^d t_{i,i,\pi(i),\pi(i)} \right) \left( \prod_{i=1}^d \prod_{j=i+1}^d t_{i,j,\pi(i),\pi(j)}^2 \right) \\
 &= \sum_{\pi \in S_d} \frac{\prod_{i=1}^d \prod_{j=i}^d t_{i,j,\pi(i),\pi(j)}^2}{\prod_{i=1}^d t_{i,i,\pi(i),\pi(i)}}.
 \end{aligned}$$

The advantage of the above formula is that we can reduce the computational costs for the hyperpermanent. Additionally, we do not have to compute all elements of  $T$ , which also reduces the computational costs.

**A.3. Hyperpermanents for graphs with isolated nodes**

In order to compare graphs of different sizes, we include artificial isolated nodes in section 5.2. That is, the adjacency matrix of a given graph is expanded by adding zero entries. In this case, all permutations among the isolated nodes do not change the result of the product  $\prod_{i=1}^d \prod_{j=1}^d t_{i,j,\pi(i),\pi(j)}$ . Assume that the dimension of the adjacency matrix  $A$  used in (5) and (6) is initially  $d' < d$  before the expansion to  $R^{d \times d}$ . Then, it holds that

$$t_{i,j,\pi(i),\pi(j)} = t_{d'+1,d'+1,\pi(i),\pi(j)},$$

if  $i > d'$  or  $j > d'$ . Thus, given two permutations  $\pi_1$  and  $\pi_2$  with

$$\pi_1(i) = \pi_2(i) \text{ for } 1 \leq i \leq d',$$

it follows that

$$\frac{\prod_{i=1}^d \prod_{j=i}^d t_{i,j,\pi_1(i),\pi_1(j)}^2}{\prod_{i=1}^d t_{i,i,\pi_1(i),\pi_1(i)}} = \frac{\prod_{i=1}^d \prod_{j=i}^d t_{i,j,\pi_2(i),\pi_2(j)}^2}{\prod_{i=1}^d t_{i,i,\pi_2(i),\pi_2(i)}}.$$

This means that for each permutation  $\pi \in S_d$ , any of the  $(d - d')!$  permutations of the set  $\{\pi(d' + 1), \dots, \pi(d)\}$  does not change the value of the quotient in (7). This fact can be exploited using the formula

$$\text{hper}(T) = \sum_{\pi \in S_d} \pi(d' + 1) < \dots < \pi(d) (d - d')! \frac{\prod_{i=1}^d \prod_{j=i}^d t_{i,j,\pi(i),\pi(j)}^2}{\prod_{i=1}^d t_{i,i,\pi(i),\pi(i)}},$$

which enables us to decrease the number of considered permutations significantly if  $d'$  is much smaller than  $d$ .

## Appendix B. Quotient representation of symmetric Gaussian kernels

Under the same assumptions as given in example 4.10, we use lemma 3.12 in order to write  $k_s(x, x')$  as

$$k_s(x, x') = \frac{k_a^{(1)}(x, x')}{k_a^{(2)}(x, x')} = \frac{e^{-\frac{(x_1-x'_1)^2}{2\sigma_1^2}} e^{-\frac{(x_2-x'_2)^2}{2\sigma_1^2}} - e^{-\frac{(x_1-x'_2)^2}{2\sigma_1^2}} e^{-\frac{(x_2-x'_1)^2}{2\sigma_1^2}}}{e^{-\frac{(x_1-x'_1)^2}{2\sigma_2^2}} e^{-\frac{(x_2-x'_2)^2}{2\sigma_2^2}} - e^{-\frac{(x_1-x'_2)^2}{2\sigma_2^2}} e^{-\frac{(x_2-x'_1)^2}{2\sigma_2^2}}},$$

where  $x_1 \neq x_2$  and  $x'_1 \neq x'_2$ . From l'Hôpital's rule follows that

