On Representing (Anti)Symmetric Functions

Marcus Hutter

DeepMind & ANU

http://www.hutter1.net/

October 1, 2020

Abstract

Permutation-invariant, -equivariant, and -covariant functions and anti-symmetric functions are important in quantum physics, computer vision, and other disciplines. Applications often require most or all of the following properties: (a) a large class of such functions can be approximated, e.g. all continuous function, (b) *only* the (anti)symmetric functions can be represented, (c) a fast algorithm for computing the approximation, (d) the representation itself is continuous or differentiable, (e) the architecture is suitable for learning the function from data. (Anti)symmetric neural networks have recently been developed and applied with great success. A few theoretical approximation results have been proven, but many questions are still open, especially for particles in more than one dimension and the anti-symmetric case, which this work focusses on. More concretely, we derive natural polynomial approximations in the symmetric case, and approximations based on a *single* generalized Slater determinant in the anti-symmetric case. Unlike some previous super-exponential and discontinuous approximations, these seem a more promising basis for future tighter bounds. We provide a complete and explicit universality proof of the Equivariant MultiLayer Perceptron, which implies universality of symmetric MLPs and the FermiNet.

Contents

Keywords

Neural network, approximation, universality, Slater determinant, Vandermonde matrix, equivariance, symmetry, anti-symmetry, symmetric polynomials, polarized basis, multilayer perceptron, continuity, smoothness, ...

1 Introduction

Neural networks. Neural Networks (NN), or more precisely, Multi-Layer Perceptrons (MLP), are universal function approximators [Pin99] in the sense that every (say) continuous function can be approximated arbitrarily well by a sufficiently large NN. The true power of NN though stems from the fact that they apparently have a bias towards functions we care about and that they can be trained by local gradient-descent or variations thereof.

Covariant functions. For many problems we have additional information about the function, e.g. symmetries under which the function of interest is invariant or covariant. Here we consider functions that are covariant[1] under permutations.[2] Of particular interest are functions that are invariant[3], equivariant[4], or anti-symmetric[5] under permutations.

Definition 1 ((Anti)aymmetric and equivariant functions) *A function $\phi:\mathcal{X}^n\to\mathbb{R}$ in $n\in\mathbb{N}$ variables is called* symmetric *iff $\phi(x_1,...,x_n)=\phi(x_{\pi(1)},...,x_{\pi(n)})$ for all $x_1,...,x_n\in\mathcal{X}$ for all permutations $\pi\in S_n$, where $S_n:=\{\pi:\{1:n\}\to\{1:n\}\wedge\pi$ is bijection$\}$ is called the symmetric group and $\{1:n\}$ is short for $\{1,...,n\}$. Similarly, a function $\psi:\mathcal{X}^n\to\mathbb{R}$ is called* anti-symmetric (AS) *iff $\psi(x_1,...,x_n)=\sigma(\pi)\psi(x_{\pi(1)},...,x_{\pi(n)})$, where $\sigma(\pi)=\pm1$ is the parity or sign of permutation π. A function $\boldsymbol{\varphi}:\mathcal{X}^n\to\mathcal{X}'^n$ is called equivariant under permutations iff $\boldsymbol{\varphi}(S_\pi(\mathbf{x}))=S_\pi(\boldsymbol{\varphi}(\mathbf{x}))$, where $\mathbf{x}\equiv(x_1,...,x_n)$ and $S_\pi(x_1,...,x_n):=(x_{\pi(1)},...,x_{\pi(n)})$.*

Of course (anti)symmetric functions are also just functions, hence a NN of sufficient capacity can also represent (anti)symmetric functions, and if trained on an (anti)symmetric target could converge to an (anti)symmetric function. But NNs that can represent *only* (anti)symmetric functions are desirable for multiple reasons. Equivariant MLP (EMLP) are the basis for constructing symmetric functions by simply summing the output of the last layer, and for anti-symmetric (AS) functions by multiplying with Vandermonde determinants or by computing their generalized Slater determinant (GSD) defined later.

Applications. The most prominent application is in quantum physics which represents systems of identical (fermions) bosons with (anti)symmetric wave functions [PSMF20]. Another application is classification of point clouds in computer vision, which should be invariant under permutation of points [ZKR⁺18].

Exact (anti)symmetry. Even if a general NN can learn the (anti)symmetry, it will only do so approximately, but some applications require exact (anti)symmetry, for instance in quantum physics to guarantee upper bounds on the true ground state energy [PSMF20]. This has spawned interest in NNs that can represent *only* (anti)symmetric functions [ZKR⁺18, HLL⁺19]. A natural question is whether such NNs can represent *all* reasonable (anti)symmetric functions, which is the focus of this paper. We will answer this question for the (symmetric) EMLP [ZKR⁺18] defined in Section 7 and for the (AS) FermiNet [PSMF20] defined in Sections 4&5&7.

[1] In full generality, a function $f:\mathcal{X}\to\mathcal{Y}$ is covariant under group operations $g\in G$, if $f(R_g^X(x))=R_g^Y(f(x))$, where $R_g^X:\mathcal{X}\to\mathcal{X}$ and $R_g^Y:\mathcal{Y}\to\mathcal{Y}$ are representations of group (element) $g\in G$.

[2] The symmetric group $G=S_n$ is the group of all permutations=bijections $\pi:\{1,...,n\}\to\{1,...,n\}$.

[3] R_g^Y=Identity. Permutation-invariant functions are also called 'totally symmetric functions' or simply 'symmetric function'.

[4] General $\mathcal{Y}$ and $\mathcal{X}$, often $\mathcal{Y}=\mathcal{X}$ and $R_g^Y=R_g^X$, also called *covariant*.

[5] $R_g^Y=\pm1$ for even/odd permutations.

Desirable properties. Approximation architectures need to satisfy a number of criteria to be practically useful:

(a) they can approximate a large class of functions,
e.g. all continuous (anti)symmetric functions,
(b) *only* the (anti)symmetric functions can be represented,
(c) a fast algorithm exists for computing the approximation,
(d) the representation itself is continuous or differentiable,
(e) the architecture is suitable for learning the function from data
(which we don't discuss).

Content. Section 2 reviews existing approximation results for (anti)symmetric functions. Section 3 discusses various "naive" representations (linear, sampling, sorting) and their (dis)advantages, before introducing the "standard" solution that satisfies (a)-(e) based on algebraic composition of basis functions, symmetric polynomials, and polarized bases. For simplicity the section considers only totally symmetric functions of their n real-valued inputs (the $d=1$ case), i.e. particles in one dimension. Section 4 proves the representation power of a single GSD for totally anti-symmetric (AS) functions (also $d=1$). Technically we reduce the GSD to a Vandermonde determinant, and determine the loss of differentiability due to the Vandermonde determinant. From Sections 5 on we consider the general case of functions with $n\cdot d$ inputs that are (anti)symmetric when permuting their n d-dimensional input vectors. The case $d=3$ is particularly relevant for particles and point clouds in 3D space. The difficulties encountered for $d=1$ transfer to $d>1$, while the positive results don't, or only with considerable extra effort. Section 6 reviews classical NN approximation theory as a preparation for Equivariant MLPs proven universal in Section 7, which are then used in Section 8 to prove universality of Symmetric MLPs and of the AS FermiNet. Section 9 concludes. A list of notation can be found in Appendix A.

What's new. Our main novel contributions are establishing the universality of the anti-symmetric FermiNet with a single GSD (Theorems 3&5&8&15) for $d=1$ and $d>1$ (the results are non-trivial and unexpected), and the universality of (2-hidden-layer) symmetric MLPs (Theorem 14) with a complete and explicit and self-contained equivariant universality construction based on (smooth) polynomials. We took care to avoid relying on results with inherently asymptotic or tabulation or discontinuous character, to enable (in future work) good approximation rates for specific function classes, such as smooth functions or those with 'nice' Fourier transform [Bar93, Mak96],

2 Related Work

NN approximation theory [Pin99, LSYZ20]. The study of universal approximation properties of NN has a long history, see e.g. [Pin99] for a pre-millennium survey, and e.g. [LSYZ20] for recent results and references. For (anti)symmetric NN such investigation has only recently begun [ZKR$^+$18, WFE$^+$19, HLL$^+$19, SI19].

Zaher&al.(2018) [ZKR$^+$18]. Functions on sets are necessarily invariant under permutation, since the order of set elements is irrelevant. For countable domain, [ZKR$^+$18] derive a general representation based on encoding domain elements as bits into the binary expansion of real numbers. They conjecture that the construction can be generalized to uncountable domains such as $\mathbb{R}^d$, but it would have to involve pathological everywhere discontinuous functions [WFE+19]. Functions on sets of fixed size n are equivalent to symmetric functions in n variables. [ZKR+18] prove a symmetric version of Kolmogorov-Arnold's superposition theorem [Kol57] (for $d=1$) based on elementary symmetric polynomials und using Newton's identities, also known as Girard-Newton or Newton-Girard formulae, which we will generalize to $d>1$. Another proof is provided based on homeomorphisms between vectors and ordered vectors, also with no obvious generalization to $d>1$. They do not consider AS functions.

Han&al.(2019) [HLL+19]. For symmetric functions and any $d\geq 1$, [HLL+19] provide two proofs of the symmetric superposition theorem of [ZKR+18]: Every symmetric function can be approximated by symmetric polynomials, symmetrized monomials can be represented as a permanents, and Ryser's formula brings the representation into the desired polarized superposition form. The down-side is that computing permanents is NP complete, and exponentially many symmetrized monomials are needed to approximate f. The second proof discretizes the input space into a $n\cdot d$-dimensional lattice and uses indicator functions for each grid cell. They then symmetrize the indicator functions, and approximate f by these piecewise constant symmetric indicator functions instead of polynomials, also using Ryser formula for the final representation. Super-exponentially many indicator functions are needed, but explicit error bounds are provided. The construction is discontinuous but they remark on how to make it continuous. Approximating AS f for $d\geq 1$ is based on a similar lattice construction, but by summing super-exponentially many Vandermonde determinants, leading to a similar bound. We show that a single Vandermonde/Slater determinant suffices but without bound. Additionally for $d=1$ we determine the loss in smoothness this construction suffers from.

Sannei&al.(2019) [SI19]. [SI19] prove tighter but still exponential bounds if f is Lipschitz w.r.t. ℓ^∞ based on sorting which inevitably introduces irreparable discontinuities for $d>1$.

Pfau&al.(2019) [PSMF20]. The FermiNet [PSMF20] is also based on EMLPs [ZKR+18] but anti-symmetrizes not with Vandermonde determinants but with GSDs. It has shown remarkable practical performance for modelling the ground state of a variety of atoms and small molecules. To achieve good performance, a linear combination of GSDs has been used. We show that in principle a single GSD suffices, a sort of generalized Hartree-Fock approximation. This is contrast to the increasing number of conventional Slater determinants required for increasing accuracy. Our result implies (with some caveats) that the improved practical performance of multiple GSDs is due to a limited (approximation and/or learning) capacity of the EMLP, rather than a fundamental limit of the GSD.

3 One-Dimensional Symmetry

This section reviews various approaches to representing symmetric functions, and is the broadest review we are aware of. To ease discussion and notation, we consider $d=1$ in this section. Most considerations generalize easily to $d>1$, some require significant effort, and others break. We discuss various "naive" representations (linear, sampling, sorting) and their (dis)advantages, before introducing the "standard" solution that can satisfy (a)-(e). All representations consist of a finite set of fixed (inner) basis functions, which are linearly, algebraically, functionally, or otherwise combined. We then provide various examples, including composition by inversion and symmetric polynomials, which can be used to prove the "standard" representation theorem for $d=1$.

Motivation. Consider $n\in\mathbb{N}$ one-dimensional particles with coordinates $x_i\in\mathbb{R}$ for particle $i=1,...,n$. In quantum mechanics the probability amplitude of the ground state can be described by a real-valued joint wave function $\chi(x_1,...,x_n)$. Bosons ϕ have a totally symmetric wave function: $\phi(x_1,...,x_n)=\phi(x_{\pi(1)},...,x_{\pi(n)})$ for all permutations $\pi\in S_n\subset\{1:n\}\to\{1:n\}$. Fermions ψ have totally Anti-Symmetric (AS) wave functions: $\psi(x_1,...,x_n)=\sigma(\pi)\psi(x_{\pi(1)},...,x_{\pi(n)})$, where $\sigma(\pi)=\pm1$ is the parity or sign of permutation π. Wave functions are continuous and almost everywhere differentiable, and often posses higher derivatives or are even analytic. Nothing in this work hinges on any special properties wave functions may possess or interpreting them as such, and the precise conditions required for our results to hold are stated in the theorems.

