Scalars are universal: Equivariant machine learning, structured like classical physics

Soledad Villar
Department of Applied Mathematics and Statistics
Johns Hopkins University

David W. Hogg
Flatiron Institute
a divison of the Simons Foundation

Kate Storey-Fisher
Center for Cosmology and Particle Physics
Department of Physics, New York University

Weichi Yao
Department of Technology, Operations, and Statistics
Stern School of Business, New York University

Ben Blum-Smith
Center for Data Science
New York University

Abstract

There has been enormous progress in the last few years in designing neural networks that respect the fundamental symmetries and coordinate freedoms of physical law. Some of these frameworks make use of irreducible representations, some make use of high-order tensor objects, and some apply symmetry-enforcing constraints. Different physical laws obey different combinations of fundamental symmetries, but a large fraction (possibly all) of classical physics is equivariant to translation, rotation, reflection (parity), boost (relativity), and permutations. Here we show that it is simple to parameterize universally approximating polynomial functions that are equivariant under these symmetries, or under the Euclidean, Lorentz, and Poincaré groups, at any dimensionality d. The key observation is that nonlinear O(d)-equivariant (and related-group-equivariant) functions can be universally expressed in terms of a lightweight collection of scalars—scalar products and scalar contractions of the scalar, vector, and tensor inputs. We complement our theory with numerical examples that show that the scalar-based method is simple, efficient, and scalable.

1 Introduction

There is a great deal of current interest in building machine-learning methods that respect exact or approximate symmetries, such as translation, rotation, and physical gauge symmetries [19, 50]. Some of this interest is inspired by the great success of convolutional neural networks (CNNs) [52], which are naturally translation equivariant. The implementation of convolutional layers in CNNs has been given significant credit for the success of deep learning, in a domain (natural images) in which the convolutional symmetry is only approximately valid. In many data-analysis problems in astronomy, physics, and chemistry there are *exact* symmetries that must be obeyed by any generalizable law or rule. Since the approximate symmetries introduced by convolutional networks help in the natural-image domain, then we have high hopes for the value of encoding exact symmetries for problems where these symmetries are known to hold exactly.

In detail, the symmetries of physics are legion. Translation symmetry (including conservation of linear momentum), rotation symmetry (including conservation of angular momentum), and time-

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

translation symmetry (including conservation of energy) are the famous symmetries [70]. But there are many more: there is a form of reflection symmetry (charge-parity-time or CPT); there is a symplectic symmetry that permits reinterpretations of positions and momenta; there are Lorentz and Poincaré symmetries (the fundamental symmetries of Galilean relativity and special relativity) that include velocity boosts; there is the generalization of these (general covariance) that applies in curved spacetime; there are symmetries associated with baryon number, lepton number, flavor, and color; and there are dimensional and units symmetries (not usually listed as symmetries, but they are) that restrict what kinds of quantities can be multiplied or added. If it were possible to parameterize a universal or universally approximating function space that is explicitly equivariant under a large set of non-trivial symmetries, that function space would—in some sense—contain within it *every possible law of physics*. It would also provide a basis for good new machine learning methods.

The most expressive approaches to equivariant machine learning make use of irreducible representations of the relevant symmetry groups. Implementing these approaches requires a way to explicitly decompose tensor products of known representations into their irreducible components. This is called the *Clebsch-Gordan problem*. The solution for SO(3) has been implemented, and there is recent exciting progress for other Lie groups [1, 41]. This is an area of current, active research.

Here we give an approach that bypasses the need for a solution to this problem. We find that, for a large class of problems relevant to classical physics, the space of equivariant functions can be constructed from functions only of a complete subset of the scalar products and scalar contractions of the input vectors and tensors. That is, invariant scalars are powerful objects, and point the way to building universally approximating functions that are constrained by exact, non-trivial symmetries.

Equivariance is the form of symmetry core to physical law: For an equivariant function, transforming the input results in an output representation transformed in the same way. An *invariant* function, on the other hand, produces the same output for transformed and non-transformed inputs.

Definition 1*:* Given a function $f : \mathcal{X} \to \mathcal{Y}$ and a group G acting on $\mathcal{X}$ and in $\mathcal{Y}$ as $\star$ (possibly the action is defined differently in $\mathcal{X}$ and $\mathcal{Y}$). We say that f is:

$$G\text{-invariant:} \quad f(g\star x) = f(x) \quad \text{for all } x\in\mathcal{X},\ g\in G\ ; \tag{1}$$

$$G\text{-equivariant:} \quad f(g\star x) = g\star f(x) \quad \text{for all } x\in\mathcal{X},\ g\in G\ . \tag{2}$$

Our contributions: In this work we provide a complete and computationally tractable characterization of all scalar functions $f : (\mathbb{R}^d)^n \to \mathbb{R}$, and of all vector functions $h : (\mathbb{R}^d)^n \to \mathbb{R}^d$ that satisfy all of the symmetries of classical physics. The groups corresponding to these symmetries are given in Table 1; they act according to the rules in Table 2. The characterization we provide is physically principled: It is based on invariant scalars. It is also connected to the symmetries encoded in the Einstein summation rules, a common notation in physics to write expressions compactly but that also allows only equivariant objects to be produced (see Appendix F).

Our characterization is based on simple mathematical observations. The first is the First Fundamental Theorem of Invariant Theory for O(d): *a function of vector inputs returns an invariant scalar if and only if it can be written as a function only of the invariant scalar products of the input vectors* [94, Section II.A.9]. There is a similar statement for the Lorentz group O$(1, d)$. The second observation is that *a function of vector inputs returns an equivariant vector if and only if it can be written as a linear combination of invariant scalar functions times the input vectors*. In particular, if $h : (\mathbb{R}^d)^n \to \mathbb{R}^d$ of inputs $v_1, . . , v_n$ is O(d) or O$(1, d)$-equivariant, then it can be expressed as:

$$h(v_1, v_2, . . , v_n) = \sum_{t=1}^{n} f_t\Big(\langle v_i, v_j\rangle_{i,j=1}^{n}\Big)\, v_t\ , \tag{3}$$

where f_t can be arbitrary functions, but if h is a polynomial function the f_t can be chosen to be polynomials. In other words, the O(d) and O$(1, d)$-equivariant vector functions are generated as a module over the ring of invariant scalar functions by the projections to each input vector. In this expression, $\langle\cdot,\cdot\rangle$ denotes the invariant scalar product, which can be the usual Euclidean inner product, or the Minkowski inner product defined in terms of a metric Λ (see Table 1):

$$\text{Euclidean: } \langle v_i, v_j\rangle = v_i^\top v_j\ , \qquad \text{Minkowski: } \langle v_i, v_j\rangle = v_i^\top \Lambda\, v_j\ . \tag{4}$$

Our mathematical observations lead to a simple characterization for a very general class of equivariant functions, simpler than any based on irreducible representations or the imposition of symmetries

Orthogonal	$\mathrm{O}(d) = \{Q \in \mathbb{R}^{d\times d} : Q^\top Q = Q\,Q^\top = I_d\}$,
Rotation	$\mathrm{SO}(d) = \{Q \in \mathbb{R}^{d\times d} : Q^\top Q = Q\,Q^\top = I_d,\ \det(Q) = 1\}$
Translation	$\mathrm{T}(d) = \{w \in \mathbb{R}^d\}$
Euclidean	$\mathrm{E}(d) = \mathrm{T}(d) \rtimes \mathrm{O}(d)$
Lorentz	$\mathrm{O}(1,d) = \{Q \in \mathbb{R}^{(d+1)\times(d+1)} : Q^\top \Lambda\, Q = \Lambda,\ \Lambda = \mathrm{diag}([1,-1,\ldotp\ldotp,-1])\}$
Poincaré	$\mathrm{IO}(1,d) = \mathrm{T}(d+1) \rtimes \mathrm{O}(1,d)$
Permutation	$\mathrm{S}_n = \{\sigma : [n] \to [n]$ bijective function$\}$

Table 1: **The groups considered in this work.**

Orthogonal; Lorentz	$Q \star (v_1, \ldotp\ldotp, v_n) = (Q\,v_1, \ldotp\ldotp, Q\,v_n)$
Translation	$w \star (v_1, \ldotp\ldotp, v_n) = (v_1 + w, \ldotp\ldotp, v_k + w, v_{k+1}, \ldotp\ldotp, v_n)$ (where the first k vectors are position vectors)
Euclidean; Poincaré	$(w, Q) \star (v_1, \ldotp\ldotp, v_n) = (Q\,v_1 + w, \ldotp\ldotp, Q\,v_k + w, Q\,v_{k+1}, \ldotp\ldotp, Q\,v_n)$
Permutation	$\sigma \star (v_1, \ldotp\ldotp, v_n) = (v_{\sigma(1)}, \ldotp\ldotp, v_{\sigma(n)})$

Table 2: **The actions of the groups on vectors.** For the Euclidean group, the position vectors are positions of points; for the Poincaré group, the position vectors are positions of *events*.

through constraints (these methods are currently state of the art; see Section 2). This implies that very simple neural networks based on scalar products of input vectors—that enormously generalize those in [81]—can universally approximate invariant and equivariant functions. This justifies the numerical success of the neural network model proposed in [81], and mathematically shows that their method can be extended to a universal equivariant architecture with respect to more general group actions. We note that the formulation in (3) might superficially resemble an attention mechanism [88], but it actually comes from the characterization of invariant functions of the group, in particular the picture for SO(d)-invariant functions is a little different (see Proposition 5).

In Section 2 (and Appendix A) we describe the state of the art for encoding symmetries, the expressive power of graph neural networks, and universal approximation of invariant and equivariant functions. In Section 3 we mathematically characterize the invariant and equivariant functions with respect to the groups in Table 1. In Section 4 we present some examples of physically meaningful equivariant functions, and show how to express them in the parameterization developed in Section 3. In Section 5 and Appendix G we discuss which (of all possible) pairwise inner products one ought to provide, and in Section 6 we discuss some limitations of our approach. We present numerical experiments using our scalar-based approach compared to other methods in Section 7 (see also [97]).

We also note that the symmetries considered in this work are all global symmetries, as they act on all points in the same way. Our characterization thus does not obviously generalize to all gauge symmetries, which are local symmetries that apply changes independently to points at different locations (see [12, 93]). That said, we believe our model could be made general enough to encompass gauge symmetries if we replace the global metric by any position-dependent metric Λ_x. In this case, spatially separated vectors would need to be propagated to the same point such that they can be input to locally invariant functions, and this can be done with parallel transport. The parallel transport operations would also have to obey our invariance characterization, but we believe this is possible to do; we will explore it in future work.

2 Related work

Group invariant and equivariant neural networks: Symmetries have been used successfully for learning functions on images, sets, point clouds, and graphs. Neural networks can be designed to parameterize classes of functions satisfying different forms of symmetries, from the classical (approximately) translation-invariant convolutional neural networks [53]—as well as new approaches that enforce additional symmetries (rotation, scale) on these networks [91, 30, 86]—to more recent architectures that define permutation invariant and equivariant functions on point clouds pioneered by deep sets and pointnets [100, 74, 75], to permutation-equivariant functions on graphs expressed as graph neural networks [34, 82, 48, 24, 31, 16].

For instance, deep sets and pointnets parameterize functions on $(\mathbb{R}^d)^n$ that are invariant or equivariant with respect to the group of permutations S_n acting as in Table 2. Invariant theory shows that all invariant and equivariant functions with respect to such actions can be approximated by easily characterized invariant polynomials [94]. However, the permutation group can act in significantly more complicated ways. For instance, graph neural networks are equivariant with respect to a different action by permutations (conjugation) that is much harder to characterize (see Appendix A).

Since being introduced by [74, 100], deep learning on point clouds has been extremely fruitful, especially in computer vision [37]. Recently, new symmetries and invariances have been incorporated into the design of neural networks on point clouds; especially invariances and equivariances with respect to rigid motions such as translations and rotations. Many architectures have been proposed to satisfy those symmetries such as [86] based on irreducible representations (irreps), [51, 101] based on convolutions, [30] employing spherical harmonics and irreps, [102] using quaternions, and [29] applying a set of constraints to satisfy the symmetries. Most of the implementations of the approaches mentioned above (except for [29]) are limited to 2D or 3D point clouds. We provide an overview of the main approaches below and in Appendix A.

Recently [76] developed an approach to enforcing permutation equivariance in neural network layers using parameter sharing, which can then model other symmetry groups. The weight sharing approach is significantly simpler to implement than the ones described above, and it has been proven to be very successful in practice, with several applications [90, 91], including autonomous driving [39]. Characterizing the space of the functions this approach can express is an interesting open problem.

Universal approximation via linear invariant layers and irreducible representations: Universally approximating invariant functions can be obtained by taking universal non-invariant functions and averaging them over the group orbits [98, 69]. However, this approach is not practical for large groups like S_n or infinite groups like $\mathrm{O}(d)$. A classical result in neural networks shows that feed-forward networks with non-polynomial activations can universally approximate continuous functions [55]. This arguably inspired the use of neural networks that are the composition of linear invariant or equivariant layers with compatible non-linear activation functions to create expressive equivariant models [50, 64, 66]. Linear G-equivariant functions can be written in terms of the irreducible representations of G (we explain this and refer the reader to related literature in Appendix A). However, the explicit parameterization of the linear maps is only known for a few groups (for instance, SO(3)-equivariant linear maps are parameterized using the Clebsh-Gordan coefficients [30, 86, 9]). Moreover, very recent work [25] shows that the classes of functions defined in terms of neural networks over irreducible representations in [30, 86] are universal. In particular, every continous SO(3)-equivariant function can be approximated uniformly in compacts sets by those neural networks. Despite universality, there is a limitation to this approach: Although decompositions into irreps are broadly studied in mathematics (also as plethysms), the explicit transformation to the irreps is not known or possible for general groups. This is in fact an area of current, active research, where there has been recent exciting progress for other Lie groups [1, 41], but the implementation is still limited.

