Equivariant Tensor Networks

M. $Hodapp^{*\dagger 1}$ and A. $Shapeev^{\ddagger 2}$

¹Materials Center Leoben Forschung GmbH (MCL), Leoben (AT)
²Skolkovo Institute of Science and Technology (Skoltech), Center for Artificial Intelligence Technology, Moscow (RU)

April 18, 2023

Abstract

Machine-learning interatomic potentials (MLIPs) have made a significant contribution to the recent progress in the fields of computational materials and chemistry due to MLIPs' ability of accurately approximating energy landscapes of quantum-mechanical models while being orders of magnitude more computationally efficient. However, the computational cost and number of parameters of many state-of-the-art MLIPs increases exponentially with the number of atomic features. Tensor (non-neural) networks, based on low-rank representations of high-dimensional tensors, have been a way to reduce the number of parameters in approximating multidimensional functions, however, it is often not easy to encode the model symmetries into them.

In this work we develop a formalism for rank-efficient equivariant tensor networks (ETNs), i.e., tensor networks that remain invariant under actions of SO(3) upon contraction. All the key algorithms of tensor networks like orthogonalization of cores and DMRG carry over to our equivariant case. Moreover, we show that many elements of modern neural network architectures like message passing, pulling, or attention mechanisms, can in some form be implemented into the ETNs. Based on ETNs, we develop a new class of polynomial-based MLIPs that demonstrate superior performance over existing MLIPs for multicomponent systems.

Contents

	1.1	Oduction 3 Motivation 3 Outline of the present work 4
2	A Sh 2.1	Tensor Diagrams
	2.3	Tensor Trains

^{*}corresponding author

[†]maxludwig.hodapp@mcl.at

[‡]a.shapeev@skoltech.ru

3	SO(3)-Equivariant Tensor Networks 8 3.1 Order-1 Tensors (Vectors) 10 3.2 Order-2 Tensors (Matrices) 10 3.3 Order-3 Tensors (where the Magic Happens) 11 3.4 Construction of Equivariant Tensor Networks 12 3.4.1 Equivariant Tensor Trains 12 3.4.2 Invariance under O(3) 13
4	Equivariant Tensor Network Numerical Algebra134.1 Trivial algorithms: gradient descent and alternating least squares144.2 QR or SVD normalization of cores144.2.1 Order-2 cores144.2.2 Order-3 cores164.3 DMRG algorithm18
5	Application to Interatomic Potentials195.1 Equivariant Tensor Train Representation of Per-Atom Energies195.2 Non-Angular Features205.3 Equivariant Tensor Network Potentials205.4 Hyperparameters of Tensor Network Potentials22
6	Performance of Equivariant Tensor Network Potentials 6.1 Optimization of the Hyperparameters 22 6.2 Numerical Examples 24 6.2.1 QM7 24 6.2.2 QM9 24 6.2.3 MoNbTaW Medium-Entropy Alloy 25 6.2.4 MoNbTaVW High-Entropy Alloy 25 6.2.5 Binary Alloys with 10 different Species 26 6.2.6 Evolution of the Hyperparameters 27 6.3 Comparison with non-polynomial MLIPs 28
7	Generalizations to Other Tensor Networks307.1Arbitrary topology of tensor network cores307.2Weight sharing307.3Richer atomic features307.4Message passing/graph convolutions, and pooling operations32
8	Concluding Remarks 33
9	Acknowledgments 34
A	Wigner 3-j Symbol for Real Spherical Harmonics 34 A.1 Selection Rules 34 A.2 Sparsity 36 A.3 Contraction Properties 38 A.4 Symmetry Properties 38 A.5 Orthogonality Properties 39
В	Functional forms of other Machine-Learning Interatomic Potentials B.1 Moment Tensor Potentials

References 41

1 Introduction

1.1 Motivation

Machine learning has reached a level of maturity that has spawned technology in many fields of science now accessible to non-experts. Prominent examples are DALL-E in computer vision, AlphaGo in board games, or ChatGPT in linguistics. In atomistic modeling, machine-learning models, or, in the field-specific language, machine-learning interatomic potentials (MLIPs), have been developed to approximate the energy landscape of an underlying quantum-mechanical model, and reductions in computational cost of several orders of magnitude have been reported, while preserving the accuracy of quantum mechanics. The three major classes of MLIPs are neuralnetwork-based potentials [1]-[7], polynomial potentials [8]-[10], and Gaussian process regressionbased potentials [11]-[13]. It is believed, and the lack of successful counterexamples is a good proof of it, that rotational and permutational (i.e., with respect to permuting chemically equivalent atoms) must be incorporated in the MLIP's functional form. For neural networks it has traditionally been achieved by using invariant input features [1]-[3], [5], [6], which was later generalized to covariant features and equivariant neural-network feature transformations [7]. The state-of-the-art polynomial models also work with covariant features and their algebraic transformations (addition and multiplication) in an equivariant manner [8]-[10]. Gaussian process-based models use invariant kernels [11] or basis functions [12], [13]. A very interesting case is [14] which is both, a neural network model whose output depends polynomially on the input features.

A recent benchmark study [15] has been conducted to compare the accuracy-vs-efficiency performance of different models on unary (i.e., with one chemical component) systems evaluated on a CPU. This study, at least in its small-dataset, a polynomial model, MTP [9], showed a comparatively favorable performance. Despite this and other success stories, polynomial-based models suffer from exponentially growing complexity when the number of features is large. In other words, neural-network potentials regain their advantage when applied to systems with complex quantum-mechanical phenomena (e.g., magnetism or electronic temperature) or a large chemical space (e.g., high-entropy alloys) since the latter models have a more controlled growth of the parameter space as the number of features increases [16]–[19].