We are interested in representing or approximating all and only such (anti)symmetric functions by neural networks. Abbreviate $\mathbf{x}\equiv(x_1,...,x_n)$ and let $S_\pi(\mathbf{x}):=(x_{\pi(1)},...,x_{\pi(n)})$ be the permuted coordinates. There is an easy way to (anti)symmetrize any function,

$$\phi(\mathbf{x}) ~=~ \frac{1}{n!}\sum_{\pi\in S_n}\chi(S_\pi(\mathbf{x})), \qquad \psi(\mathbf{x}) ~=~ \frac{1}{n!}\sum_{\pi\in S_n}\sigma(\pi)\chi(S_\pi(\mathbf{x})) \tag{1}$$

and any (anti)symmetric function can be represented in this form (proof: use $\chi:=\phi$ or $\chi:=\psi$). If we train a NN $\chi:\mathbb{R}^n\to\mathbb{R}$ to approximate some function $f:\mathbb{R}^n\to\mathbb{R}$ to accuracy $\varepsilon>0$, then ϕ (ψ) are (anti)symmetric approximations of f to accuracy $\varepsilon>0$ too, provided f itself is (anti)symmetric. Instead of averaging, the minimum or maximum or median or many other compositions would also work, but the average has the advantage that smooth χ lead to smooth ϕ and ψ, and more general, preserves many desirable properties such as (Lipschitz/absolute/...) continuity, (k-times) differentiability, analyticity, etc. It possibly has all important desirable properties, but one:

Time complexity. The problem with this approach is that it has $n!$ terms, and evaluating χ super-exponentially often is intractable even for moderate n, especially if χ is a NN. There can also be no clever trick to linearly (anti)symmetrize arbitrary functions fast, intuitively since the sum pools $n!$ independent regions of χ. Formally, consider the NP hard Travelling Salesman Problem (TSP): Let $\chi(\mathbf{x})=1$ if $x_i=\pi(i)\forall i$ for some π *and* there is a path of length $\leq L$ connecting cities in order $\pi(1)\to\pi(2)\to...\to\pi(n)\to\pi(1)$, and $\chi(\mathbf{x})=0$ for all other $\mathbf{x}$. Then $\phi(1,2,...,n)>0$ iff there exists a path of length $\leq L$ connecting the cities in *some* order. Hence ϕ solves the TSP, so $\phi\notin$P unless P=NP. For anti-symmetry, replace $\chi(\mathbf{x})=1$ by $\chi(\mathbf{x})=\sigma(\pi)$ in the above argument. The same argument also works when the averaging in (1) is replaced by min/max/median/etc.

Sampling. We could sample $O(1/\varepsilon^2)$ permutations to approximate ϕ and potentially ψ to accuracy ε, but even 10'000 samples for 1% accuracy is expensive, and cancellations of positive and negative terms may require orders of magnitude more samples. Furthermore, exactly (anti)symmetric wave functions are needed in quantum physics applications. Even if sampling were competitive for evaluation, there may be a super-exponential representation (learning) problem:

Learning. The domain of an (anti)symmetric function consist of $n!$ identical regions. The function is the same (apart from sign) on all these regions. A general NN must represent the function separately on all these regions, hence potentially requires $n!$ more training samples to learn from than an intrinsically (anti)symmetric NN, unless the NN architecture and learning algorithm are powerful enough to discover the symmetry by themselves and merge all $n!$ regions onto the same internal representation. Maybe a NN trained to sort [Wan95] exhibits this property. We are not aware of any general investigation of this idea.

Function composition and bases. Before delving into proving universality of the EMLP and the FermiNet, it is instructive to first review the general concepts of function composition and basis functions, since a NN essentially is a composition of basis functions. We want to represent/decompose functions as $f(\mathbf{x})=g(\boldsymbol{\beta}(\mathbf{x}))$. In this work we are interested in symmetric $\boldsymbol{\beta}$, where ultimately $\boldsymbol{\beta}$ will be represented by the first (couple of) layer(s) of an EMLP, and g by the second (couple of) layer(s). Of particular interest is

$$\boldsymbol{\beta}(\mathbf{x})=\sum_{i=1}^{n}\boldsymbol{\eta}(x_i) \tag{2}$$

for then $\boldsymbol{\beta}$ and hence f are obviously symmetric (permutation invariant) in $\mathbf{x}$. Anti-symmetry is more difficult and will be dealt with later. Formally let $f\in\mathcal{F}\subseteq\mathbb{R}^n\to\mathbb{R}$ be a function (class) we wish to represent or approximate. Let $\beta_b:\mathbb{R}^n\to\mathbb{R}$ be basis functions for $b=1,...,m\in\mathbb{N}\cup\{\infty\}$, and $\boldsymbol{\beta}\equiv(\beta_1,...,\beta_m):\mathbb{R}^n\to\mathbb{R}^m$ be what we call basis vector (function), and $\eta_b:\mathbb{R}\to\mathbb{R}$ a basis template, sometimes called inner function [Act18] or polarized bass function. Let $g\in\mathcal{G}\subseteq\mathbb{R}^m\to\mathbb{R}$ be a composition function (class), sometimes called 'outer function' [Act18], which creates new functions from the basis functions. Let $\mathcal{G}\circ\boldsymbol{\beta}=\{g(\boldsymbol{\beta}(\cdot)):g\in\mathcal{G}\}$ be the class of representable functions, and $\overline{\mathcal{G}\circ\boldsymbol{\beta}}$ its topological closure, i.e. the class of all approximable functions.[6] $\boldsymbol{\beta}$ is called a $\mathcal{G}$-basis for $\mathcal{F}$ if $\mathcal{F}=\mathcal{G}\circ\boldsymbol{\beta}$ or $\mathcal{F}=\overline{\mathcal{G}\circ\boldsymbol{\beta}}$, depending on context. Interesting classes of compositions are linear $\mathcal{G}_{lin}:=\{g:g(\mathbf{x})=a_0+\sum_{i=1}^m x_i;\ a_0,a_i\in\mathbb{R}\}$, algebraic $\mathcal{G}_{alg}:=\{\text{multivariate polynomials}\}$, functional $\mathcal{G}_{func}:=\mathbb{R}^m\to\mathbb{R}$, and $\mathcal{C}^k$-functional $\mathcal{G}^k_{func}:=\mathcal{C}^k$ for k-times continuously differentiable functions. **Examples.** For $n=1$, $\boldsymbol{\beta}(x)=\beta_1(x)=x$ is an algebraic basis of all polynomials x, and $\overline{\mathcal{G}_{alg}\circ\boldsymbol{\beta}}$ even includes all continuous functions $\mathcal{C}^0$, since every continuous function can be approximated arbitrarily well by polynomials. For $n=1$, $\beta_b(x)=x^b$ forms a linear basis for all polynomials of degree $m<\infty$ and includes all $\mathcal{C}^0$ functions via closure for $m=\infty$. $\beta_b(x)=x^{2b-1}$ for $m=\infty$ is a linear basis for all continuous axis-AS functions. For $n=3$, $\boldsymbol{\beta}(\mathbf{x})=\beta_1(\mathbf{x})=x^2+y^2+z^2$ is a functional basis for all rotationally-invariant functions. For $n=2$, the two elementary symmetric polynomials $\boldsymbol{\beta}(\mathbf{x})=(x_1+x_2,x_1x_2)$ constitute an algebraic basis for all symmetric polynomials, which already requires a bit of work to prove. Since $2x_1x_2=(x_1+x_2)^2-(x_1^2+x_2^2)$, also $\boldsymbol{\beta}(\mathbf{x})=(x_1+x_2,x_1^2+x_2^2)$ is an algebraic basis for all symmetric polynomials, and its closure includes all symmetric continuous functions. This last basis is of the desired sum form with $\eta_1(x)=x$ and $\eta_2(x)=x^2$, and has generalizations to $n>2$ and $d>1$ discussed later. The examples above illustrate that larger composition classes $\mathcal{G}$ allow (drastically) smaller bases (m) to represent the same functions $\mathcal{F}$. On the other hand, as we will see, algebraic bases can be harder to construct than linear bases.

Composition by inversion. Any injective $\boldsymbol{\beta}$ is a functional basis for all functions: For any f, $g(\boldsymbol{w}):=f(\boldsymbol{\beta}^{-1}(\boldsymbol{w}))$ with $\boldsymbol{w}\in\text{Image}(\boldsymbol{\beta})$ represents f as $f(\mathbf{x})=g(\boldsymbol{\beta}(\mathbf{x}))$. If $\boldsymbol{\beta}:\mathbb{R}\to$

[6] Functions may be defined on sub-spaces of $\mathbb{R}^k$, function composition may not exists, and convergence can be w.r.t. different topologies. We will ignore these technicalities unless important for our results, but the reader may assume compact-open topology, which induces uniform convergence on compacta.

Image($\boldsymbol{\beta}$) is a homeomorphism (diffeomorphism), then it is a continuous (differentiable) function basis for all continuous (differentiable) functions, and similarly for other general functions classes.

Generally invariant linear bases. Consider now functions $\mathcal{F}_\mathcal{S}$ invariant under some symmetry $\mathcal{S}\subseteq\mathbb{R}^n\to\mathbb{R}^n$, where $\mathcal{S}$ must be closed under composition, i.e. $\mathcal{F}_\mathcal{S}=\{f\in\mathcal{F}: f(\mathbf{x})=f(S(\mathbf{x}))\;\forall S\in\mathcal{S},\mathbf{x}\in\mathbb{R}^n\}$ for some $\mathcal{F}$. If $\boldsymbol{\beta}$ is a linear basis for $\mathcal{F}$, then for finite (compact) $\mathcal{S}$, $\boldsymbol{\beta}_\mathcal{S}(\mathbf{x}):=\sum_{S\in\mathcal{S}}\boldsymbol{\beta}(S(\mathbf{x}))$ ($\boldsymbol{\beta}_\mathcal{S}(\mathbf{x}):=\int_\mathcal{S}\boldsymbol{\beta}(S(\mathbf{x}))dS$) is a linear basis for $\mathcal{F}_\mathcal{S}$, not necessarily minimal. Above we mentioned the class of rotations $\mathcal{S}=O(3)$ and rotation-invariant functions/bases. Symmetrized monomials are discussed below.

Symmetric functions by sorting. Our prime interest is the symmetry class of permutations $\mathcal{S}_n:=\{S_\pi:\pi\in S_n\}$, where $S_\pi(x_1,...,x_n)=(x_{\pi(1)},...,x_{\pi(n)})$. An easy functional basis for symmetric functions is $\beta_b(\mathbf{x}):=\mathbf{x}_{[i]}$, where $\mathbf{x}_{[i]}$ is the i-th smallest value among $x_1,...,x_n$ (also called order statistics), i.e.

$$\min\{x_1,...,x_n\}=\mathbf{x}_{[1]}\leq\mathbf{x}_{[2]}\leq...\leq\mathbf{x}_{[n-1]}\leq\mathbf{x}_{[n]}=\max\{x_1,...,x_n\}$$

Obviously $\boldsymbol{\beta}(\mathbf{x})$ is symmetric (it just sorts its arguments), and any symmetric function ϕ can be represented as

$$\phi(\mathbf{x})\;=\;g(\boldsymbol{\beta}(\mathbf{x}))\;=\;\phi(\mathbf{x}_{[1]},...,\mathbf{x}_{[n]}),\quad\text{e.g. by choosing}\quad g(\mathbf{x})=\phi(\mathbf{x})$$

Note that $\boldsymbol{\beta}$ is continuous, hence continuous ϕ have a continuous representation, but $\boldsymbol{\beta}$ is not differentiable whenever $x_i=x_j$ for some $i\neq j$, so smooth ϕ have only non-smooth sorting representations. Still this is a popular representation for 1-dimensional quantum particles. This construction still works in higher dimensions, but leads to discontinuous functions, as we will see.