Expressive power of (non-universal) neural networks on graphs: Graph neural networks express functions on graphs that are equivariant with respect to a certain action by permutations. However, the architectures that are used in practice, typically based on message passing [31] or graph convolutions [24, 21], are not universal in general. Implementing a universally approximating graph neural network using the formulation from the previous section would be prohibitively expensive. There is work characterizing the expressive power of message passing neural networks, mainly in terms of the graph isomorphism problem [95, 68, 17, 60, 85, 15], and there is research on the design of graph networks that are expressive enough to perform specific tasks, like solving specific combinatorial optimization problems [6, 14, 71, 45, 96, 44, 10].

Machine learning for physics with symmetry preservation: Machine learning has been applied extensively to problems in physics. While many of these problems require certain symmetries—in many cases, exact symmetries—most applied projects to date rely on the data to encode the symmetry and hope that the model learns it. For instance, CNNs are commonly used to classify galaxy images; data augmentation is used to teach the model rotational symmetry [40, 3, 23, 32]. One well-known example is the Kaggle Galaxy Challenge, a classification competition based on the

Galaxy Zoo project [59]; the winning model improved performance by concatenating features from transformed images of each galaxy before further training [22].

There have been recent successes in enforcing physical symmetries in the architecture of the models themselves [99], for instance, in weather and climate modeling [46] and in modeling chaotic dynamical systems such as turbulence [90, 91]. In quantum many-body physics, recent work has shown that the symmetries of quantum energy levels on lattices can be enforced with gauge equivariant and invariant neural networks [18, 11, 89, 61, 79, 62]. There is significant work on imposing permutation symmetry in jet assignment for high-energy particle collider experiments with self-attention networks [28, 54]. In molecular dynamics, rotationally invariant neural networks have been shown to better learn molecular properties [2, 92, 83], and Hamiltonian neural networks have been constructed to better preserve molecular conformations [56]. More broadly, Hamiltonian networks have been shown to improve physical characteristics, such as better conservation of energy, and to better generalize [35, 80, 103], and Lagrangian neural networks can also enforce conservation laws [63, 20].

Invariant theory as a basis for enforcing symmetry in neural networks: We are aware of two lines of prior work that develop approaches related to the one taken here: [57, 58] and [36, 38].

In [57] certain tasks in turbulence modeling and materials science are considered, which have built-in O(3) and (in the latter case) octahedral symmetry. Each of these problems involves a specific representation of the given symmetry group, for which an explicit generating set for the invariant algebra (there called an "integrity basis") is known. The authors construct and test models that learn an invariant function, built on these generating sets. In [58], the turbulence example is taken up again, this time with the goal of learning an equivariant 2-tensor.

The idea of using the invariant algebra to enforce physical symmetries is also contemplated in [36]. In this work, the authors are focused on developing the underlying invariant theory in the case of simultaneous Lorentz and permutation invariance. In the followup work [38], this idea is developed into three models, using different generating sets, which are tested against each other.

The present work shares with these works the idea of using invariant-theoretic descriptions of invariant functions to hard-code physical symmetries into a neural network that models a physical system. We work in somewhat greater generality. While [57, 58] focus on specific (small) representations of O(3) and finite extensions, and while [36, 38] are focused on invariant functions and restricted to linearly reductive groups, we consider both invariant and equivariant functions for almost all groups relevant to physics, including groups (Euclidean and Poincaré) that are not reductive, although their invariant theory remains under control.

Inductive bias benefits of incorporating symmetries: The value of incorporating exact symmetries in machine learning has been recently established empirically in several applications (see for instance [91]). Mathematical theory has been developed to explain how much one can improve in terms of sample complexity and generalization [67, 7, 27] but many questions remain open.

3 Equivariant maps

In this Section we provide a simple characterization of all invariant and equivariant functions with respect to the actions in Table 2 by the groups in Table 1. The proof technique involves the characterization of equivariant maps from knowledge of the invariants, and it is explained in more generality in [8]. In what follows, $v_1, v_2, .., v_n$ will be vectors in $\mathbb{R}^d$, G will be a group acting in $\mathbb{R}^d$ and $(\mathbb{R}^d)^n$ as in Table 2, $f : (\mathbb{R}^d)^n \to \mathbb{R}$ will be an invariant function with respect to the action, and $h : (\mathbb{R}^d)^n \to \mathbb{R}^d$ will be an equivariant function with respect to the same action. V will denote a $d \times n$ matrix whose columns are the v_i vectors.

O(d) and SO(d) invariance and equivariance: The following classical result in invariant theory (e.g., [94, Section II.A.9]) shows that O(d) invariant functions are functions of the scalars $v_i^\top v_j$.

Lemma 1 (First Fundamental Theorem for O(d))*:* If f is an O(d)-invariant scalar function of vector inputs $v_1, .., v_n \in \mathbb{R}^d$, then $f(v_1, v_2, .., v_n)$ can be written as a function of only the scalar products of the v_i. That is, there is a function $g(\cdot)$ such that

$$f(v_1, v_2, .., v_n) = g(V^\top V) = g\big((v_i^\top v_j)_{i,j=1}^n\big) \ . \tag{5}$$

Proof. Given $M = V^\top V \in \mathbb{R}^{n\times n}$, we can reconstruct $v_1,..,v_n$ modulo the orthogonal group O(d) by computing the Cholesky decomposition of M (see for instance [87] p. 174). Therefore, the function f is uniquely determined by the inner product matrix $V^\top V$. □

The classical theorem also includes the fact that if f is polynomial (in the entries of the v_j's), then g can be taken to be polynomial. In Section 5 we observe that the function g can be determined by a small subset of the scalars $v_i^\top v_j$. There is an analogous statement for SO(d) (again see [94, Section II.A.9]):

Lemma 2 (First Fundamental Theorem for SO(d))*:* If f is an SO(d)-invariant scalar function of vector inputs $v_1, v_2, .., v_n \in \mathbb{R}^d$, then $f(v_1, v_2, .., v_n)$ can be written as a function of the scalar products of the v_i and the $d \times d$ subdeterminants of the $d \times n$ matrix V.

Lemma 3*:* If h is an O(d)-equivariant vector function of n vector inputs $v_1, v_2, .., v_n$, then $h(v_1, v_2, .., v_n)$ must lie in the subspace spanned by the input vectors $v_1, v_2, .., v_n$.

Proof. Let $\{w_1, .., w_r\} \subset \mathbb{R}^d$ be an orthonormal basis of the orthogonal complement to span($v_1, .., v_n$). Then we can write $h(v_1, .., v_n) = \sum_{t=1}^n \alpha_t v_t + \sum_{t=1}^r \beta_t w_t$ for some choice of α_t, β_t. We claim that the equivariance of h implies $\beta_t = 0$ for $t = 1, .., r$: Consider $\hat{Q} \in O(d)$ such that $\hat{Q}(v) = v$ for all span($v_1, .., v_n$), and $\hat{Q}(w) = -w$ for all w in the orthogonal complement.

Since $(\hat{Q}v_1, .., \hat{Q}v_n) = (v_1, .., v_n)$ we have $h(\hat{Q}v_1, .., \hat{Q}v_n) = \sum_{t=1}^d \alpha_t v_t + \sum_{t=1}^r \beta_t w_t$ while $\hat{Q}(h(v_1, .., v_n)) = \sum_{t=1}^d \alpha_t v_t - \sum_{t=1}^r \beta_t w_t$. Therefore equivariance implies that all $\beta_t = 0$. □

Note that Lemma 3 doesn't hold for SO(d): although when the codimension of span($v_1, \ldots, v_n$) is 0 or ≥ 2 a similar argument works, the situation $\dim_{\mathbb{R}} \text{span}(v_1, \ldots, v_n) = d-1$ breaks the proof, and there do exist equivariant vector functions that don't lie in the span when it has codimension 1. For instance, the cross product of two vectors in $\mathbb{R}^3$ is an SO(3)-equivariant function that is not in the span of its inputs. Proposition 5 shows that generalized cross products are the *only* way that SO(d)-equivariant vector functions can escape the span of the inputs. We further discuss this in Appendix F. Proposition 4 below gives a characterization of all O(d)-equivariant functions in terms of the scalars. We prove it in Appendix B. See [8] for a more general explanation of the proof technique.

Proposition 4*:* If h is an O(d)-equivariant vector function of n vector inputs $v_1, v_2, .., v_n$, then there are n O(d)-invariant scalar functions $f_t(\cdot)$ such that

$$h(v_1, v_2, .., v_n) = \textstyle\sum_{t=1}^n f_t(v_1, v_2, .., v_n)\, v_t\ . \tag{6}$$

Moreover, if h is a polynomial function, the f_t can be chosen to be polynomial.

The equivariant scalar functions form a module over the ring of invariant scalar functions. Proposition 4 states that this module is generated by the projections $(v_1, \ldots, v_n) \mapsto v_j$. Proposition 5 extends this result to SO(d) in terms of the generalized cross product. The definition of the generalized cross product and the proof of Proposition 5 are in Appendix B.

Proposition 5*:* If h is an SO(d)-equivariant vector function of n vector inputs $v_1, v_2, .., v_n$ then a similar characterization of Proposition 4 holds (with SO(d)-invariant scalar coefficients), except when $v_1, v_2, .., v_n$ span a $(d-1)$-dimensional space. In that case, there exist SO(d)-invariant scalar functions $f_t(\cdot)$ and $f_S(\cdot)$ such that

$$h(v_1, v_2, .., v_n) = \textstyle\sum_{t=1}^n f_t(v_1, v_2, .., v_n)\, v_t + \sum_{S \in \binom{[n]}{d-1}} f_S(v_1, v_2, .., v_n)\, v_S\ , \tag{7}$$

where $[n] := \{1, \ldots, n\}$, $\binom{[n]}{d-1}$ is the set of all $(d-1)$-subsets of $[n]$, and v_S represents the generalized cross product of vectors v_j with $j \in S$ (taken in ascending order). Moreover, if h is polynomial, the f_t and f_S can be taken to be polynomial.

E(d) invariance and equivariance: When modelling point clouds, we may want to express functions that are translation invariant or equivariant with respect to a subset of the input vectors (for instance, position vectors are translation equivariant). To this end, we consider the group of translations parametrized by $w \in \mathbb{R}^d$ acting on the vectors in $(\mathbb{R}^d)^n$ by translating the position vectors, and leaving everything else unchanged: $w \star (v_1, .., v_k, v_{k+1}, .., v_n) = (v_1 + w, .., v_k + w, v_{k+1}, .., v_n)$. In this Section we characterize all functions that are translation and rotation invariant/equivariant. In the exposition below we assume for simplicity that all vectors are position vectors, but the results generalize trivially to a mix of vectors.

Lemma 6*:* Any translation-invariant function $f : (\mathbb{R}^d)^n \to \mathbb{R}^\ell$ with inputs $v_1, .., v_n$ can be written uniquely as $f(v_1, v_2, .., v_n) = \tilde{f}(v_2 - v_1, .., v_n - v_1)$, where $\tilde{f} : (\mathbb{R}^d)^{n-1} \to \mathbb{R}^\ell$ is an arbitrary function. If f is polynomial, $\tilde{f}$ is polynomial, and vice versa. If f is equivariant for the action of any given subgroup $G \subset GL(n, \mathbb{R})$, then so is $\tilde{f}$, and vice versa.

The proof is given in Appendix C. There is nothing special about subtracting v_1; there exist more natural choices to express translation invariant functions. For instance, one classical way in physics to express translation invariance is to take class representatives of the form $(v_1, .., v_n)$ where $\sum v_i = 0$ (for example, subtracting the center of mass). Proposition 7 characterizes the space of O(d)-equivariant functions that are also translation invariant or translation equivariant. The proof is in Appendix C.

Proposition 7*:* An O(d)-equivariant function $h : (\mathbb{R}^d)^n \to \mathbb{R}^d$ that is translation-invariant can be written as (6) where the f_t are O(d) and translation invariant and $\sum_{t=1}^n f_t(v_1, v_2, .., v_n) = 0$. Similarly, if h is translation-equivariant then we can choose $\sum_{t=1}^n f_t(v_1, v_2, .., v_n) = 1$.

Lorentz symmetry: The Lorentz group acts on Minkowski spacetime as Lorentz transformations, which keep the metric tensor invariant. Lorentz transformations relate space and time between inertial reference frames, which move at a constant relative velocity; spacetime intervals are invariant across frames. The group is made up of spatial rotations in the three space dimensions and linear velocity "boosts" along each dimension. This set of symmetries—required for special relativity—is not O(4), but rather the non-compact group O(1,3) defined in Table 1.

The characterization of invariant and equivariant functions we obtained for the orthogonal group can be extended to the Lorentz group, obtaining a very similar result, summarized in Proposition 8 and proven in Appendix D.