$$\begin{aligned} \lim_{x_2 \rightarrow x_1} \frac{k_a^{(1)}(x, x')}{k_a^{(2)}(x, x')} &= \lim_{x_2 \rightarrow x_1} \frac{\frac{\partial}{\partial x_2} k_a^{(1)}(x, x')}{\frac{\partial}{\partial x_2} k_a^{(2)}(x, x')} \\ &= \lim_{x_2 \rightarrow x_1} \frac{-\frac{x_2-x'_2}{\sigma_1^2} e^{-\frac{(x_1-x'_1)^2}{2\sigma_1^2}} e^{-\frac{(x_2-x'_2)^2}{2\sigma_1^2}} + \frac{x_2-x'_1}{\sigma_1^2} e^{-\frac{(x_1-x'_2)^2}{2\sigma_1^2}} e^{-\frac{(x_2-x'_1)^2}{2\sigma_1^2}}}{-\frac{x_2-x'_2}{\sigma_2^2} e^{-\frac{(x_1-x'_1)^2}{2\sigma_2^2}} e^{-\frac{(x_2-x'_2)^2}{2\sigma_2^2}} + \frac{x_2-x'_1}{\sigma_2^2} e^{-\frac{(x_1-x'_2)^2}{2\sigma_2^2}} e^{-\frac{(x_2-x'_1)^2}{2\sigma_2^2}}} \\ &= \frac{\left(-\frac{x_1-x'_2}{\sigma_1^2} + \frac{x_1-x'_1}{\sigma_1^2}\right) e^{-\frac{(x_1-x'_1)^2}{2\sigma_1^2}} e^{-\frac{(x_1-x'_2)^2}{2\sigma_1^2}}}{\left(-\frac{x_1-x'_2}{\sigma_2^2} + \frac{x_1-x'_1}{\sigma_2^2}\right) e^{-\frac{(x_1-x'_1)^2}{2\sigma_2^2}} e^{-\frac{(x_1-x'_2)^2}{2\sigma_2^2}}} \\ &= \frac{\sigma_2^2 e^{-\frac{\|(x_1, x_1) - (x'_1, x'_2)\|^2}{2\sigma_1^2}}}{\sigma_1^2 e^{-\frac{\|(x_1, x_1) - (x'_1, x'_2)\|^2}{2\sigma_2^2}}}. \end{aligned}$$

## ORCID iDs

Stefan Klus  <https://orcid.org/0000-0002-9672-3806>

Patrick Gelß  <https://orcid.org/0000-0002-3645-9513>

Feliks Nüske  <https://orcid.org/0000-0003-2444-7889>

Frank Noé  <https://orcid.org/0000-0003-4169-9324>

## References

- [1] Li J, Jiang B and Guo H 2013 Permutation invariant polynomial neural network approach to fitting potential energy surfaces. II. Four-atom systems *J. Chem. Phys.* **139** 204103
- [2] Koner D and Meuwly M 2020 Permutationally invariant, reproducing kernel-based potential energy surfaces for polyatomic molecules: from formaldehyde to acetone *J. Chem. Theory Comput.* **16** 5474–84
- [3] Hutter M 2020 On representing (anti)symmetric functions (arXiv:2007.15298)
- [4] Qi C R, Su H, Mo K and Guibas L J 2017 Pointnet: deep learning on point sets for 3D classification and segmentation *2017 Conf. on Computer Vision and Pattern Recognition (IEEE Computer Society)* pp 77–85
- [5] Lee J, Lee Y, Kim J, Kosiorek A, Choi S and Teh Y 2019 Set transformer: a framework for attention-based permutation-invariant neural networks *J. Mach. Learn. Res.* **97** 3744–53
- [6] Uteva E, Graham R S, Wilkinson R D and Wheatley R J 2017 Interpolation of intermolecular potentials using Gaussian processes *J. Chem. Phys.* **147** 161706
- [7] Bartók A P, Gillan M J, Manby F R and Csányi G 2013 Machine-learning approach for one- and two-body corrections to density functional theory: applications to molecular and condensed water *Phys. Rev. B* **88** 054104
- [8] Bartók A P, Kondor R and Csányi G 2013 On representing chemical environments *Phys. Rev. B* **87** 184115
- [9] Haasdonk B and Burkhardt H 2007 Invariant kernel functions for pattern analysis and machine learning *Mach. Learn.* **68** 35–61
- [10] Han J, Zhang L and Weinan E 2019 Solving many-electron Schrödinger equation using deep neural networks *J. Comput. Phys.* **399** 108929
- [11] Pfau D, Spencer J S, Matthews A G D G and Foulkes W M C 2020 Ab initio solution of the many-electron Schrödinger equation with deep neural networks *Phys. Rev. Res.* **2** 033429