Linear basis for symmetric polynomials. The infinite class of monomials $\beta_{\boldsymbol{b}}(\mathbf{x})=x_1^{b_1}\cdot...\cdot x_n^{b_n}$ for $\boldsymbol{b}\in\mathbb{N}_0^n$ are a linear basis of all polynomials in n variables. Hence the symmetrized monomials $\beta_{\boldsymbol{b}}^{S_n}(\mathbf{x})=\sum_{\pi\in S_n}x_{\pi(1)}^{b_1}\cdot...\cdot x_{\pi(n)}^{b_n}$ form a linear basis for all symmetric polynomials, and by closure includes all continuous symmetric functions. This does *not* work for algebraic bases: While $\boldsymbol{\beta}(\mathbf{x})=(x_1,...,x_n)$ is an algebraic basis of all polynomials and by closure of $\mathcal{C}^0$, $\beta_b^{S_n}(\mathbf{x})=(n-1)!(x_1+...+x_n)$ algebraically generates only (ridge) functions of the form $g(x_1+...+x_n)$ which are constant in all directions orthogonal to (1,...1), so this is not a viable path for constructing algebraic bases.

Algebraic basis for symmetric polynomials. It is well-known that the elementary symmetric polynomials $e_b(\mathbf{x})$ generated by

$$\prod_{i=1}^n(1+\lambda x_i)\;=:\;1+\lambda e_1(\mathbf{x})+\lambda^2e_2(\mathbf{x})+...+\lambda^ne_n(\mathbf{x})\tag{3}$$

are an algebraic basis of all symmetric polynomials. Explicit expressions are $e_1(\mathbf{x})=\sum_i x_i$, and $e_2(\mathbf{x})=\sum_{i<j}x_ix_j$, ..., and $e_n(\mathbf{x})=x_1...x_n$, and in general $e_b(\mathbf{x})=\sum_{i_1<...<i_b}x_{i_1}...x_{i_b}$. For given $\mathbf{x}$, the polynomial in λ on the l.h.s. of (3) can be expanded to the r.h.s. in quadratic time or by FFT even in time $O(n\log n)$, so the $\boldsymbol{e}(\mathbf{x})$ can be computed in time $O(n\log n)$, but is not of the desired form (2). Luckily Newton already solved this problem for us. Newton's identities express the elementary symmetric polynomials $e_1(\mathbf{x}),...,e_n(\mathbf{x})$ as polynomials in $p_b(\mathbf{x}):=\sum_{i=1}^n x_i^b$, $b=1,...,n$, hence also $\boldsymbol{\beta}(\mathbf{x}):=(p_1(\mathbf{x}),...,p_n(\mathbf{x}))$ is an algebraic basis for all symmetric polynomials, hence by closure for all continuous symmetric functions, and is of desired form (2):

Theorem 2 (Symmetric polarized superposition [ZKR+18, WFE+19, Thm.7]) *Every continuous symmetric function $\phi:\mathbb{R}^n\to\mathbb{R}$ can be represented as $\phi(\mathbf{x})=g(\sum_i\boldsymbol{\eta}(x_i))$ with $\boldsymbol{\eta}(x)=(x,x^2,...,x^n)$ and continuous $g:\mathbb{R}^n\to\mathbb{R}$.*

[ZKR+18] provide two proofs, one based on 'composition by inversion', the other using symmetric polynomials and Newton's identities. The non-trivial generalization to $d>1$ is provided in Section 5.

Theorem 2 is a symmetric version of the infamous Kolmogorov-Arnold superposition theorem [Kol57], which solved Hilbert's 13th problem. Its deep and obscure[⑦] constructions continue to fill whole PhD theses [Liu15, Act18]. It is quite remarkable that the symmetric version above is very natural and comparably easy to prove.

For given $\mathbf{x}$, the basis $\boldsymbol{\beta}(\mathbf{x})$ can be computed in time $O(n^2)$, so is actually slower to compute than $\boldsymbol{e}(\mathbf{x})$. The elementary symmetric polynomials also have other advantages (integral coefficients for integral polynomials, works for fields other than $\mathbb{R}$, is numerically more stable, mimics 1,2,3,... particle interactions), so symmetric NN based on e_b rather than p_b may be worth pursuing. Note that we need at least $m\geq n$ functional bases for a *continuous* representation, so Theorem 2 is optimal in this sense [WFE+19].

Table 1 summarizes the bases and properties discussed in this section and beyond.

4 One-Dimensional AntiSymmetry

We now consider the anti-symmetric (AS) case for $d=1$. We provide representations of AS functions in terms of generalized Slater determinants (GSD) of *partially* symmetric functions. In later sections we will discuss how these partially symmetric functions arise from equivariant functions and how to represent equivariant functions by EMLP. The reason for deferral is that EMLP are inherently tied to $d>1$. Technically we show that the GSD can be reduced to a Vandermonde determinant, and exhibit a potential loss of differentiability due to the Vandermonde determinant.

Analytic Anti-Symmetry. Let $\varphi_i:\mathbb{R}\to\mathbb{R}$ be single-particle wave functions. Consider the matrix

$$\Phi(\mathbf{x}) \;=\; \begin{pmatrix} \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{pmatrix}$$

where $\mathbf{x}\equiv(x_1,...,x_n)$. The (Slater) determinant $\det\Phi(\mathbf{x})$ is anti-symmetric, but can represent only a small class of AS functions, essentially the AS analogue of product (wave) functions (pure states, Hartree-Fock approximation). Every continuous AS function can be approximated/represented by a finite/infinite linear combination of such determinants:

$$\psi(x_1,...,x_n) \;=\; \sum_{k=1}^{\infty}\det\Phi^{(k)}(\mathbf{x}),\;\; \text{where}\;\; \Phi_{ij}^{(k)}(\mathbf{x}):=\varphi_i^{(k)}(x_j)$$

[⑦] involving continuous η with derivative 0 almost everywhere, and not differentiable on a dense set of points.

Table 1: **Bases and properties for $\boldsymbol{d\!=\!1}$. Last column comments on $\boldsymbol{d\!>\!1}$.** Many different representations for symmetric functions have been suggested. The representations are very heterogenous, so this table is our best but limited attempt to unify and press them into one table. The table is for $d\!=\!1$, but when (some aspects of) the method generalizes to $d\!>\!1$ we provide #Bases for $d\!\geq\!1$ and a comment or reference in the last column. Most representations can be viewed as instantiations of $\phi(\mathbf{x}) = g(\boldsymbol{\beta}(\mathbf{x}))$ with outer function g composing inner base functions $\boldsymbol{\beta}$ possibly polarized as $\boldsymbol{\beta}\!=\!\sum_{i=1}^n \boldsymbol{\eta}$. See main text and references for details, and glossary below. The table is roughly in order as described in Section 3. For the last three rows, see references. The meaning of the columns is described in the main text.

Base	$O()$ comp.	#Bases	Basis			Prop.	$\boldsymbol{\beta}\!=$?	Refs	
$\boldsymbol{\beta}\vert\boldsymbol{\eta}\vert$else	time $\boldsymbol{\beta}(\mathbf{x})$	m	type$\mathcal{G}$	$\mathcal{G}\circ\boldsymbol{\beta}$	$\overline{\mathcal{G}\circ\boldsymbol{\beta}}$	of $\boldsymbol{\beta}$	$\sum\boldsymbol{\eta}$?		$d\!>\!1$
Σ_π	$n!$	$\aleph_2$	Id	SymFct	–	–	–	Sec.3	same
Inversion	depends	n	$\mathcal{C}^0$	$\mathcal{C}^0$	=	$\mathcal{C}^0$	✓	[ZKR+18]	un-known
$\beta_b(\mathbf{x})\!=\!\mathbf{x}_{[b]}$	$n\log n$	n	$\mathcal{C}^0$	Sym$\mathcal{C}^0$	Sym$\mathcal{C}^0$	$\mathcal{C}^0$	no	Sec.3	$\boldsymbol{\beta}\!\notin\!\mathcal{C}^0$
SymMonom. $\leq$ deg D	D^n	$\binom{D+n}{n}$	Lin	SymPoly^D	=	$\mathcal{C}^\infty$	no		same
AllSym Monomials	$\aleph_0$	$\aleph_0$	Lin	SymPoly	Sym$\mathcal{C}^0$	$\mathcal{C}^\infty$	no		same
$e_b(\mathbf{x})$	$n\log n$	$\binom{n+d}{d}\!-\!1$	Alg	SymPoly	Sym$\mathcal{C}^0$Fct	$\mathcal{C}^\infty$	no	Sec.3	Sec.5
$\eta_b(x)\!=\!x^b$	n^2	$\binom{n+d}{d}\!-\!1$	Alg	SymPoly	Sym$\mathcal{C}^0$Fct	$\mathcal{C}^\infty$	✓	[ZKR+18]	Sec.5
η_b=Cantor	$n\log n$	1	All	AllSymFct	–	total.discont.	✓	[ZKR+18]	same
ε-Grid	$(^1/_\varepsilon)^{dn}$	$(^1/_\varepsilon)^{dn}$	Lin	SymPCG	=	∞-indicator	✓	[LSYZ20]	same
Smoothed ε-Grid	$(^1/_\varepsilon)^{dn}$	$(^1/_\varepsilon)^{dn}$	Lin	Smoothed SymPC	$\mathcal{C}^0$	∞-indicator	✓	[LSYZ20]	same

Symbol	*Explanation*
$d{,}n\!\in\!\mathbb{N}$	dimensionality,number of particles
$\mathbf{x}\!\in\!\mathbb{R}^n$	$\mathbf{x}\!=\!(x_1{,}...{,}x_n)$, function argument, NN input, n 1d particles ($d\!=\!1$)
$b\!\in\!\{1\!:\!m\}$	index of basis function
$\boldsymbol{\beta}\!:\!\mathbb{R}^n\!\to\!\mathbb{R}^m$	m symmetric basis functions
$\boldsymbol{\eta}\!:\!\mathbb{R}\!\to\!\mathbb{R}^m$	m polarized basis functions
$g\!:\!\mathbb{R}^m\!\to\!\mathbb{R}$	composition or outer function, used as $g(\boldsymbol{\beta}(\mathbf{x}))$
Sym(PCG)	symmetric (piecewise constant grid)
Poly$^{(D)}$	Multivariate polynomial (of degree at most D)
$\mathcal{G}_{type}\!\ni\!g$	Composition class (Id-entity, Lin-ear, Alg-ebraic, $\mathcal{C}^0$=cont., All fcts)

An alternative is to generalize the Slater determinant itself [PSMF20] by allowing the functions $\varphi_i(x_j)$ to depend on all variables

$$\Phi(\mathbf{x}) \;=\; \begin{pmatrix} \varphi_1(x_1|x_{\neq 1}) & \cdots & \varphi_n(x_1|x_{\neq 1}) \\ \vdots & \ddots & \vdots \\ \varphi_1(x_n|x_{\neq n}) & \cdots & \varphi_n(x_n|x_{\neq n}) \end{pmatrix}$$

where $x_{\neq i}\equiv(x_1,...,x_{i-1},x_{i+1},...,x_n)$. If $\varphi_i(x_j|x_{\neq j})$ is symmetric in $x_{\neq j}$, which we henceforth assume[8], then exchanging $x_i\leftrightarrow x_j$ is (still) equivalent to exchanging rows i and j in $\Phi(\mathbf{x})$, hence $\det\Phi$ is still AS. The question arises how many GSD are needed to be able to represent *every* AS function ψ. The answer turns out to be 'just one', but with non-obvious smoothness relations: Any AS ψ can be represented by some Φ, any analytic ψ can be represented by an analytic Φ. The case of continuous(ly differentiable) ψ is more complicated.