Proposition 8*:* A continuous function $h : (\mathbb{R}^{d+1})^n \to \mathbb{R}^{d+1}$ is Lorentz-equivariant (with respect to the action in Table 2) if and only if

$$h(v_1, .., v_n) = \textstyle\sum_{t=1}^n f_t(v_1, .., v_n)\, v_t \tag{8}$$

where $f_t : (\mathbb{R}^{d+1})^n \to \mathbb{R}$ are Lorentz-invariant scalar functions. Moreover, the functions f_t are uniquely determined by the pairwise Minkowski inner product $\langle \cdot, \cdot \rangle$ (4):

$$f_t(v_1, .., v_n) = g_t(\langle v_i, v_j \rangle_{i,j=1}^n)\ . \tag{9}$$

If h is polynomial, the f_t and corresponding g_t can be taken to be polynomial.

Poincaré symmetry: The Poincaré group combines translation symmetry with Lorentz symmetry. Together these complete the symmetries of special relativity, forming the full group of spacetime transformations that preserve the Minkowski metric. The generalization from a Lorentz-equivariant formulation to a Poincaré-equivariant formulation is similar to the generalization from O(d) to E(d): The *position* vectors take on a special role, in which only *differences* of position can appear as vector inputs to the functions; all other vectors can act unchanged.

Proposition 9 generalizes the results above to the Poincaré group action. As before, we assume that all vectors are position vectors for simplicity. The proof is analogous to the proof of Proposition 7.

Proposition 9*:* A continuous function $h : (\mathbb{R}^{d+1})^n \to \mathbb{R}^{d+1}$ is Poincaré-equivariant (with respect to the action in Table 2) if and only if

$$h(v_1, .., v_n) = \textstyle\sum_{t=1}^n f_t(v_1, .., v_n)\, v_t \tag{10}$$

where the $f_t : (\mathbb{R}^{d+1})^n \to \mathbb{R}$ are translation and Lorentz-invariant scalar functions, determined as in (9) by the pairwise Minkowski inner products (4), but also satisfying $\sum_{t=1}^n f_t(v_1, v_2, .., v_n) = 1$. Furthermore, if h is polynomial, the f_t and corresponding g_t (as in (9)) can be taken to be polynomial.

Permutation invariance and equivariance: Most physics problems are permutation-invariant, in that once you know the masses, sizes, shapes, and so on of the objects in the problem, the physical predictions are invariant to labeling. In particle physics, this invariance is raised to a fundamental symmetry; fundamental particles (like electrons) are identical and exchangeable.

The characterization of permutation-invariant functions with respect to the action in Table 2 is classical [94, pp. 36–39]. Here we prove an extension that describes permutation-invariant vector functions that are also O(d)-equivariant. The proof is in Appendix E.

Proposition 10*:* Let $h : (\mathbb{R}^d)^n \to \mathbb{R}^d$ be O(d)-equivariant (or continuous O($1, d-1$)-equivariant) and also permutation-invariant with respect to the action in Table 2. Then h can be written as

$$h(v_1, .\,., v_n) = \textstyle\sum_{t=1}^n f(v_t, v_1, .\,., v_{t-1}, v_{t+1}, .\,., v_n)\, v_t, \tag{11}$$

where $f : (\mathbb{R}^d)^n \to \mathbb{R}$ is O(d)-invariant (or O($1, d-1$)-invariant) and permutation-invariant with respect to the last $n-1$ inputs.

Proposition 11 proven in Appendix E extends the characterization above to permutation equivariant functions (see also [36, 49]).

Proposition 11*:* Let $h : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$ be O(d)-equivariant (or continuous O($1, d-1$)-equivariant) and also permutation-equivariant with respect to the action in Table 2. Then h can be written as $h = (h_1, .\,., h_n)$ where each $h_i : (\mathbb{R}^d)^n \to \mathbb{R}^d$ is O(d)-equivariant (or continuous O($1, d-1$)-equivariant) and

$$h_i(v_1, .\,., v_n) = \textstyle\sum_{t=1}^n f_t^{(i)}(v_1, .\,., v_n)\, v_t, \tag{12}$$

where all the $f_j^{(i)} : (\mathbb{R}^d)^n \to \mathbb{R}$ are O(d)-invariant (or O($1, d-1$)-invariant), and for all $i, j = 1, .\,., n$ and all $\sigma \in S_n$ we have

$$f^{(i)}_{\sigma^{-1}(j)}(v_{\sigma(1)}, .\,., v_{\sigma(n)}) = f_j^{(\sigma(i))}(v_1, .\,., .\,., v_n). \tag{13}$$

4 Examples

Here we briefly state two classical physics expressions that obey all the symmetries, and show how to formulate them in terms of invariant scalars.

Total mechanical energy: In Newtonian gravity, the total mechanical energy T of n particles with scalar masses m_i, vector positions r_i, and vector velocities v_i is a scalar[1] function:

$$T = \frac{1}{2} \textstyle\sum_{i=1}^n m_i\, |v_i|^2 - \frac{1}{2} \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \frac{G\, m_i\, m_j}{|r_i - r_j|}\ , \tag{14}$$

where G is Newton's constant (a fundamental constant, and hence scalar). Since $|a| \equiv (a^\top a)^{1/2}$, this expression (14) is manifestly constructed from functions only of scalars m_i and scalar products of vectors. It is also worthy of note that the positions r_i only appear in differences of position.

Electromagnetic force law: The total electromagnetic force F acting on a test particle of charge, 3-vector position, and 3-vector velocity (q, r, v) given a set of n other charges (q_i, r_i, v_i) is a vector function [42]:

$$F = \underbrace{\textstyle\sum_{i=1}^n k\, q\, q_i\, \frac{(r - r_i)}{|r - r_i|^3}}_{\text{electrostatic force}} + \underbrace{\textstyle\sum_{i=1}^n k\, q\, q_i\, \frac{v \times (v_i \times (r - r_i))}{c^2\, |r - r_i|^3}}_{\text{magnetic force}}\ , \tag{15}$$

where k is an electromagnetic constant, c is the speed of light, and $a \times b$ represents the cross product that produces a pseudo-vector perpendicular to vectors a and b according to the right-hand rule. This doesn't obviously obey our equivariance requirements, because cross products deliver parity-violating pseudo-vectors; these can't be written in O(3)-equivariant form. However, a cross of a cross of vectors is a vector, so this expression is in fact O(3)-equivariant, as are all forces (because forces must be O(3)-equivariant vectors in order for the theory to be self-consistent).

We can expand the vector triple product using the identity $a \times (b \times c) = (a^\top c)\, b - (a^\top b)\, c$:

$$\begin{aligned} F &= \sum_{i=1}^n k\, q\, q_i\, \frac{(r - r_i)}{|r - r_i|^3} + \sum_{i=1}^n k\, q\, q_i\, \frac{(v^\top (r - r_i))\, v_i - (v^\top v_i)\, (r - r_i)}{c^2\, |r - r_i|^3} \\ &= \sum_{i=1}^n k\, q\, q_i \left(1 - \frac{v^\top v_i}{c^2}\right) \frac{(r - r_i)}{|r - r_i|^3} + \sum_{i=1}^n k\, q\, q_i\, \frac{(v^\top (r - r_i))\, v_i}{c^2\, |r - r_i|^3}\ , \end{aligned} \tag{16}$$

[1] The energy is *not* a scalar in special relativity or general relativity, but it is a scalar in Newtonian physics.

where the quantity $v^\top v_i$ is the scalar product of the velocities. All of the quantities are now straightforwardly functions of invariant scalar products times the input vectors.

5 How many scalars are needed?

Our analysis in Section 3 shows that the invariant and equivariant functions of interest (under actions in Table 2 from groups in Table 1) with input vectors $v_1, .., v_n$, can be expressed in terms of the scalars $\langle v_i, v_j \rangle_{i \geq j=1}^n$ (4). This greatly simplifies the parameterization of such functions, but it significantly increases the number of features if $n \gg d$. In Appendix G we remark that the scalars can be uniquely determined by a small subset of size approximately $(d+1)\,n$. This is related to the rigidity theory of Gram matrices [78] that answers when there exists a unique set of vectors that realize a partial set of distances, and it is closely related to the low rank matrix completion problem [84]. Furthermore, there is a vast literature studying high probability robust reconstruction of all scalars from a random subset via convex optimization techniques [13]. Recently developed optimization techniques on Gram matrices could provide efficient algorithms to learn invariant and equivariant functions [43].

6 Limitations and caveats

Although the principal results presented here work for many groups, and work naturally at all spatial dimensions d (unlike methods based on irreps, for example), they do not solve all problems for all use cases of equivariant machine learning. For one, there are myriad groups—and especially discrete groups—that apply to physical and chemical systems where invariant and equivariant functions do not have such a nice characterization. Such is the case for the GNNs discussed in Appendix A.

One example of a situation in which our formulation might not be practical is provided by multipole expansions (for example, those used in the fast multipole method [5] and n-body networks [50]). In the fast multipole method, a hierarchical spatial graph is constructed, and high-order tensors are used to aggregate information from lower-level nodes into higher-level nodes. This aggregation is concise and linear when it is performed using high-order tensors; this aggregation is hard (or maybe impossible) when only scalars can be transmitted, without the use of the irreps of the relevant symmetry groups. This is a research direction we are currently exploring.

Another example that suggests that our universal functions might be cumbersome is the current forms in which classical theories—such as electromagnetism and general relativity—are written. For example, in Section 4 we showed that the electromagnetic force law can be written in the form of functions of scalars and scalar products times vectors, but that is *not* how the theory is traditionally written. It is traditionally written in terms of the magnetic field (a pseudo-vector) or the electromagnetic tensor (an order-2 tensor). As another example, general relativity is traditionally written in terms of contractions of an order-4 curvature tensor. That is, although the theories can in principle be written in the forms we suggest, they will in general be much more concise or simple or clear in forms that make use of higher-order or non-equivariant forms.

Finally, although our results apply to many physically relevant groups, they do not encode all of the symmetries of classical physics. For example, one critical symmetry is the dimensional or units symmetry: You cannot add or subtract terms that have different units (positions and forces, for example). This symmetry or consideration has implications for the construction of valid polynomials. It also implies that only dimensionless (unitless) scalar quantities can be the arguments of large classes of nonlinear functions, including exponentials or sigmoids. These additional symmetries must be enforced at present with some additional considerations of network architecture or constraints.

We also note that in this work we have characterized *global* symmetries, which act on each point in the same way. Our characterization does not obviously generalize to gauge symmetries, which are local symmetries that apply changes independently to spatially separated points (see e.g. [12]). That said, we believe our model could encompass general gauge symmetries if we replace the local metric by any position-dependent metric Λ_x. Recent work has shown that equivariance under gauge symmetries is possible in the realm of convolutional neural networks by defining coordinate-independent kernels [93]. In our case, we would have to propagate spatially separated vectors to the same location in order to pass them to a locally invariant function, requiring the operation of parallel

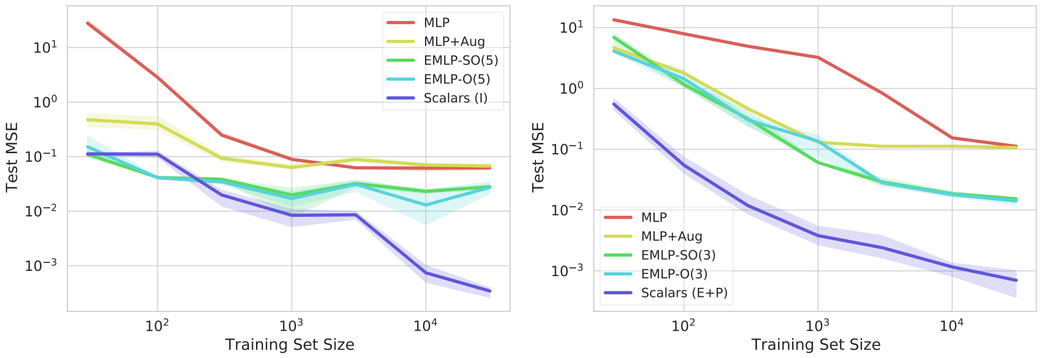

Figure 1: Test error as a function of training set size for (**Left**) the O(5)-invariant task, and (**Right**) the O(3)-equivariant task. Scalars (I) denotes the MLP model using the scalars method for O(5)-invariance construction, and Scalars (E+P) denotes the MLP model using the scalars method for O(3)-equivariance and permutation invariance construction. MLP denotes a standard multilayer perceptron, and MLP+Aug denotes an MLP that has been trained with data augmentation to the given symmetry group. EMLP-G denotes the EMLP models from [29] with different relevant symmetry groups G. For both tasks, the scalar method outperforms all other methods. The shaded regions depict 95% confidence intervals taken over 3 runs.

transport. This operation would then also have to obey our invariance characterization; we believe this is possible and we will detail it in future work.

7 Numerical experiments

We demonstrate our approach using scalar-based multi-layer perceptrons (MLP) on two toy learning tasks from [29]: an O(5)-invariant task and an O(3)-equivariant task. Further numerical experiments with these methods applied to dynamical systems appear in [97]. The code is available on GitHub[2], and it reuses much of the functionality provided by EMLP [29].