- [12] Hermann J, Schätzle Z and Noé F 2020 Deep-neural-network solution of the electronic Schrödinger equation *Nat. Chem.* **12** 891–7
- [13] Choo K, Mezzacapo A and Carleo G 2020 Fermionic neural-network states for ab-initio electronic structure *Nat. Commun.* **11** 2368
- [14] Schätzle Z, Hermann J and Noé F 2021 Convergence to the fixed-node limit in deep variational Monte Carlo *J. Chem. Phys.* **154** 124108
- [15] López Ríos P, Ma A, Drummond N D, Towler M D and Needs R J 2006 Inhomogeneous backflow transformations in quantum Monte Carlo calculations *Phys. Rev. E* **74** 066701
- [16] Han J, Li Y, Lin L, Lu J, Zhang J and Zhang L 2019 Universal approximation of symmetric and anti-symmetric functions (arXiv:1912.01765)
- [17] Schölkopf B, Smola A and Müller K-R 1998 Nonlinear component analysis as a kernel eigenvalue problem *Neural Comput.* **10** 1299–1319
- [18] Melzer T, Reiter M and Bischof H 2001 Nonlinear feature extraction using generalized canonical correlation analysis *Artificial Neural Networks—Icann 2001* ed G Dorffner, H Bischof and K Hornik (Berlin Heidelberg: Springer) pp 353–60
- [19] Steinwart I and Christmann A 2008 *Support Vector Machines* 1st edn (New York: Springer)
- [20] Schölkopf B and Smola A J 2001 *Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond* (Cambridge, MA: The MIT press)
- [21] Wendland H 2004 Scattered data approximation *Cambridge Monographs on Applied and Computational Mathematics* (Cambridge: Cambridge University Press)
- [22] Zhou D-X 2008 Derivative reproducing properties for kernel methods in learning theory *J. Comput. Appl. Math.* **220** 456–63
- [23] Micchelli C A, Xu Y and Zhang H 2006 Universal kernels *J. Mach. Learn. Res.* **7** 2651–67
- [24] Sriperumbudur B K, Fukumizu K and Lanckriet G R G 2011 Universality, characteristic kernels and RKHS embedding of measures *J. Mach. Learn. Res.* **12** 2389–410
- [25] Klus S, Nüske F and Hamzi B 2020 Kernel-based approximation of the Koopman generator and Schrödinger operator *Entropy* **22** 722
- [26] Baker C 1973 Joint measures and cross-covariance operators *Trans. Am. Math. Soc.* **186** 273–89
- [27] Pahikkala T, Viljanen M, Airola A and Waegeman W 2015 Spectral analysis of symmetric and anti-symmetric pairwise kernels (arXiv:1506.05950)
- [28] Gnecco G 2018 Symmetric and antisymmetric properties of solutions to kernel-based machine learning problems *Neurocomputing* **306** 141–59
- [29] Mollenhauer M, Schuster I, Klus S and Schütte C 2020 Singular value decomposition of operators on reproducing kernel Hilbert spaces *Advances in Dynamics, Optimization and Computation* ed O Junge, O Schütze, G Froyland, S Ober-Blöbaum and K Padberg-Gehle (Cham: Springer) pp 109–31
- [30] Zaki M J and Meira Jr W 2014 *Data Mining and Analysis: Fundamental Concepts and Algorithms* (Cambridge: Cambridge University Press)
- [31] Shawe-Taylor J and Cristianini N 2004 *Kernel Methods for Pattern Analysis* (Cambridge: Cambridge University Press)
- [32] Sturmfels B 2008 *Algorithms in Invariant Theory* 2nd edn (Vienna: Springer)
- [33] Stanley R P 2011 *Enumerative Combinatorics* vol 1 2nd edn (Cambridge: Cambridge University Press)
- [34] Foldy L L 1962 Antisymmetric functions and Slater determinants *J. Math. Phys.* **3** 531–9
- [35] Szegő G 1913 Lösung zu Aufgabe 424 *Arch. Math. Phys.* **21** 291–2
- [36] Kuck J, Dao T, Rezaatofghi H, Sabharwal A and Ermon S 2019 Approximating the permanent by sampling from adaptive partitions *Advances in Neural Information Processing Systems* vol 32
- [37] Hall B C 2013 *Quantum Theory for Mathematicians Graduate Texts in Mathematics* (New York: Springer)
- [38] Golub G H 1973 Some modified matrix eigenvalue problems *SIAM Rev.* **15** 318–34
- [39] Gaüzère B, Brun L and Villemin D 2012 Two new graphs kernels in chemoinformatics *Pattern Recognit. Lett.* **33** 2038–47
- [40] Murphy K P 2012 *Machine Learning: A Probabilistic Perspective* (Cambridge, MA: The MIT Press)
- [41] Owhadi H and Yoo G R 2019 Kernel flows: from learning kernels from data into the abyss *J. Comput. Phys.* **389** 22–47