Theorem 3 (Representation of all (analytic) AS $\boldsymbol{\psi}$) *For every (analytic) AS function $\psi(\mathbf{x})$ there exist (analytic) $\varphi_i(x_j|x_{\neq j})$ symmetric in $x_{\neq j}$ such that $\psi(\mathbf{x})=\det\Phi(\mathbf{x})$.*

Proof. Let $\varphi_1(x_j|x_{\neq j}):=\chi(x_{1:n})$ be totally symmetric in all $(x_1,...,x_n)$ to be determined later. Let $\varphi_i(x_j|x_{\neq j}):=x_j^{i-1}$ for $1<i\leq d$. Then

$$\det\Phi(\mathbf{x})=\begin{vmatrix} \chi(x_{1:n}) & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \chi(x_{1:n}) & x_n & \cdots & x_n^{n-1} \end{vmatrix}=\chi(x_{1:n})\cdot\begin{vmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^{n-1} \end{vmatrix}=\chi(x_{1:n})\prod_{1\leq j<i\leq n}(x_i-x_j)$$

where the (second) last expression is (the expression for) the Vandermonde determinant. Since ψ is AS, $\psi(\mathbf{x})=0$ if $x_i=x_j$ for any $i\neq j$, hence ψ has factors x_i-x_j for all $i\neq j$. Therefore $\chi(x_{1:n}):=\psi(x_{1:n})/\prod_{j<i}(x_i-x_j)$ is totally symmetric, since $\prod_{j<i}(x_i-x_j)$ is AS, and obviously $\det\Phi(\mathbf{x})=\psi(\mathbf{x})$ for this choice. For general AS ψ and $x_i=x_j$ we can define $\chi(\mathbf{x})$ arbitrarily, as long as it is symmetric, e.g. $\chi(\mathbf{x})=0$ will do. For analytic ψ, the next lemma shows that χ has an analytic extension. ■

Lemma 4 (Symmetric $\boldsymbol{\psi/\Delta}$ is analytic if AS $\boldsymbol{\psi}$ is analytic) *Let $\psi:\mathbb{R}^n\to\mathbb{R}$ be AS and analytic. Then $\chi(\mathbf{x}):=\psi(\mathbf{x})/\Delta(\mathbf{x})$, where $\Delta(\mathbf{x}):=\prod_{1\leq j<i\leq n}(x_i-x_j)$, is totally symmetric and analytic on $\mathbb{R}^n\setminus\{\mathbf{x}:\Delta(\mathbf{x})=0\}$ and has a symmetric analytic continuation to all $\mathbb{R}^n$.*

Proof. Since ψ is analytic, it has a multivariate Taylor series expansion. For $\boldsymbol{k}=(k_1,...,k_n)\in\mathbb{N}_0^n$ and $\mathbf{x}^{\boldsymbol{k}}:=x_1^{k_1}\cdots x_n^{k_n}$, we have $\psi(\mathbf{x})=\sum_{\boldsymbol{k}}a_{\boldsymbol{k}}\mathbf{x}^{\boldsymbol{k}}$ for some $a_{\boldsymbol{k}}\in\mathbb{R}$. If we anti-symmetrize both sides by $AS[\mathbf{x}^{\boldsymbol{k}}]:=\sum_{\pi\in S_n}\sigma(\pi)x_{\pi(1)}...x_{\pi(n)}$, we get $\psi(\mathbf{x})=\sum_{\boldsymbol{k}}a_{\boldsymbol{k}}AS[\mathbf{x}^{\boldsymbol{k}}]$. Every AS polynomial $AS[\mathbf{x}^{\boldsymbol{k}}]$ can be represented as $AS[\mathbf{x}^{\boldsymbol{k}}]=\Delta(\mathbf{x})S_{\boldsymbol{k}}(\mathbf{x})$ for some symmetric polynomials $S_{\boldsymbol{k}}$. This follows by successively dividing out all factors (x_j-x_i) [Wey46, Sec.II.2]. Hence $\psi(\mathbf{x})=\Delta(\mathbf{x})\chi(\mathbf{x})$ with $\chi(\mathbf{x}):=\sum_{\boldsymbol{k}}a_{\boldsymbol{k}}S_{\boldsymbol{k}}(\mathbf{x})$, which is obviously symmetric and analytic. ■

Continuous/differentiable AS. We can weaken the analyticity condition as follows:

[8] The bar | is used to visually indicate this symmetry, otherwise there is no difference to using a comma.

Theorem 5 (Representation of continuous/differentiable ψ) *Let $\mathcal{C}^k(\mathbb{R}^n)$ be the k-times continuously differentiable functions ($k\in\mathbb{N}_0$).*[9] *For AS function $\psi\in\mathcal{C}^{k+n(n+1)/2}(\mathbb{R}^n)$ there exist $\varphi_i(x_j|x_{\neq j})\in\mathcal{C}^k(\mathbb{R}^n)$ symmetric in $x_{\neq j}$ such that $\psi(\mathbf{x})=\det\Phi(\mathbf{x})$.*

This is a much weaker result than for *linear* anti-symmetrization (1), where all $\psi\in\mathcal{C}^k$ could be represented by $\chi\in\mathcal{C}^k$, in particular continuous ψ had continuous representations. For instance, Theorem 5 (only) implies that $\frac{1}{2}n(n+1)$-times differentiable ψ have continuous representations, but leaves open whether less than $\frac{1}{2}n(n+1)$-times differentiable ψ *may* only have discontinuous representations. It turns out that this is not the case. In Section 5 we show that continuous ψ can be represented by continuous Φ, but whether $\psi\in\mathcal{C}^k$ has representations with $\Phi\in\mathcal{C}^k$ is open for $k>0$.

Proof. Consider functions $\psi_A:\mathbb{R}^n\to\mathbb{R}$ with $\psi_A(\mathbf{x})=0$ if $x_i=x_j$ for some $(i,j)\in A\subseteq\{(i,j):1\leq i<j\leq n\}=:P$. Note that $\psi_P:=\psi$ satisfies this condition, but the constructed ψ_A will *not* be AS for $A\neq P$. The proof recursively divides out factors x_j-x_i: For $A=A'\dot{\cup}\{(i,j)\}$ define

$$\psi_{A'}(\mathbf{x}) \;:=\begin{cases} \frac{\psi_A(\mathbf{x})}{x_j-x_i} & \text{if} \quad x_j\neq x_i \\ \frac{\partial\psi_A(\mathbf{x})}{\partial x_j}|_{x_j=x_i} & \text{if} \quad x_j=x_i \end{cases}$$

If $\psi_A\in\mathcal{C}^k$ then $\psi_{A'}\in\mathcal{C}^{k-1}$ by the Lemma 6 below. The recursive elimination is independent of the order in which we choose pairs from A. Though not needed, note also that the definition is symmetric in $i\leftrightarrow j$, since $\partial_j\psi|_{x_j=x_i}=-\partial_i\psi|_{x_i=x_j}$ due to $\psi(x,x)\equiv 0$ implying $0=d\psi(x,x)/dx=\partial_1\psi+\partial_2\psi$. For $x_j\neq x_i$ we obviously have $\psi_{A'}(\mathbf{x})=0$ if $x_\jmath=x_\imath$ for some $(\imath,\jmath)\in A'$. $\psi_{A'}(\mathbf{x})=0$ also holds for $x_j=x_i$ by a continuity argument or direct calculation. We hence can recursively divide out x_j-x_i (in any order)

$$\chi(\mathbf{x}) \;:=\; \psi_{\{\}}(\mathbf{x}) \;=...=\; \frac{\psi_A(\mathbf{x})}{\prod_{(i,j)\in A}(x_j-x_i)} \;=...=\; \frac{\psi_P(\mathbf{x})}{\prod_{(i,j)\in P}(x_j-x_i)} \;=\; \frac{\psi(\mathbf{x})}{\Delta(\mathbf{x})}$$

and $\chi\in\mathcal{C}^k$ if $\psi\in\mathcal{C}^{k+|P|}$. Since ψ and Δ are AS, χ is symmetric. This shows that $\varphi_1:=\chi\in\mathcal{C}^k$ and the other $\varphi_i=x_j^{i-1}$ are even analytic. ■

Unfortunately this construction does not generalize to $d>1$ dimensions, but a different construction in Section 5 will give a (somewhat) weaker result. The proof above used the following lemma:

Lemma 6 ($f\in\mathcal{C}^k$ implies $f/x\in\mathcal{C}^{k-1}$) *Let $f:\mathbb{R}\to\mathbb{R}$ be $k\geq 1$-times differentiable and $f(0)=0$, then $g(x):=f(x)/x$ for $x\neq 0$ and $g(0):=f'(0)$ is $k-1$-times continuously differentiable, i.e. $g\in\mathcal{C}^{k-1}(\mathbb{R})$.*

The lemma can be proven by equating the remainder of the Taylor series expansions of f up to term k with that of g. Only showing continuity of $g^{(k-1)}$ requires some work. Note that we neither require $f^{(k)}$ to be continuous, nor $f(-x)=-f(x)$ or so.

[9] For $\boldsymbol{k}=(k_1,...,k_n)$, $\mathcal{C}^{\boldsymbol{k}}$ means $\partial_{x_1^{k_1}}\cdots\partial_{x_n^{k_n}}$ exists and is continuous. $\mathcal{C}^{|\boldsymbol{k}|}:=\bigcap_{|\boldsymbol{k}|=k}\mathcal{C}^{\boldsymbol{k}}$, where $|\boldsymbol{k}|:=k_1+...+k_n$. We actually only need $\psi\in\bigcap_{|\boldsymbol{k}|=k}\mathcal{C}^{\boldsymbol{k}+(0,1,...,n-1)}\supsetneq\mathcal{C}^{k+n(n-1)/2}$.

5 d-dimensional (Anti)Symmetry

This section generalizes the theorems from Sections 3 and 4 to $d>1$: the symmetric polynomial algebraic basis and the generalized Slater determinant representation.

Motivation. We now consider $n\in\mathbb{N}$, d-dimensional particles with coordinates $\boldsymbol{x}_i\in\mathbb{R}^d$ for particles $i=1,...,n$. For $d=3$ we write $\boldsymbol{x}_i=(x_i,y_i,z_i)^\top\in\mathbb{R}^3$. As before, Bosons/Fermions have symmetric/AS wave functions $\chi(\boldsymbol{x}_1,...,\boldsymbol{x}_n)$. That is, χ does not change/changes sign under the exchange of two vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. It is *not* symmetric/AS under the exchange of individual coordinates e.g. $y_i\leftrightarrow y_j$. $\mathbf{X}\equiv(\boldsymbol{x}_1,...,\boldsymbol{x}_n)$ is a matrix with n columns and d rows. The (representation of the) symmetry group is $\mathcal{S}_n^d:=\{S_\pi^d:\pi\in S_n\}$ with $S_\pi^d(\boldsymbol{x}_1,...,\boldsymbol{x}_n):=(\boldsymbol{x}_{\pi(1)},...,\boldsymbol{x}_{\pi(n)})$, rather than $\mathcal{S}_{n\cdot d}$. Functions $f:\mathbb{R}^{d\cdot n}\to\mathbb{R}$ invariant under $\mathcal{S}_n^d$ are sometimes called multisymmetric or block-symmetric, if calling them symmetric could cause confusion.