O(5)-invariant task: Given observations of the form $(x_1^i, x_2^i, f(x_1^i, x_2^i))_{i=1}^N$, where $x_1^i, x_2^i \in \mathbb{R}^5$ and f is an O(5)-invariant scalar function, we aim to learn f. In this case f is the example from [29]:

$$f(x_1, x_2) = \sin(\|x_1\|) - \frac{\|x_2\|^3}{2} + \frac{x_1^\top x_2}{\|x_1\|\|x_2\|}, \quad x_1, x_2 \in \mathbb{R}^5. \tag{17}$$

We model f using Lemma 1, namely $f(x_1, x_2) = g(x_1^\top x_1, x_1^\top x_2, x_2^\top x_2)$ where $g : (\mathbb{R})^3 \to \mathbb{R}$ could be any function. We learn g by implementing it as an MLP.

O(3)-equivariant task: In this task, the data are $n = 5$ point masses and positions $(m_i, x_i)_{i=1}^5$, and the goal is to predict the matrix $\mathcal{I} = \sum_{i=1}^5 m_i(x_i^\top x_i I - x_i x_i^\top)$. To this end we aim to learn

$$\begin{aligned} h : (\mathbb{R} \times \mathbb{R}^3)^5 &\to \mathbb{R}^{3\times 3} \\ (m_i, x_i)_{i=1}^5 &\mapsto \mathcal{I}. \end{aligned} \tag{18}$$

The function h is O(3)-equivariant in positions, and S_5-invariant, namely, for all $Q \in \mathrm{O}(3)$ and $\sigma \in S_5$ we have $h((m_{\sigma(i)}, Qx_{\sigma(i)})_{i=1}^5) = Qh((m_i, x_i)_{i=1}^5)$. By Proposition 10 we know there exists functions f_0, f_1, and f_2 such that

$$\begin{aligned} h((m_i, x_i)_{i=1}^5) = \sum_{i=1}^5 f_0(x_i^\top x_i, m_i, \{x_k^\top x_l, m_k, m_l\}_{k,l\neq i})\, x_i x_i^\top + \\ + \sum_{i>j=1}^5 f_1(x_i^\top x_j, m_i, m_j, \{x_k^\top x_l, m_k, m_l\}_{k,l\neq i,j})\, x_i x_j^\top + f_2(\{x_i^\top x_j, m_i, m_j\}_{i,j=1}^5) I. \end{aligned}$$

Here the set notation means that the function f_i is permutation-invariant with respect to the inputs in the set. The function h is O(3)-equivariant by construction (see Section 3). We model f_0, f_1 ,and f_2 with MLPs on deep sets [**?**].

[2] `https://github.com/weichiyao/ScalarEMLP`

Acknowledgements: It is a pleasure to thank Tim Carson (Google), Miles Cranmer (Princeton), Johannes Klicpera (TUM), Risi Kondor (Chicago), Lachlan Lancaster (Princeton), Yuri Tschinkel (NYU), Fedor Bogomolov (NYU), Gregor Kemper (TUM), and Rachel Ward (UT Austin) for valuable comments and discussions. We especially thank Gerald Schwartz (Brandeis) for indicating the line of the polynomial argument to us, and Marc Finzi (NYU) for their help with the EMLP codebase. SV was partially supported by NSF DMS 2044349, the NSF–Simons Research Collaboration on the Mathematical and Scientific Foundations of Deep Learning (MoDL) (NSF DMS 2031985), and the TRIPODS Institute for the Foundations of Graph and Deep Learning at Johns Hopkins University. KSF was supported by the Future Investigators in NASA Earth and Space Science and Technology (FINESST) award number 80NSSC20K1545.

References

[1] A. Alex, M. Kalus, A. Huckleberry, and J. von Delft. A numerical algorithm for the explicit calculation of SU(N) and SL($N, \mathbb{C}$) Clebsch-Gordan coefficients. *Journal of Mathematical Physics*, 52(2):023507, 2011.

[2] B. Anderson, T. S. Hy, and R. Kondor. Cormorant: Covariant molecular neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[3] A. K. Aniyan and K. Thorat. Classifying radio galaxies with the convolutional neural network. *The Astrophysical Journal Supplement Series*, 230(2):20, Jun 2017.

[4] W. Azizian and M. Lelarge. Characterizing the expressive power of invariant and equivariant graph neural networks. *arXiv:2006.15646*, 2020.

[5] R. Beatson and L. Greengard. A short course on fast multipole methods. *Wavelets, Multilevel Methods and Elliptic PDEs*, 1:1–37, 1997.

[6] Y. Bengio, A. Lodi, and A. Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 2020.

[7] A. Bietti, L. Venturi, and J. Bruna. On the sample complexity of learning with geometric stability. *arXiv preprint arXiv:2106.07148*, 2021.

[8] B. Blum-Smith and S. Villar. Equivariant maps from invariant functions, 2022.

[9] A. Bogatskiy, B. Anderson, J. Offermann, M. Roussi, D. Miller, and R. Kondor. Lorentz group equivariant neural network for particle physics. In *International Conference on Machine Learning*, pages 992–1002. PMLR, 2020.

[10] G. Bouritsas, F. Frasca, S. Zafeiriou, and M. M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *arXiv:2006.09252*, 2020.

[11] D. Boyda, G. Kanwar, S. Racanière, D. J. Rezende, M. S. Albergo, K. Cranmer, D. C. Hackett, and P. E. Shanahan. Sampling using $SU(N)$ gauge equivariant flows. *Phys. Rev. D*, 103(7):074504, 2021.

[12] M. M. Bronstein, J. Bruna, T. Cohen, and P. Velivckovi'c. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *ArXiv*, abs/2104.13478, 2021.

[13] E. J. Candes and B. Recht. Exact low-rank matrix completion via convex optimization. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 806–812. IEEE, 2008.

[14] Q. Cappart, D. Chételat, E. Khalil, A. Lodi, C. Morris, and P. Veličković. Combinatorial optimization and reasoning with graph neural networks. *arXiv:2102.09544*, 2021.

[15] Z. Chen, L. Chen, S. Villar, and J. Bruna. Can graph neural networks count substructures? In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Neural Information Processing Systems*, volume 33, pages 10383–10395. Curran Associates, Inc., 2020.

[16] Z. Chen, L. Li, and J. Bruna. Supervised community detection with line graph neural networks. *International Conference on Learning Representations*, 2019.

[17] Z. Chen, S. Villar, L. Chen, and J. Bruna. On the equivalence between graph isomorphism testing and function approximation with gnns. In *Neural Information Processing Systems*, pages 15868–15876, 2019.

[18] K. Choo, G. Carleo, N. Regnault, and T. Neupert. Symmetries and many-body excitations with neural-network quantum states. *Physical Review Letters*, 121:167204, Oct 2018.

[19] T. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2019.

[20] M. D. Cranmer, S. Greydanus, S. Hoyer, P. W. Battaglia, D. N. Spergel, and S. Ho. Lagrangian neural networks. *arXiv:2003.04630*, 2020.

[21] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *neural information processing systems*, pages 3844–3852, 2016.

[22] S. Dieleman, K. W. Willett, and J. Dambre. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 450(2):1441–1459, Apr 2015.

[23] H. Domínguez Sánchez, M. Huertas-Company, M. Bernardi, D. Tuccillo, and J. L. Fischer. Improving galaxy morphologies for sdss with deep learning. *Monthly Notices of the Royal Astronomical Society*, 476(3):3661–3676, Feb 2018.

[24] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Neural Information Processing systems*, pages 2224–2232, 2015.

[25] N. Dym and H. Maron. On the universality of rotation equivariant point cloud networks. *arXiv:2010.02449*, 2020.

[26] A. Einstein. Die Grundlage der allgemeinen Relativitätstheorie. *Annalen der Physik*, 354(7):769–822, Jan. 1916.

[27] B. Elesedy and S. Zaidi. Provably strict generalisation benefit for equivariant models. *arXiv preprint arXiv:2102.10333*, 2021.

[28] M. J. Fenton, A. Shmakov, T.-W. Ho, S.-C. Hsu, D. Whiteson, and P. Baldi. Permutationless Many-Jet Event Reconstruction with Symmetry Preserving Attention Networks. *arXiv:2010.09206*, 2020.

[29] M. Finzi, M. Welling, and A. G. Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. *arXiv:2104.09459*, 2021.

[30] F. Fuchs, D. Worrall, V. Fischer, and M. Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Neural Information Processing Systems*, 33, 2020.

[31] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[32] R. González, R. Muñoz, and C. Hernández. Galaxy detection and identification using deep learning and data augmentation. *Astronomy and Computing*, 25:103–109, 2018.

[33] R. Goodman and N. R. Wallach. *Symmetry, representations, and invariants*, volume 255. Springer, 2009.

[34] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

[35] S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[36] B. Gripaios, W. Haddadin, and C. G. Lester. Lorentz-and permutation-invariants of particles. *Journal of Physics A: Mathematical and Theoretical*, 54(15):155201, 2021.

[37] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Transactions on Pattern Analysis and machine intelligence*, 2020.

[38] W. Haddadin. Invariant polynomials and machine learning. *arXiv preprint arXiv:2104.12733*, 2021.

[39] K. Huang. Traffic agent movement prediction using resnet-based model. In *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pages 136–139. IEEE, 2021.

[40] M. Huertas-Company, R. Gravet, G. Cabrera-Vives, P. G. Pérez-González, J. S. Kartaltepe, G. Barro, M. Bernardi, S. Mei, F. Shankar, P. Dimauro, and et al. A catalog of visual-like morphologies in the 5 candels fields using deep learning. *The Astrophysical Journal Supplement Series*, 221(1):8, Oct 2015.

[41] A. Ibort, A. López Yela, and J. Moro. A new algorithm for computing branching rules and Clebsch-Gordan coefficients of unitary representations of compact groups. *Journal of Mathematical Physics*, 58(10):101702, 2017.

[42] J. D. Jackson. *Classical Electrodynamics; 2nd ed.* Wiley, New York, NY, 1975.

[43] A. Jalali, M. Fazel, and L. Xiao. Variational gram functions: Convex analysis and optimization. *SIAM Journal on Optimization*, 27(4):2634–2661, 2017.

[44] C. K. Joshi, T. Laurent, and X. Bresson. An efficient graph convolutional network technique for the travelling salesman problem. *arXiv:1906.01227*, 2019.

[45] N. Karalias and A. Loukas. Erdos goes neural: an unsupervised learning framework for combinatorial optimization on graphs. *arXiv:2006.10643*, 2020.

[46] K. Kashinath, M. Mustafa, A. Albert, J. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, et al. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, 379(2194):20200093, 2021.

[47] N. Keriven and G. Peyré. Universal invariant and equivariant graph neural networks. In *Neural Information Processing Systems*, pages 7092–7101, 2019.

[48] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv:1609.02907*, 2016.

[49] J. Klicpera, F. Becker, and S. Günnemann. Gemnet: Universal directional graph neural networks for molecules. *arXiv preprint arXiv:2106.08903*, 2021.

[50] R. Kondor. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials. *arXiv:1803.01588*, 2018.

[51] R. Kondor, H. T. Son, H. Pan, B. Anderson, and S. Trivedi. Covariant compositional networks for learning graphs. *arXiv:1801.02144*, 2018.

[52] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[53] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[54] J. S. H. Lee, I. Park, I. J. Watson, and S. Yang. Zero-Permutation Jet-Parton Assignment using a Self-Attention Network. *arXiv:2012.03542*, 2020.

[55] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

[56] Z. Li, S. Yang, G. Song, and L. Cai. Hamnet: Conformation-guided molecular representation with hamiltonian neural networks, 2021.

[57] J. Ling, R. Jones, and J. Templeton. Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*, 318, 05 2016.

[58] J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[59] C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Raddick, R. C. Nichol, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey*. *Monthly Notices of the Royal Astronomical Society*, 389(3):1179–1189, 2008.

[60] A. Loukas. What graph neural networks cannot learn: depth vs width. In *International Conference on Learning Representations*, 2019.

[61] D. Luo, G. Carleo, B. K. Clark, and J. Stokes. Gauge equivariant neural networks for quantum lattice gauge theories, 2020.

[62] D. Luo, Z. Chen, K. Hu, Z. Zhao, V. M. Hur, and B. K. Clark. Gauge Invariant Autoregressive Neural Networks for Quantum Lattice Models. *arXiv:2101.07243*, 2021.

[63] M. Lutter, C. Ritter, and J. Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *7th International Conference on Learning Representations (ICLR)*. ICLR, May 2019.

[64] H. Maron, H. Ben-Hamu, N. Shamir, and Y. Lipman. Invariant and equivariant graph networks. In *International Conference on Learning Representations*, 2018.

[65] H. Maron, E. Fetaya, N. Segol, and Y. Lipman. On the universality of invariant networks. In *International Conference on Machine Learning*, pages 4363–4371, 2019.

[66] H. Maron, O. Litany, G. Chechik, and E. Fetaya. On learning sets of symmetric elements. In *International Conference on Machine Learning*, pages 6734–6744. PMLR, 2020.

[67] S. Mei, T. Misiakiewicz, and A. Montanari. Learning with invariances in random features and kernel models. *arXiv preprint arXiv:2102.13219*, 2021.

[68] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. *Association for the Advancement of Artificial Intelligence*, 2019.

[69] R. Murphy, B. Srinivasan, V. Rao, and B. Ribeiro. Relational pooling for graph representations. In *International Conference on Machine Learning*, pages 4663–4673, 2019.

[70] E. Noether. Invariant variation problems. *Transport Theory and Statistical Physics*, 1(3):186–207, 1971.

[71] A. Nowak, S. Villar, A. S. Bandeira, and J. Bruna. A note on learning algorithms for quadratic assignment with graph neural networks. *arXiv:1706.07450*, 2017.