Algebraic basis for multisymmetric polynomials. The elementary symmetric polynomials (3) have a generalization to $d>1$ [Wey46]. We only present them for $d=3$. The general case is obvious from them. They can be generated from

$$\prod_{i=1}^n(1+\lambda x_i+\mu y_i+\nu z_i) \;=:\; \sum_{0\leq p+q+r\leq n}\lambda^p\mu^q\nu^r e_{pqr}(\mathbf{X}) \tag{4}$$

Even for $d=3$ the expressions are rather cumbersome:

$$\begin{aligned} e_{pqr}(\mathbf{X}) &= \sum_{\substack{1\leq i_1<...<i_p\leq n\\ 1\leq j_1<...<j_q\leq n\\ 1\leq k_1<...<k_r\leq n}, \text{all}\neq} x_{i_1}...x_{i_p}y_{j_1}...y_{j_q}z_{k_1}...z_{k_r} \\ &= \frac{1}{p!q!r!}\sum_{\pi\in S_n}x_{\pi(1)}...x_{\pi(p)}y_{\pi(p+1)}...y_{\pi(p+q)}z_{\pi(p+q+1)}...z_{\pi(p+q+r)} \end{aligned}$$

One can show that $\{e_{pqr}:p+q+r\leq n\}$ is an algebraic basis of size $m=\binom{n+3}{3}-1$ for all multisymmetric polynomials [Wey46]. Note that constant e_{000} is not included/needed. For a given $\mathbf{X}$, their values can be computed in time $O(mn)$ by expanding (4) or in time $O(m\log n)$ by FFT, where $m=O(n^d)$. Newton's identities also generalize: $e_{pqr}(\mathbf{X})$ are polynomials in the polarized sums $p_{pqr}(\mathbf{X}):=\sum_{i=1}^n\eta_{pqr}(\boldsymbol{x}_i)$ with $\eta_{pqr}(\boldsymbol{x}):=x^py^qz^r$. The proofs are much more involved than for $d=1$. For the general d-case we have:

Theorem 7 (Multisymmetric polynomial algebraic basis [Wey46]) *Every continuous (block=multi)symmetric function $\phi:\mathbb{R}^{n\cdot d}\to\mathbb{R}$ can be represented as $\phi(\mathbf{X})=g(\sum_{i=1}^n\boldsymbol{\eta}(\boldsymbol{x}_i))$ with continuous $g:\mathbb{R}^m\to\mathbb{R}$ and $\boldsymbol{\eta}:\mathbb{R}^d\to\mathbb{R}^m$ defined as $\eta_{p_1...p_d}(\boldsymbol{x})=x^{p_1}y^{p_2}...z^{p_d}$ for $1\leq p_1+...+p_d\leq n$ ($p_i\in\{0,...,n\}$), hence $m=\binom{n+d}{d}-1$.*

The basis can be computed in time $O(m\cdot d)$ by first computing all powers $x^{p_1},...,z^{p_d}$ with $(n-1)\cdot d$ multiplications, then each of the m η can be computed with just $d-1$ multiplications. Since $m\leq(n+1)^d$, this implies $O(md)\subseteq O(d(n+1)^d)$. Note that there could be much smaller functional bases of size $m=dn$ as per "our" composition-by-inversion argument for continuous representations in Section 3, which readily generalizes to $d>1$, while the above minimal algebraic basis has larger size $m=O(n^d)$ for $n\gg d>1$. It is an open question whether a continuous functional basis of size $O(dn)$ exists, whether in polarized form (2) or not.

Anti-Symmetry. For $d=1$, *all* AS functions ψ have the *same* (core) zeros (called Fermion nodes), namely when $x_i = x_j$ for some $i \neq j$, which form a union of linear spaces dividing $\mathbb{R}^n$ into $n!$ isomorphic partitions on which ψ are identical apart from sign ± 1. This fact allowed representing every ψ as a product of a symmetric function ϕ and the universal anti-symmetric polynomial Δ, leading to representation Theorem 3. For $d>1$, the Fermion nodes $\{\mathbf{X} : \psi(\mathbf{X}) = 0\}$ form essentially arbitrary ψ-dependent unions of (non-linear) manifolds partitioning $\mathbb{R}^{dn}$ into an arbitrary even number of cells of essentially arbitrary topology [Mit07]. This fact prevents a similar simple factoring ($\psi = \phi \cdot \Delta$) and Vandermonde-like reduction in the proof of Theorem 3, and indeed prevents finite *algebraic* bases for AS polynomials. We can still show a similar representation result, albeit weaker and via a different construction:

As in Section 4, consider $\Phi_{ij}(\mathbf{X}) := \varphi_i(\boldsymbol{x}_j | \boldsymbol{x}_{\neq j})$, i.e.

$$\Phi(\mathbf{X}) \;=\; \begin{pmatrix} \varphi_1(\boldsymbol{x}_1|\boldsymbol{x}_{\neq 1}) & \cdots & \varphi_n(\boldsymbol{x}_1|\boldsymbol{x}_{\neq 1}) \\ \vdots & \ddots & \vdots \\ \varphi_1(\boldsymbol{x}_n|\boldsymbol{x}_{\neq n}) & \cdots & \varphi_n(\boldsymbol{x}_n|\boldsymbol{x}_{\neq n}) \end{pmatrix}$$

where $\varphi_i(\boldsymbol{x}_j|\boldsymbol{x}_{\neq j})$ is symmetric in $\boldsymbol{x}_{\neq j}$.

Theorem 8 (Representation of all AS ψ) *For every AS function $\psi(\mathbf{X})$ there exist $\varphi_i(\boldsymbol{x}_j|\boldsymbol{x}_{\neq j})$ symmetric in $\boldsymbol{x}_{\neq j}$ such that $\psi(\mathbf{X}) = \det\Phi(\mathbf{X})$.*

Proof. Define any total order $<$ on $\mathbb{R}^d$. For definiteness choose "lexicographical" order $\boldsymbol{x}_i < \boldsymbol{x}_j$ iff $x_i < x_j$ or ($x_i = x_j$ and $y_i < y_j$) or ($x_i = x_j$ and $y_i = y_j$ and $z_i < z_j$), etc. for $d>3$. Let $\bar\pi \in S_n$ be the permutation which sorts the particles $\boldsymbol{x}_i \in \mathbb{R}^d$ in increasing order, i.e. $\boldsymbol{x}_{\bar\pi(1)} \leq \boldsymbol{x}_{\bar\pi(2)} \leq \ldots \leq \boldsymbol{x}_{\bar\pi(n)}$. Note that this permutation depends on $\mathbf{X}$, but since $\mathbf{X}$ is held fixed for the whole proof, we don't need to worry about this. Now temporarily assume $\psi(\mathbf{X}) \geq 0$, and define

$$\varphi_i(\boldsymbol{x}_j|\boldsymbol{x}_{\neq j}) \;:=\; \begin{cases} \psi(\boldsymbol{x}_{\bar\pi(1)},\ldots,\boldsymbol{x}_{\bar\pi(n)})^{1/n} & \text{if} \quad j = \bar\pi(i) \\ 0 & \text{else} \end{cases} \tag{5}$$

which is symmetric in $\boldsymbol{x}_{\neq j}$. If the $\boldsymbol{x}_i$ are already sorted, i.e. $\bar\pi(i) = i \;\forall i$, then $\varphi_i(\boldsymbol{x}_j|\boldsymbol{x}_{\neq j}) = 0$ unless $j = i$. Hence $\Phi(\mathbf{X})$ is diagonal with

$$\det\Phi(\boldsymbol{x}) \;=\; \begin{vmatrix} \varphi_1(\boldsymbol{x}_1|\boldsymbol{x}_{\neq 1}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \varphi_n(\boldsymbol{x}_n|\boldsymbol{x}_{\neq n}) \end{vmatrix} \;=\; \prod_{i=1}^n \varphi_i(\boldsymbol{x}_i|\boldsymbol{x}_{\neq i}) \;=\; \psi(\mathbf{X})$$

For a general permutation $\bar\pi$, Φ is a permuted diagonal matrix with only row $\bar\pi(i)$ being non-zero in column i:

$$\det\Phi(\boldsymbol{x}) = \begin{vmatrix} \vdots & \cdots & \vdots \\ \vdots & \ddots & \varphi_n(\boldsymbol{x}_{\bar\pi(n)}|\boldsymbol{x}_{\neq\bar\pi(n)}) \\ \varphi_1(\boldsymbol{x}_{\bar\pi(1)}|\boldsymbol{x}_{\neq\bar\pi(1)}) & \ddots & \vdots \\ \vdots & \cdots & \vdots \end{vmatrix} \;=\; \sigma(\bar\pi)\prod_{i=1}^n \varphi_i(\boldsymbol{x}_{\bar\pi(i)}|\boldsymbol{x}_{\neq\bar\pi(i)})$$

$$=\; \sigma(\bar\pi)\psi(\boldsymbol{x}_{\bar\pi(1)},\ldots,\boldsymbol{x}_{\bar\pi(n)}) \;=\; \sigma(\bar\pi)\sigma(\bar\pi)\psi(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) \;=\; \psi(\mathbf{X})$$

where we exploited that ψ is AS. If n is odd, this construction also works for negative ψ. In general we can replace $\psi^{1/n}$ in (5) by $\text{sign}(\psi)|\psi|^{1/n}$ in φ_1 and by $|\psi|^{1/n}$ for the other φ_i, or even ψ for φ_1 and 1 for the other φ_i. ■

Note that for $d\!=\!1$, $(x_{\bar\pi(1)},...,x_{\bar\pi(n)})$ is continuous in $(x_1,...,x_n)$, and $\bar\pi$ is (only) discontinuous when $x_i\!=\!x_j$ for some $i\!\neq\! j$, but then $\psi\!=\!0$, hence φ_i in (5) is continuous for continuous ψ. Unfortunately this is no longer true for $d\!>\!1$. For instance for $\psi(\boldsymbol{x}_1,\boldsymbol{x}_2)\!=\!y_1\!-\!y_2$, $\varphi_1(\binom{x_1}{1},\binom{x_2}{0})=[\![x_1<x_2]\!]$ is a step function.

For $n\!=\!2$ and any d, any AS continuous/smooth/analytic/$\mathcal{C}^k$ function $\psi(\boldsymbol{x}_1,\boldsymbol{x}_2)$ has an easy continuous/smooth/analytic/$\mathcal{C}^k$ representation as a GSD. Choose $\varphi_1(\boldsymbol{x}_1|\boldsymbol{x}_2)\!:=\!\frac{1}{2}$ and $\varphi_2(\boldsymbol{x}_1|\boldsymbol{x}_2)\!:=\!\psi(\boldsymbol{x}_1,\boldsymbol{x}_2)$. Whether this generalizes to $n\!>\!2$ and $d\!>\!1$ is an open problem.

6 Universality of Neural Networks

This section is mainly definitions and a few known Neural Network (NN) approximation results, for setting the stage of the next section for Equivariant MLP. We present the two key theorems for polynomial-based proofs of universality of NN, namely that MLP can approximate any polynomial, which in turn can approximate any continuous function. The NN approximation theory literature is vast, so these two results and a few key references must do.

Multi-Layer Perceptron (MLP). The standard Multi-Layer Perceptron (MLP) aims at approximately representing functions $f:\mathbb{R}^n\to\mathbb{R}^m$ ($n,m\in\mathbb{N}$, most often $m=1$) as follows: Let $\sigma:\mathbb{R}\to\mathbb{R}$ be some non-polynomial[10] continuous activation function. If not mentioned explicitly otherwise, we will assume that $\sigma(z)=\max\{0,z\}$ or $\sigma(z)=\tanh(z)$, and results are valid for both. Most choices will also do in practice and in theory, though there can be some differences. We use function vectorization notation $\boldsymbol{\sigma}(\boldsymbol{x})_i\!:=\!\sigma(x_i)$.

The NN input is $\boldsymbol{x}\in\mathbb{R}^n$, its output is $\boldsymbol{y}\in\mathbb{R}^m$. We use upper indices (ℓ) and $(\ell\!+\!1)$ without braces for indexing layers. They are never exponents. Layer $\ell\!+\!1$ is computed from layer $\ell\in\{0,...,L\!-\!1\}$ by

$$\boldsymbol{x}^{\ell+1} \;:=\; \boldsymbol{\tau}^\ell(\boldsymbol{x}^\ell) \;\equiv\; \boldsymbol{\tau}_{\mathbf{W}^\ell,\boldsymbol{u}^\ell}(\boldsymbol{x}^\ell) \;:=\; \boldsymbol{\sigma}(\mathbf{W}^\ell\boldsymbol{x}^\ell\!+\!\boldsymbol{u}^\ell) \tag{6}$$

where $\boldsymbol{\tau}^\ell\!:\!\mathbb{R}^{n_\ell}\!\to\!\mathbb{R}^{n_{\ell+1}}$ is the transfer function with synaptic weight matrix $\mathbf{W}^\ell\!\in\!\mathbb{R}^{n_{\ell+1}\times n_\ell}$ from layer ℓ to layer $\ell\!+\!1$, and $\boldsymbol{u}^\ell\!\in\!\mathbb{R}^{n_{\ell+1}}$ the biases of neurons in layer $\ell\!+\!1$, and n_ℓ the width of layer ℓ.