[72] D. L. Pimentel-Alarcón, N. Boston, and R. D. Nowak. A characterization of deterministic sampling patterns for low-rank matrix completion. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):623–636, 2016.

[73] V. L. Popov and E. B. Vinberg. Invariant theory. In *Algebraic geometry IV*, pages 123–278. Springer, 1994.

[74] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[75] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++ deep hierarchical feature learning on point sets in a metric space. In *Neural Information Processing Systems*, pages 5105–5114, 2017.

[76] S. Ravanbakhsh, J. Schneider, and B. Poczos. Equivariance through parameter-sharing. *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[77] M. M. G. Ricci and T. Levi-Civita. Méthodes de calcul différentiel absolu et leurs applications. *Mathematische Annalen*, 54(1):125–201, 1900.

[78] B. Roth. Rigid and flexible frameworks. *The American Mathematical Monthly*, 88(1):6–21, 1981.

[79] C. Roth and A. H. MacDonald. Group Convolutional Neural Networks Improve Quantum State Accuracy. *arXiv:2104.05085*, 2021.

[80] A. Sanchez-Gonzalez, V. Bapst, K. Cranmer, and P. Battaglia. Hamiltonian Graph Networks with ODE Integrators. *arXiv:1909.12790*, 2019.

[81] V. G. Satorras, E. Hoogeboom, and M. Welling. E(n) equivariant graph neural networks. *arXiv:2102.09844*, 2021.

[82] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[83] K. T. Schütt, O. T. Unke, and M. Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra, 2021.

[84] A. Singer and M. Cucuringu. Uniqueness of low-rank matrix completion by rigidity theory. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1621–1641, 2010.

[85] B. Tahmasebi and S. Jegelka. Counting substructures with higher-order graph neural networks: Possibility and impossibility results. *arXiv:2012.03174*, 2020.

[86] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv:1802.08219*, 2018.

[87] L. N. Trefethen and D. Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.

[88] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[89] T. Vieijra, C. Casert, J. Nys, W. De Neve, J. Haegeman, J. Ryckebusch, and F. Verstraete. Restricted boltzmann machines for quantum states with non-abelian or anyonic symmetries. *Physical Review Letters*, 124(9), Mar 2020.

[90] R. Wang, K. Kashinath, M. Mustafa, A. Albert, and R. Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1457–1466, 2020.

[91] R. Wang, R. Walters, and R. Yu. Incorporating symmetry into deep dynamics models for improved generalization. In *International Conference on Learning Representations*, 2021.

[92] Z. Wang, C. Wang, S. Zhao, S. Du, Y. Xu, B.-L. Gu, and W. Duan. Symmetry-adapted graph neural networks for constructing molecular dynamics force fields, 2021.

[93] M. Weiler, P. Forré, E. Verlinde, and M. Welling. Coordinate independent convolutional networks – isometry and gauge equivariant convolutions on riemannian manifolds, 2021.

[94] H. Weyl. *The Classical Groups*. Princeton University Press, 1946.

[95] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2018.

[96] W. Yao, A. S. Bandeira, and S. Villar. Experimental performance of graph neural networks on random instances of max-cut. In *Wavelets and Sparsity XVIII*, volume 11138, page 111380S. International Society for Optics and Photonics, 2019.

[97] W. Yao, K. Storey-Fisher, D. W. Hogg, and S. Villar. A simple equivariant machine learning method for dynamics based on scalars, 2021.

[98] D. Yarotsky. Universal approximations of invariant maps by neural networks. *arXiv:1804.10306*, 2018.

[99] R. Yu, P. Perdikaris, and A. Karpatne. Physics-guided ai for large-scale spatiotemporal data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 4088–4089, New York, NY, USA, 2021. Association for Computing Machinery.

[100] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov, and A. J. Smola. Deep sets. In *Neural Information Processing Systems*, pages 3394–3404, 2017.

[101] Z. Zhang, B.-S. Hua, D. W. Rosen, and S.-K. Yeung. Rotation invariant convolutions for 3d point clouds deep learning. In *2019 International Conference on 3D Vision (3DV)*, pages 204–213. IEEE, 2019.

[102] Y. Zhao, T. Birdal, J. E. Lenssen, E. Menegatti, L. Guibas, and F. Tombari. Quaternion equivariant capsule networks for 3d point clouds. In *European Conference on Computer Vision*, pages 1–19. Springer, 2020.

[103] Y. D. Zhong, B. Dey, and A. Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. In *7th International Conference on Learning Representations (ICLR)*. ICLR, May 2020.

A Symmetry-enforcing universal neural networks and irreducible representations

In this Section we describe the general principles to parameterize invariant and equivariant universally approximating functions. As mentioned in Section 2, the most common approach is is to write the invariant/equivariant functions as the composition of linear invariant/equivariant layers and non-linear compatible pointwise activation functions. We start by describing the G-invariant graph networks from [65], which are invariant with respect to a group G (typically a subgroup of the symmetric group S_n, since the work is in the context of graph neural networks) acting on $\mathbb{R}^n$ as $g \star x$ ($g \in G$, $x \in \mathbb{R}^n$). They extend the action $\star$ to tensors $t \in \mathbb{R}^{n^k}$ by acting in each of the k dimensions of the tensor.

Consider a graph $X = (V, E)$ on n nodes. Let $V = (v_1, .., v_n) \in (\mathbb{R}^d)^n$, where $v_i \in \mathbb{R}^d$ are the node features, and assume for simplicity that the edge features are real numbers represented by the matrix $E \in \mathbb{R}^{n \times n}$, then a graph neural network learns equivariant functions $f : X \to (\mathbb{R}^\ell)^n$, where the group of permutations S_n acts in $(\mathbb{R}^d)^n$ and $(\mathbb{R}^\ell)^n$ as in Table 2, and it acts on the graph X as:

$$\Pi \star (V, E) = (\Pi \star V, \Pi \star E) \quad \text{where } \Pi \star E = \Pi E \Pi^\top . \tag{19}$$

The GNN learns an embedding of the form:

$$\mathcal{N}(X) = \theta \circ L_T \circ \ldots \circ L_2 \circ \theta \circ L_1(E) \, , \tag{20}$$

where θ is a point-wise non-linearity and $L_i : \mathbb{R}^{n^{\otimes k_i}} \to \mathbb{R}^{n^{\otimes k_{i+1}}}$ is a linear *equivariant* map. When G is the group of permutations, the network (20) can universally approximate all continuous equivariant functions as long as the order of the intermediate tensors can be arbitrarily large [65, 47]. The key insight is that the k_i-tensors in the intermediate layers can express all (equivariant) polynomial functions of the input of degree k_i. Universality follows from a generalization of the Stone-Weierstrass theorem that states that every continuous equivariant function defined on a compact set can be uniformly approximated as closely as desired by equivariant polynomial functions (for the specific action by permutations) [47, 4]. The disavantage of this approach is that the space of linear equivariant functions L_i, even though it is fully characterized, has dimension that grows super-exponentially with the order of the tensors [64].

The approach for general groups is very similar. Group-equivariant neural networks are written as the composition of linear equivariant functions going to higher order tensors, composed with non-linear pointwise activation functions (for general groups there may be restrictions on the activation functions so that equivariance is preserved). Recent work proposes to enforce equivariance of these linear maps by imposing constraints [29]. But in general, the most prevalent approach is to express intermediate layers as linear maps between (group) representations of G.

Let G be a group acting on $\mathbb{R}^d$ as $\star$. In representation-theory language, a representation of G is a map $\rho : G \to \mathrm{GL}(V)$ that satisfies $\rho(g_1 g_2) = \rho(g_1)\rho(g_2)$ (where V is a vector space and $\mathrm{GL}(V)$ denotes the automorphisms of V, that is, invertible linear maps $V \to V$). The group action $\star$ of G on $\mathbb{R}^d$ is equivalent to the group representation $\rho : G \to \mathrm{GL}(\mathbb{R}^d)$ so that $\rho(g)(v) = g \star v$. We can extend the action $\star$ to the tensor product $(\mathbb{R}^d)^{\otimes k}$ so that the group acts independently in every tensor factor (i.e., in every dimension), namely $\rho_k = \otimes_{r=1}^k \rho : G \to \mathrm{GL}((\mathbb{R}^d)^{\otimes k})$.

The first step is to note that a linear equivariant map $L_i : (\mathbb{R}^d)^{\otimes k_i} \to (\mathbb{R}^d)^{\otimes k_{i+1}}$ corresponds to a map between group representations such that $L_i \circ \rho_{k_i}(g) = \rho_{k_{i+1}}(g) \circ L_i$ for all $g \in G$. Homomorphisms between group representations are easily parametrizable if we decompose the representations in terms of irreps:

$$\rho_{k_i} = \bigoplus_{\ell=1}^{T_{k_i}} \mathcal{T}_\ell \, . \tag{21}$$

In particular, Schur's Lemma implies that a map between two irreps over $\mathbb{C}$ is either zero or a multiple of the identity.

The equivariant neural-network approach consists in decomposing the group representations in terms of irreps and explicitly parameterizing the maps [50, 86, 30]. In general it is not clear how to decompose an arbitrary group representation into irreps. However in the case where $G = \mathrm{SO}(3)$, the decomposition of a tensor representation as a sum of irreps is given by the Clebsh-Gordan decomposition:

$$\otimes_{s=1}^{k} \rho_s = \oplus_{\ell=1}^{T} \mathcal{T}_\ell \tag{22}$$

The Clebsh-Gordan decomposition not only gives the decomposition of the RHS of (22) but also it gives the explicit change of coordinates. This decomposition is fundamental for implementing the equivariant 3D point-cloud methods defined in [30, 86, 9]. Moreover, very recent work [25] shows that the classes of functions defined in [30, 86] are universal, meaning that every continous SO(3)-equivariant function can be approximated uniformly in compacts sets by those neural networks. However, there exists a clear limitation to this approach: Even though decompositions into irreps are broadly studied in mathematics (a.k.a. plethysm), the explicit transformation that allows us to write the decomposition of tensor representations into irreps is a hard problem in general. It is called the *Clebsch-Gordan problem*. There is exciting, recent progress on this problem for large classes of groups [1, 41]. The approach taken in the present work sidesteps this problem altogether.

B Equivariant functions under rotations and the orthogonal group

Generalized cross-product: The generalized cross product of $d-1$ vectors in $\mathbb{R}^d$ is defined to be the Hodge dual of the exterior product of the $d-1$ vectors. Namely, given $v_1, .., v_{d-1}$ the cross product $v_1 \times .. \times v_{d-1}$ is the unique vector that satisfies that for all $y \in \mathbb{R}^d$

$$\langle v_1 \times .. \times v_{d-1}, y\rangle = \det(v_1, .., v_{d-1}, y), \tag{23}$$

where $\langle \cdot, \cdot \rangle$ denotes the usual inner product in $\mathbb{R}^d$, and $\det(v_1, .., v_d)$ corresponds to the determinant of the matrix with rows $v_1, .., v_d$.

Proof of Proposition 4: The purely set-theoretic statement in Proposition 4 and the polynomial statement have independent proofs.

For the set-theoretic statement, let $h : (\mathbb{R}^d)^n \to \mathbb{R}^d$ be an arbitrary O(d)-equivariant vector function. Lemma 3 shows that for any fixed n-tuple of vectors $(v_1, .., v_n)$, there exists an n-tuple of real numbers $(a_1, .., a_n)$ such that

$$h(v_1, \ldots, v_n) = \sum_{t=1}^{n} a_t v_t. \tag{24}$$

Pick one representative tuple $(v_1, \ldots, v_n)$ from each O(d)-orbit on $(\mathbb{R}^d)^n$, and find a corresponding tuple $(a_1, \ldots, a_n)$ satisfying this equation. (Note that the tuple $(a_1, \ldots, a_n)$ thus depends on the orbit; however, this dependence is suppressed in the notation to prevent it from becoming cumbersome.) Define O(d)-invariant functions $f_t : (\mathbb{R}^d)^n \to \mathbb{R}$ by defining $f_t(v_1, \ldots, v_n) = a_t$ for the a_t corresponding to the chosen representative of $(v_1, \ldots, v_n)$'s orbit. Then

$$h(v_1, \ldots, v_n) = \sum f_t(v_1, \ldots, v_n) v_t \tag{25}$$

at the chosen orbit representatives, and the O(d)-equivariance of both sides then implies this equation is satisfied everywhere.

For the polynomial statement, now assume $h : (\mathbb{R}^d)^n \to \mathbb{R}^d$ is a *polynomial* O(d)-equivariant map. Define a new map $\overline{h} : (\mathbb{R}^d)^{n+1} \to \mathbb{R}$ by

$$\overline{h}(v_1, \ldots, v_n, y) = \langle h(v_1, \ldots, v_n), y\rangle, \tag{26}$$

where $\langle \cdot, \cdot \rangle$ is the usual inner product on $\mathbb{R}^d$. Then for any $Q \in$ O(d), we have

$$\begin{aligned} \overline{h}(Q\, v_1, \ldots, Q\, v_n, Q\, y) &= \langle Q\, h(v_1, \ldots, v_n), Q\, y\rangle \\ &= \langle h(v_1, \ldots, v_n), y\rangle \\ &= \overline{h}(v_1, \ldots, v_n, y), \end{aligned}$$

where the first equality is by the definition of $\overline{h}$ and the O(d)-equivariance of h and the second is the fact that the inner product is preserved by O(d). In other words, $\overline{h}$ is an O(d)-invariant scalar function with respect to O(d)'s natural action on $(\mathbb{R}^d)^{n+1}$.