The NN input is $\boldsymbol{x}=\boldsymbol{x}^0\in\mathbb{R}^n$ $(n=n_0)$ and the output of an L-layer NN is $\boldsymbol{y}=\boldsymbol{x}^L\in\mathbb{R}^m$ $(m=n_L)$. The NN at layer $\ell\!+\!1$ computes the function $\boldsymbol{\nu}^{\ell+1}(\boldsymbol{x}):=\boldsymbol{\tau}^\ell(\boldsymbol{\nu}^\ell(\boldsymbol{x}))$ with $\boldsymbol{\nu}^0(\boldsymbol{x})\!:=\!\boldsymbol{x}$ and $\boldsymbol{y}\!:=\!\boldsymbol{\nu}^L(\boldsymbol{x})$. If σ has bounded range, functions f of greater range cannot be represented. For this reason, the non-linearity σ in the last layer is often removed, which we also assume when relevant, and call this an $L\!-\!1$-hidden layer NN.

Any σ which is continuously differentiable at least in a small region of its domain, which is virtually all σ used in practice, can approximate arbitrarily well linear hidden neurons and hence skip-connections by choosing very small/large weights [Pin99], which we occasionally exploit.

Literature on approximation results. There are a large number of NN representation theorems which tell us how well which functions f can be approximated, for

[10] For Deep networks this can be relaxed to non-linear.

function classes of different smoothness (e.g. Lipschitz continuous or $\mathcal{C}^k$), domain ($\mathbb{R}^n$ or a compact subset, typically $[0;1]^n$), for different distance measures (e.g. L^p-norm for $p\in[1;\infty]$ or Sobolev), asymptotic or finite bounds in terms of accuracy ε, network width $N:=\max\{n_0,...,n_L\}$ or depth L, esp. deep ($L\gg N=O(1)$) vs. shallow ($N\gg L=O(1)$), or number of neurons $n_+=n_1+...+n_L$, or total number of weights and biases $\sum_{\ell=0}^{L-1}(n_\ell+1)n_{\ell+1}$, or only counting the non-zero parameters, different activation functions σ, and that are only some choices within this most simple MLP NN model. See e.g. [Yar17, LSYZ20, GPEB19, RT18, LTR17] and for surveys [Pin99, FMZ19], to name a few. A powerful general approximation theory has been developed in [GPEB19], which very crudely interpreted shows that any function that can approximately be computed can also be computed by a NN with roughly comparable resources, which includes even fractal functions such as the Weierstrass function. Since NN can efficiently emulate Boolean circuits, this may not be too surprising. For simplicity we focus on the most-easy-to-state results, but briefly mention and reference improvements or variations.

Approximation results. We say that $\rho:\mathbb{R}^n\to\mathbb{R}^m$ *uniformly approximates* f on $[-D;D]^n$ for some $D>0$, or ρ *ε-approximates* f, if

$$||f-\rho||_\infty := \sup_{\boldsymbol{x}\in[-D;D]^n}\max_{1\leq i\leq n}|f_i(\boldsymbol{x})-\rho_i(\boldsymbol{x})| \leq \varepsilon$$

Other norms, most notably p-norms and Sobolev norms have been considered [Pin99]. Adaptation to other compact subsets of $\mathbb{R}^n$ of most results is straightforward, but results on all of $\mathbb{R}^n$ are rarer/weaker. We say that f can be *approximated by a function class* $\mathcal{F}\subseteq\mathbb{R}^n\to\mathbb{R}^m$ if for every $\varepsilon>0$ and D there exists a $\rho\in\mathcal{F}$ that ε-approximates f on (compact) hypercube $[-D;D]^n$, which is called convergence uniform on compacta. This is equivalent to f being in the topological closure of $\overline{\mathcal{F}}$ w.r.t. the compact-open topology. The class $\mathcal{F}$ we are interested in here is the class of MLPs defined above and subsets thereof:

$$\text{MLP} := \{\boldsymbol{\nu}^L : \mathbf{W}^\ell\in...,\boldsymbol{u}^\ell\in...;n_\ell\leq N,\ell\leq L;n,m,L,N\in\mathbb{N}\} \subseteq \mathbb{R}^*\to\mathbb{R}^*$$

Note that continuous functions on compact domains have compact range, which can be embedded in a hyper-cube, and finite compositions of continuous functions are continuous, hence each layer as well as the whole MLP maps $[-D;D]^n$ continuously into $[-D';D']^m$ for some D'. The two most important results for us are

Theorem 9 (NN approximation [Pin99]) *Every multivariate polynomial can be approximated by a (1-hidden-layer) MLP.*

Convergence is uniform on compacta, but also holds w.r.t. many other metrics and domains. For $\sigma=\max\{0,\cdot\}$ one can show that the depth L of the required network grows only with $O(\ln\varepsilon^{-1})$ and width N is constant. [GPEB19] shows $N=16$ and $L=O(d+\ln\varepsilon^{-1})$ for univariate polynomials of degree d, which easily generalizes to the multivariate case. Even better, for any twice continuously differentiable non-linear σ such as tanh, the size of the network does not even need to grow for $\varepsilon\to0$. Multiplication $x_1\cdot x_2$ can arbitrarily well be approximated by 4 neurons [LTR17], which then allows to compute any given polynomial of degree d in n variables consisting of m monomials to arbitrary precision by a NN of size $O(m\cdot n\cdot\ln d)$ independent ε [RT18].

The other result is the classical Stone-Weierstrass approximation theorem:

Theorem 10 (Stone-Weierstrass) *Every continuous function $f:\mathbb{R}^n\to\mathbb{R}^m$ can be approximated by m (multivariate) polynomials.*

Again convergence is uniform on compacta, but many extensions are known. Together with Theorem 9 this implies that MLPs can approximate any continuous function. For analytic f on $[-D;D]^n$ and $\sigma=\max\{0,\cdot\}$ (and likely most other σ), a NN of depth $L=O(\ln\varepsilon^{-1})$ and width $N=O(1)$ suffices [GPEB19]. For Lipschitz f on $[-D;D]^n$ and $\sigma=\max\{0,\cdot\}$, a NN of size $N\cdot L=\tilde{O}(m\cdot\varepsilon^{-n/2})$ suffices [LSYZ20, Table 1].

Both theorems together show that NN can approximate most functions well, and the discussion hints at, and [GPEB19] shows, with essentially optimal scaling in accuracy ε.

7 Universal Equivariant Networks

In this section we will restrict the representation power of MLPs to equivariant functions and prove their universality by construction in 4 steps. This is the penultimate step towards (anti)symmetric NN.

Equivariance and all-but-one symmetry. We are mostly interested in (anti)symmetric functions, but for (de)composition we need equivariant functions, and directly need to consider the d-dimensional case. A function $\boldsymbol{\varphi}:(\mathbb{R}^d)^n\to(\mathbb{R}^{d'})^n$ is called equivariant under permutations if $\boldsymbol{\varphi}(S_\pi^d(\mathbf{X}))=S_\pi^{d'}(\boldsymbol{\varphi}(\mathbf{X}))$ for all permutations $\pi\in S_n$. With slight abuse of notation we identify $(\boldsymbol{\varphi}(\mathbf{X}))_1\equiv\varphi_1(\mathbf{X})\equiv\varphi_1(\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_n)\equiv\varphi_1(\boldsymbol{x}_1,\boldsymbol{x}_{\neq 1})$ with $\varphi_1(\boldsymbol{x}_1|\boldsymbol{x}_{\neq 1})$. The following key Lemma shows that φ_1 suffices to describe all of $\boldsymbol{\varphi}$.

Lemma 11 (All-but-one symmetry = equivariance) *A function $\boldsymbol{\varphi}:(\mathbb{R}^d)^n\to(\mathbb{R}^{d'})^n$ is equivariant (under permutations) if and only if $\varphi_i(\mathbf{X})=\varphi_1(\boldsymbol{x}_i|\boldsymbol{x}_{\neq i})$ $\forall i$ and $\varphi_1(\boldsymbol{x}_i|\boldsymbol{x}_{\neq i})$ is symmetric in $\boldsymbol{x}_{\neq i}$.*

Proof. ($\Leftarrow$) $\varphi_i(S_\pi^d(\mathbf{X}))=\varphi_1(\boldsymbol{x}_{\pi(i)}|\boldsymbol{x}_{\pi(\neq i)})=\varphi_1(\boldsymbol{x}_{\pi(i)}|\boldsymbol{x}_{\neq\pi(i)})=\varphi_{\pi(i)}(\mathbf{X})=(S_\pi^{d'}(\boldsymbol{\varphi}(\mathbf{X})))_i$, where we used the abbreviation $\pi(\neq i)=(\pi(1),...,\pi(i-1),\pi(i+1),...,\pi(n))$. Note that $(\neq\pi(i))=(1,...,\pi(i)-1,\pi(i)+1,...,\pi(n))$ is the ordered index set, and equality holds by assumption on φ_1.
($\Rightarrow$) Assume $\pi(i)=i$, then $\varphi_1(\boldsymbol{x}_i|\boldsymbol{x}_{\pi(\neq 1)})=\varphi_1(S_\pi^d(\mathbf{X}))=(S_\pi^{d'}(\boldsymbol{\varphi}(\mathbf{X})))_1=\varphi_1(\mathbf{X})$, i.e. φ_1 is symmetric in $\boldsymbol{x}_{\neq i}$. Now assume $\pi(1)=i$, then $\varphi_i(\mathbf{X})=(S_\pi^{d'}(\boldsymbol{\varphi}(\mathbf{X})))_1=\varphi_1(S_\pi^d(\mathbf{X}))=\varphi_1(\boldsymbol{x}_i|\boldsymbol{x}_{\pi(2)},...\boldsymbol{x}_{\pi(n)})=\varphi_1(\boldsymbol{x}_i|\boldsymbol{x}_{\neq i})$. ■

Equivariant Neural Network. We aim at approximating equivariant $\boldsymbol{\varphi}$ by an Equivariant MLP (EMLP). The NN input is $\mathbf{X}\in\mathbb{R}^{d\times n}$, its output is $\mathbf{Y}\in\mathbb{R}^{d'\times n}$. Note that $d=3$ for a 3-dimensional physical n-particle system ($\boldsymbol{x}_i=(x_i,y_i,z_i)^\top$), but d' could be a "feature" "vector" of any length. We don't aim at modelling other within-vector symmetries, such as rotation or translation.

Layer $\ell+1\in\{1,...,L\}$ of EMLP is computed from layer ℓ by

$$\boldsymbol{x}_i^{\ell+1} := \tau_i^\ell(\mathbf{X}^\ell) := \tau_1^\ell(\boldsymbol{x}_i^\ell|\boldsymbol{x}_{\neq i}^\ell) \equiv \tau_{1,\mathbf{W}^\ell,\mathbf{V}^\ell,\boldsymbol{u}^\ell}(\boldsymbol{x}_i^\ell|\boldsymbol{x}_{\neq i}^\ell) := \boldsymbol{\sigma}(\mathbf{W}^\ell\boldsymbol{x}_i^\ell+\mathbf{V}^\ell\sum_{j\neq i}\boldsymbol{x}_j^\ell+\boldsymbol{u}^\ell) \quad (7)$$

where $\boldsymbol{\tau}^\ell:\mathbb{R}^{d_\ell\times n}\to\mathbb{R}^{d_{\ell+1}\times n}$ can be shown to be an equivariant "transfer" function with weight matrices $\mathbf{W}^\ell,\mathbf{V}^\ell\in\mathbb{R}^{d_{\ell+1}\times d_\ell}$ from layer ℓ to layer $\ell+1$, and $\boldsymbol{u}^\ell\in\mathbb{R}^{d_{\ell+1}}$ the biases of layer $\ell+1$, and $d_\ell\times n$ is the width of layer ℓ, now best viewed as a 2-dimensional array as in (convolutional) NN used in computer vision.[1]

The EMLP input is $\mathbf{X}=\mathbf{X}^0\in\mathbb{R}^{d\times n}$ $(d=d_0)$ and the output of an L-layer EMLP is $\mathbf{Y}=\mathbf{X}^L\in\mathbb{R}^{d'\times n}$ $(d'=d_L)$. The NN at layer $\ell+1$ computes the function $\boldsymbol{\nu}^{\ell+1}(\mathbf{X}):=\boldsymbol{\tau}^\ell(\boldsymbol{\nu}^\ell(\mathbf{X}))$ with $\boldsymbol{\nu}^0(\mathbf{X}):=\mathbf{X}$ and $\boldsymbol{\nu}^L(\mathbf{X})=\mathbf{Y}$.

Inspecting the argument of $\boldsymbol{\sigma}()$ shows that τ_1 is obviously invariant under permutation of $\boldsymbol{x}_{\neq i}$. It is also easy to see that the argument of $\boldsymbol{\sigma}()$ is the only linear function in $\mathbf{X}$ with such invariance [ZKR$^+$18, Lem.3]. Lemma 11 implies that $\boldsymbol{\tau}$ is equivariant. Note that if we allowed $\mathbf{W}$ or $\mathbf{V}$ or $\boldsymbol{b}$ to depend on i, this would no longer be the case. In other words, a single vector-valued function, symmetric in all-but-one vector, suffices to define any equivariant (matrix-valued) function. This tying of weights akin to convolutional networks reduces the number of weights by a factor $\approx n/2$. Since composition of equivariant functions is equivariant, $\boldsymbol{\nu}^L$ is equivariant, i.e. the network above can *only* represent equivariant functions.

The question we next answer is whether EMLPs can approximate *all* continuous equivariant functions. We show this in 4 steps: (1) representation of polynomials in a single vector $\boldsymbol{x}_i$, (2) symmetric polynomials in all-but-one vector, (3) equivariant polynomials, (4) equivariant continuous functions.

Polynomials in a single vector $\boldsymbol{x_i}$. If we set $\mathbf{V}^\ell=0$ in (7) we get $\boldsymbol{x}_i^{\ell+1}=\boldsymbol{\sigma}(\mathbf{W}^\ell\boldsymbol{x}_i^\ell+\boldsymbol{u}^\ell)$, which is one layer of a general MLP (6) in $\boldsymbol{x}_i$. Note that $\boldsymbol{\tau}_{1,\mathbf{W}^\ell,\mathbf{0},\boldsymbol{u}^\ell}$ and hence $\boldsymbol{\nu}^L[\mathbf{V}^\ell=0\forall\ell]$ compute the same function and independently for each $\boldsymbol{x}_i$. This can be interpreted as a factored MLP with n identical factors, or n independent identical MLPs each applied to one $\boldsymbol{x}_i$, or as one MLP applied to n different vectors $\boldsymbol{x}_i$.

Let $\rho_1:\mathbb{R}^{d''}\to\mathbb{R}^{d'''}$ be one such function computed by an EMLP with $\mathbf{V}^\ell=0\forall\ell$ to be specified later. That is, the factored NN computes $\boldsymbol{\rho}(\mathbf{X})=\rho_1(\boldsymbol{x}_i)_{i=1}^n$.

We need another factored $(\mathbf{V}^\ell=0\forall\ell)$ NN computing $\eta_1:\mathbb{R}^d\to\mathbb{R}^{d''}$, the multivariate polarized basis for $n-1$ (because of symmetry in $n-1$ variables only) d-dimensional vectors $\eta_1(\boldsymbol{x})=(x^{p_1}y^{p_2}...z^{p_d})_{1\leq p_1+...+p_d\leq n-1}$, where $d''=\binom{n-1+d}{d}-1$ (cf. Theorem 7).

Symmetric polynomials in all-but-one vector. If we concatenated $\boldsymbol{\eta}(\mathbf{X})=\eta_1(\boldsymbol{x}_i)_{i=1}^n$ with $\boldsymbol{\rho}$ we would get $\boldsymbol{\rho}(\boldsymbol{\eta}(\boldsymbol{X}))$, which is not what we want, but if we swap $\mathbf{V}^0=0$ with $\mathbf{W}^0$ in the NN for $\boldsymbol{\rho}$ and call it $\tilde{\boldsymbol{\rho}}$, it uses $\sum_{j\neq i}\boldsymbol{x}_j$ instead of $\boldsymbol{x}_i$ as input, so we get

$$\phi_1(\boldsymbol{x}_{\neq i}) \;:=\; \tilde\rho_1(\boldsymbol{\eta}(\boldsymbol{X})) \;=\; \rho_1(\boldsymbol{\beta}(\mathbf{X})), \tag{8}$$

$$\text{where}\quad \boldsymbol{\beta}(\mathbf{X})=\beta_1(\boldsymbol{x}_{\neq i})_{i=1}^n \quad\text{and}\quad \beta_1(\boldsymbol{x}_{\neq i}):=\sum_{j\neq i}\eta_1(\boldsymbol{x}_j) \tag{9}$$

is the multivariate polarized basis excluding $\boldsymbol{x}_i$. Now we know from Theorem 7 that any polynomial symmetric in $\boldsymbol{x}_{\neq i}$ can be represented as such ϕ_1 for suitable polynomial ρ_1. hence approximated by two concatenated EMLPs. By Lemma 11, $\boldsymbol{\rho}$, $\tilde{\boldsymbol{\rho}}$, $\boldsymbol{\beta}$, $\boldsymbol{\eta}$ are all equivariant, hence also $\boldsymbol{\phi}(\mathbf{X}):=\phi_1(\boldsymbol{x}_{\neq i})_{i=1}^n$ is, but the latter is not completely general (cf. Lemma 11), which we address now.

Equivariant polynomials. Next we construct polynomials $\varphi_1(\boldsymbol{x}_i|\boldsymbol{x}_{\neq i})$ symmetric in $\boldsymbol{x}_{\neq i}$. Any polynomial in n vectors can be written as a finite sum over $\boldsymbol{p}\equiv(p_1,...,p_d)\in P\subset\mathbb{N}_0^d$

[1] The formulation $\sum_{j\neq i}$ is from [PSMF20] which is slightly more convenient for our purpose but equivalent to unrestricted sum $\sum_j$ used by [ZKR$^+$18].

with $|P|<\infty$:

$$\varphi_1(\boldsymbol{x}_i|\boldsymbol{x}_{\neq i}) \;=\; \sum_{\boldsymbol{p}\in P}\eta_{\boldsymbol{p}}(\boldsymbol{x}_i)\cdot\text{Poly}_{\boldsymbol{p}}(\boldsymbol{x}_{\neq i}) \tag{10}$$

Since $\eta_{\boldsymbol{p}}(\boldsymbol{x})\equiv x^{p_1}y^{p_2}...z^{p_d}$ are different hence independent monomials, $\varphi_1(\boldsymbol{x}_i|\boldsymbol{x}_{\neq i})$ is invariant under permutations of $\boldsymbol{x}_{\neq i}$ if and only if all $\text{Poly}_{\boldsymbol{p}}$ are. The latter can all be represented by one large vector function consisting of the polynomials $\boldsymbol{x}_i':=\phi_1(\boldsymbol{x}_{\neq i})=(\text{Poly}_{\boldsymbol{p}}(\boldsymbol{x}_{\neq i}))_{\boldsymbol{p}\in P}$ *and* the monomials $\boldsymbol{x}_i'':=(\eta_{\boldsymbol{p}}(\boldsymbol{x}_i))_{\boldsymbol{p}\in P}$. Since $\boldsymbol{x}_i'$ and $\boldsymbol{x}_i''$ are both output in "channel" i of EMLPs, any $\varphi_1(\boldsymbol{x}_i|\boldsymbol{x}_{\neq i})=\boldsymbol{x}_i''^{\top}\boldsymbol{x}_i'$ is a scalar product (polynomial of degree 2) within channel i, hence can be computed by a factored EMLP. Now by Lemma 11, any equivariant (vector of) polynomials can be computed by an EMLP as $\boldsymbol{\varphi}(\mathbf{X})=\varphi_1(\boldsymbol{x}_i|\boldsymbol{x}_{\neq i})_{i=1}^n$.

Equivariant continuous functions. By Theorem 10, every continuous function can be approximated by a polynomial. We need a symmetric version of this, which is easy to obtain from the following Lemma:

Lemma 12 (Symmetric approximation) *Let $\mathcal{S}$ be a finite symmetry group with linear representations on $\mathbb{R}^n$ and $\mathbb{R}^m$. Let $f:\mathbb{R}^n\to\mathbb{R}^m$ be an equivariant function, i.e. $S(f(\boldsymbol{x}))=f(S(\boldsymbol{x}))\;\forall S\in\mathcal{S}$, Let $g:\mathbb{R}^n\to\mathbb{R}^m$ be an arbitrary function and $\overline{g}(\boldsymbol{x}):=\frac{1}{|\mathcal{S}|}\sum_{S\in\mathcal{S}}S^{-1}(g(S(\boldsymbol{x})))$ its linear symmetrization, and $||\cdot||$ be a norm invariant under $\mathcal{S}$, then $||f-\overline{g}||\leq||f-g||$.*

The proof is elementary and for permutations and symmetric/equivariant functions with ∞-norm we care about almost trivial. The Lemma also holds for compact groups with Haar measure and measurable functions, e.g. for rotations with Euclidean norm, but we do not need this.

Theorem 13 (Stone-Weierstrass for symmetric/equivariant functions) *Every continuous function, invariant/equivariant under permutations can be approximated by symmetric/equivariant polynomials, uniformly on compacta.*

Proof. Let g be an approximating polynomial of f, which exists by Theorem 10, and let $\overline{g}$ be its symmetrization. If f is invariant/equivariant under permutations, then by Lemma 12, $||f-\overline{g}||_\infty\leq||f-g||_\infty$, hence $\overline{g}$ also approximates f. Since a finite average of polynomials is a polynomial, and $\overline{g}$ is symmetric/equivariant by construction, this proves the theorem. ■

Since EMLPs can approximate all equivariant polynomials and only continuous equivariant functions, we get one of our main results:

Theorem 14 (Universality of (two-hidden-layer) EMLP) *For any continuous non-linear activation function, EMLPs can approximate (uniformly on compacta) all and only the equivariant continuous functions. If σ is non-polynomial, a two-hidden-layer EMLP suffices.*

Indeed, by inspecting our constructive proof, esp. (8) and (10), we see that in theory an EMLP with all-but-one layer being factored suffices, i.e. $\mathbf{V}^\ell=0$ for all-but-one ℓ. In practice we expect an EMLP allowing many/all layers to mix to perform better. Since 1-hidden-layer MLPs are universal for non-polynomial σ (Thm.9&10), the factored layers can be merged into 1 layer, leading to a 3-hidden-layer NN, with first and third layer being factored. It is easy to see that the second and third layer can actually be merged into one. We do not know whether a 1-hidden-layer EMLP is universal.

8 (Anti)Symmetric Networks

We are finally ready to combine all pieces and define (anti)symmetric NN, and state and discuss their universality properties. We briefly remark on why we believe the chosen approach is most suitable for deriving interesting error bounds.

Universal Symmmetric Network. We can approximate all and only the symmetric continuous functions by applying any symmetric continuous function $\varsigma:\mathbb{R}^{d'\times n}\to\mathbb{R}^{d'}$ with the property $\varsigma(\boldsymbol{y},...,\boldsymbol{y})=\boldsymbol{y}$ to the output of an EMLP, e.g. $\varsigma(\mathbf{Y})=\frac{1}{n}\sum_{i=1}^n\boldsymbol{y}_i$ or $\varsigma(\mathbf{Y})=\max\{y_1,...,y_n\}$ if $d'=1$. Clearly, the resulting function is symmetric under permutations. Also, if $\phi(\mathbf{X})$ is any symmetric function, then $\varphi_1(x_i|x_{\neq i}):=\phi(\mathbf{X})$ is clearly symmetric in $x_{\neq i}$, hence $\mathbf{Y}=\boldsymbol{\varphi}(\mathbf{X}):=\varphi_1(x_i|x_{\neq i})_{i=1}^n$ is equivariant and can be approximated by an EMLP, which, by applying ς to its output $\mathbf{Y}$, computes $\phi(\mathbf{X})=\varsigma(\mathbf{Y})$. Hence every symmetric continuous function can be approximated by an EMLP with a final average or max or other symmetric layer.

Note that the detour via EMLP was necessary to construct universal symmetric NNs. Assume we had started with an MLP for which every layer is a symmetric function, i.e. $\mathbf{V}^\ell=\mathbf{W}^\ell$. Such a network could only represent functions of the extremely restrictive form $\boldsymbol{y}_1=...=\boldsymbol{y}_n=\rho_1(\mathbf{W}\sum_{j=1}^n\boldsymbol{x}_j+\boldsymbol{u})$, where ρ_1 is an arbitrary continuous function.