It follows from the First Fundamental Theorem for O(d) that $\overline{h}$ is a polynomial in the inner products $\langle v_i, v_j\rangle$, $\langle v_t, y\rangle$, and $\langle y, y\rangle$. On the other hand, by its definition, it is homogeneous of degree 1 in the coordinates of y. It follows that $\langle y, y\rangle$ does not appear this polynomial expression for $\overline{h}$ in terms of the dot products, and furthermore, each term contains some $\langle v_t, y\rangle$ with degree 1, and is otherwise composed of $\langle v_i, v_j\rangle$'s. Grouping the terms according to which $\langle v_t, y\rangle$ each contains, we get

$$\overline{h}(v_1, \ldots, v_n, y) = \sum_{t=1}^{n} g_t(\langle v_i, v_j\rangle_{i,j=1}^{n})\langle v_t, y\rangle \tag{27}$$

for all $v_1, \ldots, v_n, y$. Defining $f_t(v_1, \ldots, v_n) = g_t(\langle v_i, v_j\rangle_{i,j=1}^{n})$ for each t, we find the f_t's are invariant polynomials. Unspooling the definition of $\overline{h}$ on the left side, and using the linearity of the dot product on the right, this becomes

$$\langle h(v_1, \ldots, v_n), y\rangle = \langle \left(\sum_{t=1}^{n} f_t(v_1, \ldots, v_n)v_t\right), y\rangle. \tag{28}$$

As this equation holds for all y, and dot product is a nondegenerate bilinear form, we can conclude that

$$h(v_1, \ldots, v_n) = \sum_{t=1}^{n} f_t(v_1, \ldots, v_n)v_t, \tag{29}$$

with the f_t invariant polynomials, as promised. □

Proof of Proposition 5: The proof of this proposition runs parallel to the proof of Proposition 4. For the set-theoretic part of the proposition, we need a substitute for Lemma 3 that applies to SO(d). The needed statement is that if $h : (\mathbb{R}^d)^n \to \mathbb{R}^d$ is SO(d)-equivariant, and the span of $v_1, \ldots, v_n$ has dimension different from $d-1$, then $h(v_1, \ldots, v_n)$ lies in $\operatorname{span}(v_1, \ldots, v_n)$. We see this as follows:

Let $W = \operatorname{span}(v_1, \ldots, v_n)$, and suppose this has dimension $d - m$. The pointwise stabilizer of W in SO(d), call it G_W, acts in the orthogonal complement $W^\perp$ of W. Isomorphically, G_W is SO(m), and its action on $W^\perp$ is SO(m)'s canonical action on $\mathbb{R}^m$. If $m \neq 1$, this action is irreducible; in particular, there are no nonzero vectors in $W^\perp$ that are fixed by the whole action, and it follows that there are no vectors in $\mathbb{R}^d$ lying outside of W that are fixed by G_W. On the other hand, because G_W fixes each of $v_1, \ldots, v_n$, equivariance of h implies $h(v_1, \ldots, v_n)$ is fixed by G_W as well. It follows that $h(v_1, \ldots, v_n) \in W$.

From this lemma it follows that for any tuple $(v_1, \ldots, v_n)$ with $\dim \operatorname{span}(v_1, \ldots, v_n) \neq d-1$, there exists a solution in $(a_1, \ldots, a_n) \in \mathbb{R}^n$ to the equation

$$h(v_1, \ldots, v_n) = \sum a_t v_t. \tag{30}$$

On the other hand, when $\dim \operatorname{span}(v_1, \ldots, v_n) = d - 1$, then there must exist $d - 1$ linearly independent vectors $v_{i_1}, \ldots, v_{i_{d-1}}$. In this situation, the generalized cross product $v_{i_1} \times \cdots \times v_{i_{d-1}}$ is nonzero, and linearly independent from $v_{i_1}, \ldots, v_{i_{d-1}}$ (in fact it lies in the orthogonal complement of $\operatorname{span}(v_1, \ldots, v_n)$). Thus it and $v_1, \ldots, v_n$ span $\mathbb{R}^d$. So in all cases, i.e., for any tuple $(v_1, \ldots, v_n)$, there exists a solution in the a_t and a_S to

$$h(v_1, \ldots, v_n) = \sum_{t=1}^{n} a_t v_t + \sum_{S \in \binom{[n]}{d-1}} a_S v_S, \tag{31}$$

where the notation is as in the statement of Proposition 5.

The proof of the set-theoretic statement now proceeds exactly as for the proof of Proposition 4: from each SO(d)-orbit in $(\mathbb{R}^d)^n$, choose a representative tuple $(v_1, \ldots, v_n)$; pick values of a_t and a_S that satisfy the above for this tuple; use them to define invariant functions $f_t(v_1, \ldots, v_n)$ and $f_S(v_1, \ldots, v_n)$; then the equation

$$h(v_1, \ldots, v_n) = \sum_{t=1}^{n} f_t(v_1, \ldots, v_n)v_t + \sum_{S \in \binom{[n]}{d-1}} f_S(v_1, \ldots, v_n)v_S \tag{32}$$

holds at the chosen representative tuples, and the SO(d)-equivariance of both sides shows it holds everywhere.

The proof of the polynomial statement is even more directly parallel to the proof of Proposition 4. As in that proof, given an SO(d)-equivariant polynomial vector function $h : (\mathbb{R}^d)^n \to \mathbb{R}^d$, define a polynomial scalar function

$$\overline{h}(v_1, \ldots, v_n, y) = \langle h(v_1, \ldots, v_n), y\rangle. \tag{33}$$

For the same reason as before, it is SO(d)-invariant. It follows from the First Fundamental Theorem for SO(d) that $\overline{h}$ is a polynomial in the dot products $\langle v_i, v_j\rangle$, $\langle v_t, y\rangle$, $\langle y, y\rangle$, and the $d \times d$ subdeterminants $\det(v_{i_1} \ldots v_{i_d})$ and $\det(v_{i_1} \ldots v_{i_{d-1}}\ y)$. Because it is also homogeneous of degree 1 in the coordinates of y, $\langle y, y\rangle$ cannot occur, and every term must contain either a $\langle v_t, y\rangle$ or a $\det(v_{i_1} \ldots v_{i_{d-1}}\ y)$ exactly once, and is otherwise a product of $\langle v_i, v_j\rangle$'s and $\det(v_{i_1} \ldots v_{i_d})$'s (which is to say, it is otherwise an SO(d)-invariant function of the v_j's). Grouping according to which $\langle v_t, y\rangle$ or a $\det(v_{i_1} \ldots v_{i_{d-1}}\ y)$ each term contains, we get

$$\overline{h}(v_1, \ldots, v_n) = \sum_{t=1}^{n} f_t(v_1, \ldots, v_n)\langle v_t, y\rangle + \sum_{S\in\binom{[n]}{d-1}} f_S(v_1, \ldots, v_n) \det(\tilde{v}_S, y), \tag{34}$$

where the f_t's and f_S's are invariant, and where we have abbreviated $\det(v_{i_1}, \ldots, v_{i_{d-1}}, y)$ as $\det(\tilde{v}_S, y)$ with $S = \{i_1, i_2, \ldots, i_{d-1}\}$. (The tilde is to distinguish it from $v_S = v_{i_1} \times \cdots \times v_{i_{d-1}}$.) Now,

$$\det(\tilde{v}_S, y) = \langle v_S, y\rangle, \tag{35}$$

by definition of the generalized cross product v_S, so by applying the linearity of the dot product on the right, and the definition of $\overline{h}$ on the left, we get

$$\langle h(v_1, \ldots, v_n), y\rangle = \langle \sum_{t=1}^{n} f_t(v_1, \ldots, v_n)v_t + \sum_{S\in\binom{[n]}{d-1}} f_S(v_1, \ldots, v_n)v_S, y\rangle. \tag{36}$$

As with O(d), we get the equality claimed in Proposition 5 because y is arbitrary and $\langle\cdot,\cdot\rangle$ is a nondegenerate bilinear form. □

C Translation-invariant functions

Proof of Lemma 6: Consider the map $\Pi : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^{n-1}$ given by $(v_1, v_2, .., v_n) \mapsto (v_2 - v_1, .., v_n - v_1)$. This map has a section $i : (\mathbb{R}^d)^{n-1} \hookrightarrow (\mathbb{R}^d)^n$ given by $(v_2, \ldots, v_n) \mapsto (0, v_2, \ldots, v_n)$, i.e., $\Pi \circ i$ is the identity. The fibers of Π are exactly the orbits of the translation action. Thus a translation-invariant function f on $(\mathbb{R}^d)^n$ descends via Π to a well-defined function $\tilde{f}$ on $(\mathbb{R}^d)^{n-1}$, so that

$$f = \tilde{f} \circ \Pi. \tag{37}$$

Then

$$\tilde{f} = \tilde{f} \circ \Pi \circ i = f \circ i. \tag{38}$$

Because Π, i are polynomial, these equations show that if either f or $\tilde{f}$ is polynomial then so is the other. Because i, Π are both equivariant for the action of $GL(n, \mathbb{R})$, any equivariance property of either f or $\tilde{f}$ with respect to any subgroup $G \subset GL(n, \mathbb{R})$ is passed to the other. □

Proof of Proposition 7: Let $h : (\mathbb{R}^d)^n \to \mathbb{R}^d$ be translation-invariant and O(d)-equivariant. Using Lemma 6 we can consider an O(d)-equivariant function $\tilde{h} : (\mathbb{R}^d)^{n-1} \to \mathbb{R}^d$ so that $h(v_1, v_2, .., v_n) = \tilde{h}(v_2 - v_1, .., v_n - v_1)$. Therefore there exists O(d) invariant functions $\tilde{f}_t$

$$\tilde{h}(v_2 - v_1, .., v_n - v_1) = \textstyle\sum_{t=2}^{n} \tilde{f}_t(v_2 - v_1, .., v_n - v_1)(v_t - v_1). \tag{39}$$

This implies h can be written as

$$h(v_1, v_2, .., v_n) = \textstyle\sum_{t=2}^{n} f_t(v_1, .., v_n)\, v_t - (\sum_{t=2}^{n} f_t(v_1, .., v_n))\, v_1 \tag{40}$$

where the functions f_t are O(d)- and translation-invariant, which has the claimed form.

For the case where h is O(d)- and translation-equivariant we first observe that it suffices to take one representative per orbit, define the function on that representative, and extend it everywhere else by translations:

$$h(v_1, .., v_n) = \tilde{h}(v_2 - v_1, .., v_n - v_1) + v_1, \tag{41}$$

where $\tilde{h}$ is O(d)-equivariant. Therefore any function h that is O(d)- and translation-equivariant can be written as

$$h(v_1, .., v_n) = \textstyle\sum_{t=2}^n \tilde{f}_t(v_2 - v_1, .., v_n - v_1)(v_t - v_1) + v_1 \tag{42}$$
$$= \textstyle\sum_{t=2}^n f_t(v_1, v_2 .., v_n)\, v_t + \left(1 - \sum_{t=2}^n f_t(v_1, v_2 .., v_n)\right) v_1 . \tag{43}$$

□

D Invariant and equivariant functions under the Lorentz group

Proof of Proposition 8: The proof follows the pattern of the proof of Proposition 4. The claim for polynomial h follows in exactly the same manner, because there is a First Fundamental Theorem for the Lorentz group precisely analogous to Lemma 1—in fact, both are consequences of the First Fundamental Theorem for O(d,ℂ) [33, Proposition 5.2.2], because over ℂ, the Lorentz group and the orthogonal group are related by a change of basis (namely, multiplying the space coordinates by i).

The claim for arbitrary continuous h has two added subtleties for the Lorentz group. First, unlike for O(d), the Minkowski inner products of the vectors v_j do not distinguish every pair of distinct O(1,$d-$1) orbits from each other: for example the orbit of $(v, \ldots, v)$, with $\langle v, v\rangle = 0$, is indistinguishable from 0 by the inner products, even if $v \neq 0$. So there exist O($1, d-1$)-invariant set functions on $(\mathbb{R}^d)^n$ that are not functions of the Minkowski inner products. However, they do distinguish every pair of *closed* orbits, and every orbit has a closed orbit in its closure (this is a general theorem about the invariant ring of a reductive group, see [73, Theorems 4.6 and 4.7 and their corollaries]; the Lorentz group is reductive because it is a real form of the reductive group O($d, \mathbb{C}$)). Thus every *continuous* invariant function is indeed a function of the Minkowski inner products.

The second subtlety is in proving that a continuous, equivariant vector function h always lies in the span of $v_1, \ldots, v_n$. Our approach in Section 3 to show that the invariant functions under the orthogonal group are restricted to the span of the input vectors is based on the following idea: Given $v_1, .. v_n$, if they span $\mathbb{R}^d$, then the result trivially holds. Otherwise let $\{w_1, .. w_m\}$ be a basis of span$(v_1, .., v_n)$ and extend it to a basis of $\mathbb{R}^d$. Then we do a full orthogonalization (via Gram Schmidt) to get orthogonal vectors $u_1, u_2, .., u_d$ and then construct Q:

$$u_1 \leftarrow w_1 \tag{44}$$

$$\text{then for each } j\ (2 \leq j \leq d) \text{ in order: } \ u_j \leftarrow w_j - \sum_{k=1}^{j-1} \frac{w_j^\top u_k}{u_k^\top u_k}\, u_k \tag{45}$$

$$Q \leftarrow \left[\sum_{j=1}^{m} \frac{1}{u_j^\top u_j}\, u_j\, u_j^\top\right] - \left[\sum_{j=m+1}^{d} \frac{1}{u_j^\top u_j}\, u_j\, u_j^\top\right] . \tag{46}$$

Then Q is an element of O(d) that fixes everything in span$(v_1, .., v_n)$ and does not fix anything outside it. This can be used to show that equivariant functions are restricted to the span of the inputs (see Lemma 3).

This idea can be generalized to the Lorentz group O($1, d$). Let $(t, x_1, .., x_d) \in \mathbb{R}^{d+1}$, the metric is

$$\Lambda = \hat{e}_t\, \hat{e}_t^\top - \sum_{j=1}^{d} \hat{e}_{x_j}\, \hat{e}_{x_j}^\top , \tag{47}$$

where $\hat{e}_t$ is a timelike unit vector and the $\hat{e}_{x_j}$ are orthonormal space-like vectors (all orthogonal to $\hat{e}_t$). Given $\{v_1, .. v_n\}$ we consider $\{w_1, .. w_m\}$ a basis of span$(v_1, .., v_n)$ and we extend it to $\{w_1, .. w_{d+1}\}$ a basis of $\mathbb{R}^{d+1}$ and orthogonalize all the vectors according to the Lorentz generalization of orthogonalization given above to make orthogonal vectors $u_1, u_2, .., u_{d+1}$ and then construct Q:

$$u_1 \leftarrow w_1 \tag{48}$$

$$\text{then for each } j\ (2 \leq j \leq d+1) \text{ in order: } \ u_j \leftarrow w_j - \sum_{k=1}^{j-1} \frac{\langle w_j, u_k\rangle}{\langle u_k, u_k\rangle}\, u_k \tag{49}$$

$$Q \leftarrow \left[\sum_{j=1}^{m} \frac{1}{\langle u_j, u_j\rangle}\, u_j\, u_j^\top \Lambda\right] - \left[\sum_{j=m+1}^{d+1} \frac{1}{\langle u_d, u_d\rangle}\, u_d\, u_d^\top \Lambda\right] \ . \tag{50}$$

Note that (49) and (50) require $\langle u_j, u_j\rangle \neq 0$ for all j, however, the Minkowski inner product (4) is not positive definite, in particular there can be *lightlike* vectors $u_j \neq 0$ in which $\langle u_j, u_j\rangle = 0$. In our argument we first will require that if $m = 1$ then v_1 is not lightlike. If the procedure (49) generates a lightlike u_j for any j in the range $1 \leq j \leq d+1$ we proceed in the following way:

- If $m > 1$ and $j \leq m$, replace the vectors $w_1, .., w_n$ by a linear combination of them with random coefficients and start again. The fact that the ligthlike vectors are a codimension 1 conic surface ensures that every linear subspace of dimension greater than or equal to 2 has an orthogonal basis where none of the basis elements are lightlike.
- If $j > m$ and $d \geq m+1$ then the complement of the span of $w_1, .., w_m$ has dimension at least 2 and the same procedure can be applied: one can remove lightlike vectors without changing the subspace by replacing $w_{m+1}, .., w_{d+1}$ with a linear combination of them.
- If $m = d$ the procedure will not produce a lightlike vector as its last vector. This is because an orthogonal basis cannot have a lighlike vector. Each lightlike vector is orthogonal to d linearly independent vectors but one of those is itself. So if u_{d+1} is a lightlike vector w_{d+1} is in the span of $w_1, .. w_d$ which contradicts our assumption that $\{w_1, .., w_{d+1}\}$ is linearly independent.

This shows that given $\{v_1, .., v_n\}$ set of vectors spanning a linear space of dimension $m > 1$, there exists a $Q \in$O$(1, d)$ such that $Q(v) = v$ for all $v \in$ span$(v_1, .., v_n)$ and $Q(w) \neq w$ for all $w \notin$ span$(v_1, .., v_n)$, which can be used to show that $h(v_1, .., v_n) \in$ span$(v_1, .., v_n)$ as in Lemma 3.

If $m = 1$ and v_1 is lightlike, the continuity of h and the fact that the set of non-lightlike vectors are dense imply that $h(v_1) \in$ span(v_1). Without the continuity of h this argument wouldn't work. This is because the set of lightlike vectors is invariant under the action of the Lorentz group. □

E Permutation-invariant and equivariant functions that are also orthogonal or Lorentz-equivariant

Proof of Proposition 10: Let $\Pi_{j\to i} = \{\sigma \in S_n : \sigma(j) = i\}$. We start by writing h as in (3). Using the invariance we average over the orbit of S_n we obtain:

$$\begin{aligned} h(v_1, .., v_n) &= \sum_{j=1}^{n} f_j(v_1, .., v_n) v_j && (51)\\ &= \frac{1}{n!}\sum_{j=1}^{n}\sum_{\sigma\in S_n} f_j(v_{\sigma(1)}, .., v_{\sigma(n)}) v_{\sigma(j)} && (52)\\ &= \frac{1}{n!}\sum_{j=1}^{n}\left(\sum_{i=1}^{n}\sum_{\sigma\in\Pi_{j\to i}} f_j(v_{\sigma(1)}, .., v_{\sigma(n)}) v_i\right) && (53)\\ &= \sum_{i=1}^{n}\frac{1}{n!}\left(\sum_{j=1}^{n}\sum_{\sigma\in\Pi_{j\to i}} f_j(v_{\sigma(1)}, .., v_{\sigma(n)})\right) v_i && (54)\end{aligned}$$

Let $\tilde{f}_i(v_i, v_1, .., v_{i-1}, v_{i+1}, .., v_n) = \frac{1}{n!}\left(\sum_{j=1}^{n}\sum_{\sigma\in\Pi_{j\to i}} f_j(v_{\sigma(1)}, .., v_{\sigma(n)})\right)$. For fixed j, the set of permutations $\Pi_{j\to i}$ is stable (as a set) under post-composition with any permutation τ that fixes i. As a consequence, each summand $\sum_{\sigma\in\Pi_{j\to i}} f_j(v_{\sigma(1)}, .., v_{\sigma(n)})$ is invariant under the action of any such permutation τ. As a consequence, $\tilde{f}_i$ is invariant with respect to the last $n-1$ inputs, and h can be expressed as

$$h(v_1, .., v_n) = \sum_{i=1}^{n} \tilde{f}_i(v_i, v_{[-i]})\, v_i, \tag{55}$$

using the notation $v_{[-t]} := (v_1, .., v_{t-1}, v_{t+1}, .., v_n)$. We now show that all $\tilde{f}_i$'s can be chosen to the same functions. In order to do so average over permutations again:

$$h(v_1, .., v_n) = \frac{1}{n!} \sum_{i=1}^{n} \sum_{\sigma \in S_n} \tilde{f}_i(v_{\sigma(i)}, v_{[-\sigma(i)]}) v_{\sigma(i)}, \tag{56}$$

$$= \frac{1}{n!} \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \sum_{\sigma \in \Pi_{i \to j}} \tilde{f}_i(v_j, v_{[-j]}) \right) v_j, \tag{57}$$

$$= \frac{1}{n!} \sum_{j=1}^{n} \left(\sum_{i=1}^{n} (n-1)!\, \tilde{f}_i(v_j, v_{[-j]}) \right) v_j, \tag{58}$$

$$= \sum_{j=1}^{n} \left(\frac{1}{n} \sum_{i=1}^{n} \tilde{f}_i(v_j, v_{[-j]}) \right) v_j, \tag{59}$$

Therefore we define $\hat{f}$ such as

$$\hat{f}(w, w_1, .., w_{n-1}) = \frac{1}{n} \sum_{i=1}^{n} \tilde{f}_i(w, w_1, .., w_{n-1}), \tag{60}$$

a $O(d)$-invariant function that is permutation invariant with respect to the last $n-1$ inputs. The computation in (59) shows that:

$$h(v_1, .., v_n) = \sum_{j=1}^{n} \hat{f}(v_j, v_{[-j]}) v_j, \tag{61}$$

which proves the theorem. □

Proof of Proposition 11: Let $\sigma \in S_n$ and $(h_1, .., h_n) = h : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$ a permutation-equivariant function. Then

$$h(\sigma \star (v_1, .., v_n)) = \sigma \star h(v_1, .., v_n) \tag{62}$$

$$h(v_{\sigma(1)}, .., v_{\sigma(n)}) = (h_{\sigma(1)}(v_1, .., v_n), .., h_{\sigma(n)}(v_1, .., v_n)) \tag{63}$$

$$(h_1(v_{\sigma(1)}, .., v_{\sigma(n)}), .., h_n(v_{\sigma(1)}, .., v_{\sigma(n)})) = (h_{\sigma(1)}(v_1, .., v_n), .., h_{\sigma(n)}(v_1, .., v_n)) \tag{64}$$

Since $h_i : (\mathbb{R}^d)^n \to \mathbb{R}^d$ are O(d)-equivariant (or continuous O$(1, d-1)$-equivariant) we have that for all i there exists $f_t^i : (\mathbb{R}^d)^n \to \mathbb{R}$ O(d)-invariant (or O$(1, d-1)$-invariant) such that:

$$h_i(v_1, .., v_n) = \sum_{t=1}^{n} f_t^{(i)}(v_1, .., v_n) v_t. \tag{65}$$

Combining (64) and (65) we get

$$h_i(v_\sigma(1), .., v_\sigma(n)) = \sum_{t=1}^{n} f_t^{(i)}(v_{\sigma(1)}, .., v_{\sigma(n)}) v_{\sigma(t)} \tag{66}$$

$$h_{\sigma(i)}(v_1, .., v_n) = \sum_{t=1}^{n} f_t^{(\sigma(i))}(v_1, .., v_n) v_t. \tag{67}$$

Taking $t = \sigma^{-1}(j)$ in (66) we get $t = j$ in (67) and matching coefficients we get

$$f_{\sigma^{-1}(j)}^{(i)}(v_{\sigma(1)}, .., v_{\sigma(n)}) = f_j^{(\sigma(i))}(v_1, .., .., v_n). \tag{68}$$

□

F Einstein summation notation

We can interpret the results from Section 3 as coming from the symmetries encoded in the Einstein summation rules.

An important early realization in differential geometry and in general relativity was that an enormous class of generally covariant forms can be written in a form that is commonly known (incorrectly perhaps) as "Einstein summation notation" [26]; it is a subset of the prior Ricci calculus [77]. This notation is a method for finding and checking equivariant quantities useful in physical laws: We imagine that we have O(d)-equivariant vectors u, v, w, and each has d components such that $[u]_i$ is the ith component of u. If we write products with repeated indices like $[u]_i\,[v]_i$, the sum rule is that the repeated index i is summed, so this corresponds to a O(d)-invariant scalar product (dot product or inner product) of the vectors u and v.

$$[u]_i\,[v]_i := \sum_{i=1}^{d}[u]_i\,[v]_i = u^\top v\ , \tag{69}$$

where the first $:=$ in (69) defines the summation notation and the second $=$ relates this sum to the linear-algebra operation on two column-oriented vectors ($d \times 1$ matrices). All indices appear either once (unsummed) or twice (summed) but never more than twice:

$$\underbrace{[u]_i\,[v]_i\,[w]_i}_{\text{indices can only appear once or twice!}} = \text{undefined}\ . \tag{70}$$

If there is one unsummed index in an expression, as in $[u]_i\,[v]_i\,[w]_j$, then the result will be an O(d)-equivariant vector:

$$[u]_i\,[v]_i\,[w]_j \equiv \left(\sum_{i=1}^{d}[u]_i\,[v]_i\right)\,w_j \equiv (u^\top v)\,w\ . \tag{71}$$

If there are ℓ unsummed indices, then the expression is an O(d)-equivariant order-ℓ tensor.

This notation also reveals that the directionality of the output is entirely encoded in the input vectors and their combination. This is counter-intuitive given some physics expressions, such as the electromagnetic force law explained in (15); it contains a cross product that typically requires a particular coordinate system to determine the direction of the result. In this case, the direction of the force cannot depend on the coordinate system; the second cross product saves us, but it is not immediately clear how. With Einstein notation, we can express the vector triple product in a coordinate-free way. We first rewrite the $d = 3$ cross product (pseudo-vector product) $a \times b$ in terms of the maximally anti-symmetric rank-3 tensor in $d = 3$ (the Levi-Civita symbol) ϵ_{ijk}:

$$f(u,v,w) = (u \times v) \times w \tag{72}$$

$$f_n = [u]_j\,[v]_k\,[w]_m\,\epsilon_{ijk}\,\epsilon_{imn}\ . \tag{73}$$

The properties of the anti-symmetric tensor products are such that this product can be re-written as

$$f_n = [u]_j\,[v]_k\,[w]_m\,[\delta_{jm}\,\delta_{kn} - \delta_{jn}\,\delta_{km}] \tag{74}$$

$$f_n = [u]_i\,[v]_j\,[w]_i - [u]_j\,[v]_i\,[w]_i \tag{75}$$

$$f(u,v,w) = (u^\top w)\,v - (v^\top w)\,u\ , \tag{76}$$

where δ_{ij} is the Kronecker delta. We thus see that the cross product can be expressed entirely in terms of scalar products ($u^\top w$ and $v^\top w$) and the input vectors (v and u), and that this can only be done by employing the anti-symmetric tensor.