Universal AntiSymmmetric Network. For $d=1$, any continuous AS function $\psi(x_{1:n})$ can be approximated by approximating the totally symmetric continuous function $\chi(x_{1:n}):=\psi(x_{1:n})/\prod_{j<i}(x_i-x_j)$ with an EMLP ($d'=1$), and then multiply the output by $\prod_{j<i}(x_i-x_j)$. But we are not limited to this specific construction: By Theorem 3 we know that every AS function can be represented as a GSD of n functions symmetric in all-but-one-variable. Note that the φ_i in the proof if combined to a vector $\boldsymbol{\varphi}$ is *not* equivariant, but for each i separately, $\tilde{\boldsymbol{\varphi}}_i(\mathbf{X}):=(\varphi_i(\boldsymbol{x}_j|\boldsymbol{x}_{\neq j}))_{j=1}^n$ is equivariant by Lemma 11, i.e. GSD needs an EMLP with $d'=n$.

For $d>1$, we can approximate AS $\psi(\mathbf{X})$ by approximating the n equivariant $\tilde{\boldsymbol{\varphi}}_i$, now with φ_i defined in the proof of Theorem 8, by an EMLP (again $d'=n$), and then take its Slater determinant. Note that the φ_i in the proof are defined in terms of a single symmetric function $\phi()$, which then gets anti-symmetrized essentially by multiplying with $\sigma(\bar{\pi})$. This shows that an EMLP computing a single symmetric function ($d'=1$) suffices, but this is *necessarily* and essentially always a discontinuous representation, while using the GSD with $d'=n$ equivariant functions *possibly* has a continuous representation.

Let us define a (toy) FermiNet as computing the GSD from the output of an EMLP. The real FermiNet developed in [PSMF20] contains a number of extra features, which improves practical performance, theoretically most notably particle pair representations. Since it is a superset of our toy definition, the following theorem also applies to the full FermiNet. . We arrived at the following result:

Theorem 15 (Universality of the FermiNet) *A FermiNet can approximate any continuous anti-symmetric function.*

For $d=1$, the approximation is again uniform on compacta. For $d>1$, the proof of Theorem 8 involves discontinuous $\varphi_i(\boldsymbol{x}_j|\boldsymbol{x}_{\neq j})$. Any discontinuous function can be approximated by continuous functions, not in ∞-norm but only weaker p-norm for $1\leq p<\infty$. This implies the theorem also holds for $d>1$ in L^p norm. Whether a stronger L^∞ result holds is an important open problem, important because approximating continuous functions by discontinuous components can cause all kinds of problems.

Approximation accuracy. The required NN size as a function of approximation accuracy for EMLPs should be similar to MLPs discussed in Section 6 with the following differences: Due to the permutation (anti)symmetry, weights $\mathbf{W}$,$\mathbf{V}$,$\boldsymbol{b}$ are shared between NN channels $i=1,...,n$, reducing the number of parameters by a factor of about $n/2$. On the other hand, the algebraic basis for multisymmetric polynomials has size $\binom{n+d}{d}\approx n^d$, which is crucially exploited in the polarized power basis, compared to $n\cdot d$ functions suffice for a functional basis. This of course does not mean that we need a NN of size $O(n^d)$ to accommodate all basis functions, but if there is a mismatch between the equivariant functions f we care about and the choice of basis, we may need most of them.

9 Discussion

Summary. We reviewed a variety of representations for (anti)symmetric functions $(\psi)\phi$: $(\mathbb{R}^d)^n\to\mathbb{R}$. The most direct and natural way is as a sum over $n!$ permutations of some other function χ. If $\chi\in\mathcal{C}^k$ then also $(\psi)\phi\in\mathcal{C}^k$. Unfortunately this takes exponential time, or at least is NP hard, and other direct approaches such as sampling or sorting have their own problems. The most promising approach is using Equivariant MLPs, for which we provided a constructive and complete universality proof, combined with a trivial symmetrization and a non-trivial anti-symmetrization using a large number Slater determinants. We investigated to which extent a *single generalized* Slater determinant introduced in [PSMF20], which can be computed in time $O(n^3)$, can represent all AS ψ. We have shown that for $d=1$, all AS $\psi\in\mathcal{C}^{k+n(n-1)/2}$ can be represented as $\det\Phi$ with $\Phi\in\mathcal{C}^k$. Whether $\Phi\in\mathcal{C}^k$ suffices to represent all $\psi\in\mathcal{C}^k$ is unknown for $k>0$. For $k=0$ it suffices. For $d>1$ and $n>2$, we were only able to show that AS ψ have representations using discontinuous Φ.

Open problems. Important problems regarding smoothness of the representation are open in the AS case. Whether continuous Φ can represent all continuous ψ is unknown for $d>1$, and similar for differentiability and for other properties. Indeed, whether *any* computationally efficient continuous representation of all and only AS ψ is possible is unknown.

Outlook: Error bounds. Our construction via multisymmetric polynomials is arguably more natural, and can serve as a starting point for interesting error bounds for function classes that can be represented well by polynomials, e.g. functions of different degree of smoothness. Many such results are known for general NN [Pin99], most of them are based on polynomial approximations. We therefore expect that the techniques transfer to symmetric functions, with similar bounds, and to AS and $d=1$ with worse bounds due to loss of differentiability. For $d>1$ we are lacking smoothness-preserving results. If and only if they can be established, we can expect error bounds for this case too. In any case, this is beyond the scope of this already overlong paper.

References

[Act18] Jonas Actor. *Computation for the Kolmogorov Superposition Theorem.* Thesis, May 2018.

[Bar93] A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, May 1993.

[FMZ19] Jianqing Fan, Cong Ma, and Yiqiao Zhong. A Selective Overview of Deep Learning. *arXiv:1904.05526 [cs, math, stat]*, April 2019.

[GPEB19] Philipp Grohs, Dmytro Perekrestenko, Dennis Elbrächter, and Helmut Bölcskei. Deep Neural Network Approximation Theory. *arXiv:1901.02220 [cs, math, stat]*, January 2019.

[HLL$^+$19] Jiequn Han, Yingzhou Li, Lin Lin, Jianfeng Lu, Jiefu Zhang, and Linfeng Zhang. Universal approximation of symmetric and anti-symmetric functions. *arXiv:1912.01765 [physics]*, December 2019.

[Kol57] Andrej Kolmogorov. On the Reprepsentation of Continuous Functions of Several Variables as Superpositions of Continuous Functions of One Variable and Addition. 1957.

[Liu15] Xing Liu. Kolmogorov superposition theorem and its applications. September 2015.

[LSYZ20] Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep Network Approximation for Smooth Functions. *arXiv:2001.03040 [cs, math, stat]*, January 2020.

[LTR17] Henry W. Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, September 2017.

[Mak96] Y. Makovoz. Random Approximants and Neural Networks. *Journal of Approximation Theory*, 85(1):98–109, April 1996.

[Mit07] Lubos Mitas. Topology of fermion nodes and pfaffian wavefunctions, 2007. http://nano-bio.ehu.es/files/Topology_of_fermion_nodes_and_pfaffian_wavefunctions-Mitas.pdf.

[Pin99] Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8:143–195, January 1999.

[PSMF20] David Pfau, James S. Spencer, Alexander G. D. G. Matthews, and W. M. C. Foulkes. Ab-Initio Solution of the Many-Electron Schrödinger Equation with Deep Neural Networks. *Physical Review Research*, 2(3):033429, September 2020.

[RT18] David Rolnick and Max Tegmark. The power of deeper networks for expressing natural functions. In *ICLR*, 2018.

[SI19] Akiyoshi Sannai and Masaaki Imaizumi. Improved Generalization Bound of Group Invariant / Equivariant Deep Networks via Quotient Feature Space. *arXiv:1910.06552 [cs, stat]*, 2019.

[Wan95] Jun Wang. Analysis and design of an analog sorting network. *IEEE Transactions on Neural Networks*, 6(4):962–971, July 1995.

[Wey46] Hermann Weyl. *The Classical Groups: Their Invariants and Representations.* Princeton Landmarks in Mathematics and Physics Mathematics. Princeton University Press, Princeton, N.J. Chichester, 2nd ed., with suppl edition, 1946.

[WFE$^+$19] Edward Wagstaff, Fabian B. Fuchs, Martin Engelcke, Ingmar Posner, and Michael Osborne. On the Limitations of Representing Functions on Sets. In *ICML*, October 2019.

[Yar17] Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

[ZKR$^+$18] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, pages 3391–3401, April 2018.

A List of Notation

Symbol	**Explanation**
AS	Anti-Symmetric
NN	Neural Network
MLP	Multi-Layer Perceptron
EMLP	Equivariant Multi-Layer Perceptron
GSD	Generalized Slater Determinant
$n \in \mathbb{N}$	number of particles (in physics applications)
$i,j \in \{1:n\}$	particle index/number
$d \in \mathbb{N}$	dimensionality of particles (in physics applications esp. $d=3$)
$x \in \mathbb{R}$	real argument of function or input to NN, particle coordinate
$\mathbf{x} \in \mathbb{R}^n$	$\mathbf{x}=(x_1,...,x_n)$, function argument, NN input, n 1d particles ($d=1$)
$\boldsymbol{x} \in \mathbb{R}^d$	$\boldsymbol{x}=(x,y,...,z)^\top$ vector of coordinates of one d-dimensional particle
$\mathbf{X} \in \mathbb{R}^{d\times n}$	$\mathbf{X}=(\boldsymbol{x}_1,...,\boldsymbol{x}_n)$ matrix of n d-dimensional particles
$S_n \subseteq \{1:n\} \to \{1:n\}$	permutation group
$\pi \in S_n$	permutation of $(1,...,n)$
$\mathcal{S}_n \subset \mathbb{R}^n \to \mathbb{R}^n$	canonical linear representation of permutation group
$S_\pi \in \mathcal{S}_n$	$S_\pi(x_1,...,x_n) := (x_{\pi(1)},...,x_{\pi(n)})$
$\mathcal{S}_n^d \neq \mathcal{S}_{n\cdot d}$	d copies of linear permutation group representations
$S_\pi^d \in \mathcal{S}_n^d$	$S_\pi^d(\boldsymbol{x}_1,...,\boldsymbol{x}_n) := (\boldsymbol{x}_{\pi(1)},...,\boldsymbol{x}_{\pi(n)})$
$f : \mathbb{R}^{d\times n} \to \mathbb{R}$	some function to be approximated by an EMLP
$\chi : \mathbb{R}^{d\times n} \to \mathbb{R}$	general function of n d-dimensional particles
$\phi : \mathbb{R}^{d\times n} \to \mathbb{R}$	symmetric function: $\phi(S_\pi^d(\mathbf{X})) = \phi(\mathbf{X})$
$\psi : \mathbb{R}^{d\times n} \to \mathbb{R}$	anti-symmetric (AS) function: $\psi(S_\pi^d(\mathbf{X})) = \sigma(\pi)\psi(\mathbf{X})$
$\sigma(\pi) = \pm 1$	parity or sign of permutation π
$f_{\mathcal{S}}$	function f linearly symmetrized by $\mathcal{S}$
$\boldsymbol{x}_{\neq i}$	$\equiv (\boldsymbol{x}_1,...,\boldsymbol{x}_{i-1},\boldsymbol{x}_{i+1},...,\boldsymbol{x}_n)$ all but particle i
$\varphi(\boldsymbol{x}_i \vert \boldsymbol{x}_{\neq i})$	function symmetric in $n-1$ arguments $\boldsymbol{x}_{\neq i}$
$b \in \{1:m\}$	index of basis function
$\beta_b(\mathbf{X}) \in \mathbb{R}$	bth basis function, e.g. $\boldsymbol{\beta}_b(\mathbf{X}) = \sum_{i=1}^n \boldsymbol{\eta}_b(\boldsymbol{x}_i)$
$\eta_b(\boldsymbol{x}_j)$	bth polarized basis function
$g : \mathbb{R}^m \to \mathbb{R}$	composition or outer function, used as $g(\boldsymbol{\beta}(\mathbf{X}))$
ν	some NN/MLP/EMLP function
ρ	some (multivariate) polynomial