Einstein notation is often credited with making physical-law expressions more compact or brief. But what's important about the notation is that if the rules are obeyed, the notation can produce only equivariant objects in the theory.

Under these summation rules, all O(d)-invariant scalar expressions that can be made from polynomial expressions of vectors will include only terms that use even numbers of vectors. All these terms can be rearranged to be written as products of invariant scalar products. That is, any scalar function that can be written as a polynomial of vectors (or a function of such polynomials) can be written in terms only of available scalar products. This result corresponds very directly to Lemma 1. For concreteness, here is an example 4-vector scalar form, written in summation notation, reordered as an inner product of tensors or as a simple product of scalars:

$$\underbrace{[u]_i\,[v]_j\,[w]_i\,[z]_j}_{\text{4-vector scalar polynomial term}} = \underbrace{([u]_i\,[v]_j)\,([w]_i\,[z]_j)}_{\text{inner product of order-2 tensors}} = \underbrace{([u]_i\,[w]_i)\,([v]_j\,[z]_j)}_{\text{product of scalars}} . \tag{77}$$

On the other hand, all O(d)-equivariant vector expressions that can be made from polynomial expressions of vectors will include only terms that use odd numbers of vectors. These terms can be rearranged by the vector with the unsummed index. Once they are rearranged this way, the expression becomes the input vectors times polynomial expressions of scalar products. This demonstrates that any O(d)-equivariant vector expression constructible in the notation will lie in the subspace spanned by the input vectors. It also demonstrates that the O(d)-invariant scalar coefficients multiplying those vectors must themselves be constructible from polynomials (or functions of polynomials, it turns out) of scalar products. What is also true but to the best of our knowledge doesn't appear explicitly stated in the physics literature is that every O(d)-equivariant polynomial vector function can be expressed in Einstein notation. This result correspond very directly to Proposition 4.

When the metric is non-trivial, we must face the covariant/contravariant distinction, where vector components might be pre-multiplied by the metric or not. In this context, there are components $[u]_i$ of the covariant vectors and components $[u]^i$ of the contravariant vectors; the Einstein summation notation rules obtain the additional rule that repeated indices must belong to covariant-contravariant pairs:

$$[u]_i\,[v]^i \equiv \sum_{i=1}^{d}[u]_i\,[v]^i \equiv \sum_{i=1}^{d}\sum_{j=1}^{d}[u]_i\,[\Lambda]^{ij}\,[v]_j \tag{78}$$

$$[v]^i \equiv \sum_{j=1}^{d}[\Lambda]^{ij}\,[v]_j\ , \tag{79}$$

where the $[\Lambda]^{ij}$ are the components of a $d \times d$ Hermitian metric tensor. In the case of the Lorentz group, $d = 4$ and the metric is diagonal, with elements $(-1, 1, 1, 1)$ on the diagonal. In the case of the curved spacetime of general relativity, $d = 4$ and the metric is (in general) a function of spatial position and time.

We expect that these results will have generalizations for scalar, vector, and tensor functions of scalar, vector, and tensor inputs.

G Connections with low-rank matrix completion

Given a rank d, $n \times n$ matrix M, Example 4 in [72] shows that M is almost always uniquely determined by the entries $\Omega(M) := M_{i,i+s}\ i = 1,..,n,\ s = 0,..,d$, considering the indices "wrap around" (i.e., $M_{i,n+s}$ corresponds to $M_{i,s} = M_{i,n+s(\text{mod } n)}$). In particular invariant functions $f : (\mathbb{R}^d)^n \to \mathbb{R}^\ell$, can be expressed as (5)

$$f(v_1,..,v_n) = g(M) = \tilde{g}(\langle v_i, v_{i+s(\text{mod } n)}\rangle_{1\leq i\leq n,\ 0\leq s\leq d}) = \tilde{g}(\Omega(M)), \tag{80}$$

where M is either $V^\top V$ or $V^\top \Lambda\, V$. Therefore, any function $\tilde{g} : \Omega(M) \to \mathbb{R}^\ell$ uniquely determines f on almost all possible inputs $v_1,..,v_n$. This observation provides a parameterization for Lorentz and orthogonal-invariant and equivariant functions with a small number of scalars.

We note that the results from [72] don't assume the matrix M is positive semi-definite, only that it is low rank. This is useful since M is not positive semi-definite in the Minkowski case. We also remark that the learning of the functions (3) and (5) can be done from $\Omega(M)$ without ever needing to compute all the scalars.

One disadvantage of the subset of scalars proposed in this section is that the sampling procedure $\Omega(M)$ is not permutation invariant. One of our future goals is to find a set of permutation invariant scalars that are universally expressive.

H Model-building considerations

The results in Section 3 provide a simple characterization of all scalar functions and vector functions that satisfy important symmetries (see Table 1) for classical physics and special relativity. Here we discuss how we might use this to design and build a model.

First we need to identify the inputs, what groups are acting, and how the group action affects the inputs and outputs. For example, the translation group acts differently on position vectors than displacement vectors (see Table 2) than other kind of vectors (velocities, accelerations, fields, etc). Moreover, permutations don't typically act on the entire set of inputs but on tuples of inputs like the individual-particle charge, position, and velocity (q_i, r_i, v_i) in the electromagnetic example in Section 4.

The causal and group-action structure of the model must correspond to the structure of the physics problem. For example, suppose we are interested in solving an n-body problem in which we predict the trajectories of a set of interacting charged particles. The target functions $h_i()$, which predict the position of particle i at time t, can be described as functions of that particle's charge, position, and velocity, and the set of charges, positions, and velocities of all the others. The group actions (rotation, translation, and permutation) can be written as:

$$h_i(q_i, Q\,r_i, Q\,v_i, (q_j, Q\,r_j, Q\,v_j)_{j=1;j\neq i}^n) = Q\,h_i(q_i, r_i, v_i, (q_j, r_j, v_j)_{j=1;j\neq i}^n) \tag{81}$$

$$h_i(q_i, r_i + w, v_i, (q_j, r_j + w, v_j)_{j=1;j\neq i}^n) = h_i(q_i, r_i, v_i, (q_j, r_j, v_j)_{j=1;j\neq i}^n) + w \tag{82}$$

$$h_i(q_i, r_i, v_i, (q_{\sigma_i(j)}, r_{\sigma_i(j)}, v_{\sigma_i(j)})_{j=1;\sigma_i(j)\neq i}^n) = h_i(q_i, r_i, v_i, (q_j, r_j, v_j)_{j=1;j\neq i}^n) \tag{83}$$

$$h_{\sigma(i)}(q_i, r_i, v_i, (q_j, r_j, v_j)_{j=1;j\neq i}^n) = h_i(q_i, r_i, v_i, (q_j, r_j, v_j)_{j=1;j\neq i}^n)\ , \tag{84}$$

where σ is any permutation on n elements, and σ_i is a permutation that fixes i, i.e., $\sigma_i(i) = i$. The primary results of this paper imply that, in this n-body case, the rotation symmetries can be enforced by constructing the invariant scalar products, and building functions thereof. The translation symmetries can be enforced by taking positional differences with respect to r_i prior to constructing the scalars. The first permutation symmetry (83) can be enforced by working on a set-based neural network (or graph neural network), and the second permutation symmetry (84) can be enforced by taking all the functions $h_i = h$ to be identical. In general, graph-based or set-based methods are probably good frameworks for physics problems [20, 80], and the dynamics can probably be implemented with a form of message-passing.

1 Synthetic Experiments

1.1 O(5) Invariant Task

A synthetic O(5) invariant regression problem given by the function

$$f(x_1, x_2) = \sin(\|x_1\|) - \frac{\|x_2\|^3}{2} + \frac{x_1^\top x_2}{\|x_1\|\|x_2\|}, \quad x_1, x_2 \in \mathbb{R}^5. \tag{1}$$

We train a neural network to find a function $g(\cdot)$ such that:

$$f(x_1, x_2) = g(x_1^\top x_1, x_1^\top x_2, x_2^\top x_2). \tag{2}$$

See results in Figure 1.

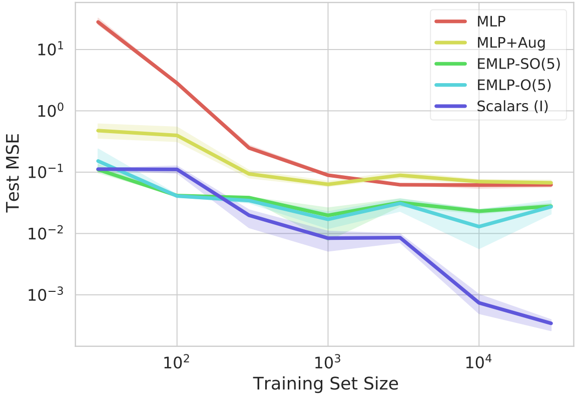

Figure 1: O(5) invariant task.

1.2 O(3) Equivariant Task

Predicting the moment of inertia matrix from $n = 5$ point masses and positions $\{(m_i, x_i)\}_{i=1}^5$,

$$\mathcal{I} = \sum_{i=1}^5 m_i(x_i^\top x_i I - x_i x_i^\top), \quad m_i \in \mathbb{R}, x_i \in \mathbb{R}^5, i = 1, 2, \ldots, 5. \tag{3}$$

The inputs $X = \{(m_i, x_i)\}_{i=1}^5$ are of type $5T_0 + 5T_1$ and outputs are of type T_2, both transform under the group.

O(d)-equivariant formulation See Proposition 4. Denote the set of 3×3 matrices $\mathcal{M} = \{x_i x_j^\top, \forall i, j\} \cup I$ (identity matrix). The idea is to approximate the following

$$\sum_{M \in \mathcal{M}} f_M\left(x_1^\top x_1, x_1^\top x_2, \ldots, x_4^\top x_5, x_5^\top x_5, m_1, m_2, \ldots, m_5\right) M, \tag{4}$$

where each corresponding f_M is some multi-layer perceptron. See `Scalars (E)` in Figure 2.

O(d)-equivariant + permutation invariant formulation See Proposition 10. The general idea is to approximate the sum of values of the two functions

$$\sum_{i,j} f_{ij}(\{x_i^\top x_j, m_i, m_j\}, \{x_k^\top x_l, m_k, m_l, \forall k, l\})\, x_i x_j^\top \tag{5}$$

and

$$f_0(\{x_i^\top x_j, m_i, m_j, \forall i, j\})I, \tag{6}$$

where f_{ij} and f_0 can be some multi-layer perceptrons. For example, we can apply a function on $x_i x_j^\top$ for all pairs of particles (i, j), $i \neq j$, and another function on $x_i x_i^\top$ for all i. To be more specific,

$$\sum_{i \neq j} f_1\big(x_i^\top x_j, \theta_1(m_i) + \theta_1(m_j), \sum_{k,l} \phi_1(x_k^\top x_l), \sum_s \psi_1(m_s)\big) x_i x_j^\top \tag{7}$$

$$\sum_i f_2\big(x_i^\top x_i, \theta_2(m_i), \sum_{k,l} \phi_2(x_k^\top x_l), \sum_s \psi_2(m_s)\big) x_i x_i^\top \tag{8}$$

$$\Big[\sum_{i\neq j} f_3\big(x_i^\top x_j, \theta_3(m_i)+\theta_3(m_j)\big) + \sum_i f_4\big(x_i^\top x_i, \theta_4(m_i)\big)\Big] I \tag{9}$$

where each f_i, $i = 1, 2, 3, 4$, θ_j, $j = 1, 2, 3$, ϕ_k, $k = 1, 2$, ψ_l, $l = 1, 2$ is some multi-layer perceptron. See `Scalars (E+P)` in Figure 2.

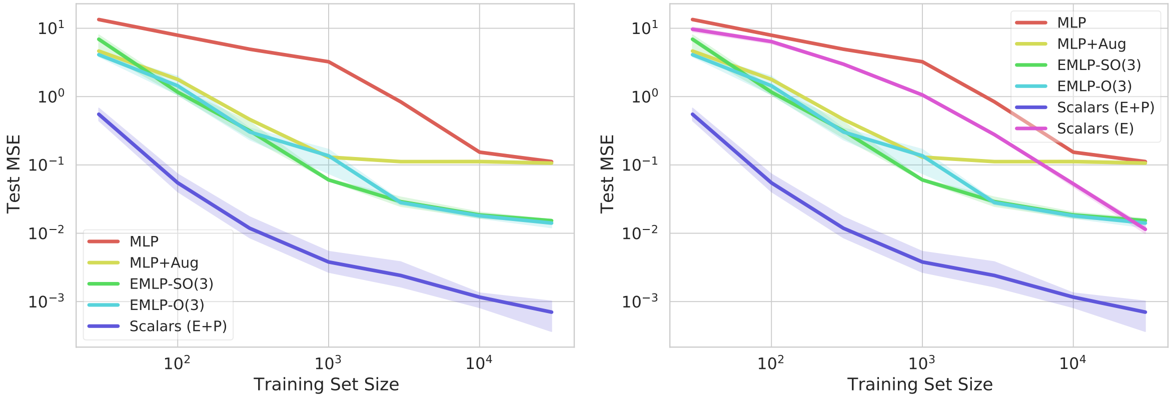

Figure 2: O(3) equivariant task. All perceptrons used in `Scalars (E+P)` and `Scalars (E)` are with 2 layers and 200 hidden nodes.