The problem of a large parameter space can be described as follows. Polynomial MLIPs are potentials whose functional form is constructed by taking polynomials of atomic features (positions, atomic species, magnetic moments, etc.). Polynomial MLIPs can be encoded with tensors, i.e., their functional form for a *d*-th order polynomial usually consists of (linear) combinations of scalar tensor contractions of the following type

$$C_{i_1...i_n}^{1}...i_n^{d}...i_n^{d}F_{i_1...i_n}^{1}...F_{i_1...i_n}^{d},$$
 (1)

where $C_{i_1^1\dots i_n^1\dots i_n^d\dots i_n^d}$ is the polynomial coefficient (parameter) tensor and F is the feature tensor. Evidently, this representation becomes increasingly inefficient with an increasing number of features n, more precisely, the total number of parameters is proportional to \bar{m}^{dn} , where \bar{m} is the average size over all dimensions of the feature tensor. Even with only radial, angular, and species features, the amount of parameters can quickly increase to the order of one thousand for alloys with a large number of components, such as high-entropy alloys.

One approach to attempt reducing the amount of free parameters is to convert the coefficient tensor in (1) into a low-rank tensor format that approximates multivariate polynomials, e.g., the tensor train (TT) format [20], [21]. Incorporation of symmetries into tensor networks if often not easy. For example, in [22] a tensor train representation was used to approximate on-lattice atomic energies as functions of chemical degrees of freedom, and permutational symmetry was

build in by explicitly symmetrizing the tensor over the corresponding point group of the lattice. In the present setting, we require that group actions of SO(3) on atomic positions must leave (1) invariant; physically speaking, the (off-lattice) potential energy of a configuration of atoms must not change under pure rotations. A related very recent development approaches this problem by utilizing a symmetric canonical tensor decomposition of the coefficient tensor [23].

1.2 Outline of the present work

In this work we propose **equivariant tensor networks** (ETNs), a formalism for constructing interatomic interaction models based on tensor networks that are invariant under the action of SO(3). In particular, we develop a tensor network algebra (Section 2) and use it to develop the following three building blocks that compose our ETNs (Section 3):

- covariant vectors v, i.e., vectors that rotate correspondingly with a basis change under actions of SO(3),
- \bullet equivariant order-3 tensors T, i.e., tensors that describe equivariant maps of the covariant vectors, and
- feature contractions, i.e., parameterized contractions that map the feature tensors $F_{i_1...i_n}$ to some reduced covariant vector v.

Here and in what follows, the term "SO(3)-covariant vector" will be used in the context of a "feature vector" and refers to a collection, possibly large, of numbers that change (or remain constant) upon rotation, rather than specifically an element of \mathbb{R}^3 . We show that arbitrary ETNs expressing polynomials of arbitrary degree can be constructed entirely from these three building blocks.

We present the formalism in the language conventional to the field of numerical linear algebra, although we emphasize that many key ideas already exist in the field of quantum physics. In particular, the recent developments [24] concern SU(2)-equivariant tensor networks representing quantum states of qubit systems that recently receive a lot of attention in the context of quantum computing. In these tensor networks, SU(2) symmetries are incorporated using the Wigner-Eckhart Theorem that states that any spherical tensor T with three multi-indices $\{(\ell_i, m_{\ell_i}, n_{\ell_i})\}_{i=1,\dots,3}$, where ℓ is the angular momentum, and m is the phase, can be factorized to

$$T_{(\ell_1, m_{\ell_1}, n_{\ell_1})(\ell_2, m_{\ell_2}, n_{\ell_2})(\ell_3, m_{\ell_3}, n_{\ell_3})} = P_{(\ell_1, n_{\ell_1})(\ell_2, n_{\ell_2})(\ell_3, n_{\ell_3})} Q_{(\ell_1, m_{\ell_1})(\ell_2, m_{\ell_2})(\ell_3, m_{\ell_3})}, \tag{2}$$

where P contains the degrees of freedom and is independent of the phase m_{ℓ_i} , and Q is the Clebsch-Gordan coefficient, a constant tensor that is defined by the symmetry group. Key algorithms like DMRG (density matrix renormalization group) are formulated for such tensor networks [24].

In our work, we make full use of the structure of SO(3)-equivariance, namely, the possibility of building the irreducible representation based on real-valued spherical harmonics. In the real-valued case, vector contraction and dot product are the same, which enables us to build them using the Wigner 3-j coefficients. The latter are symmetric with respect to permutations of (ℓ_i, m_{ℓ_i}) which results into undirected networks (they are directed in the SU(2) case due to the lack of symmetry of the Clebsch-Gordan coefficients, which results into network nodes of different types based on the direction of the adjacent edges). This allows us to formulate the usual operations of tensor network algebra, like reshaping an order-3 tensor core into an order-2 tensor, combining two cores together, or performing decompositions and splitting of the cores, while preserving the symmetric structure of the network. Furthermore, in our framework it

 $^{^{1}}$ in the quantum mechanics, P is usually referred to as the "reduced matrix element"

is easy to also separate the reflection symmetric/antisymmetric features, resulting into the full O(3)-equivariance of the tensor network.

We exemplify the construction of ETNs by developing an equivariant version of the TT format

$$T_{i_1j_2}T_{j_2i_2j_3}\dots T_{j_di_d}v_{i_1}\dots v_{i_d}.$$
 (3)

We will show that the number of parameters of this representation is proportional to $(d+n)\bar{m}\bar{r}^2$, where \bar{r} is the average rank over all T's. Hence, if there is enough similarity between the features, the ranks can be made small and the representation (3) will be more efficient than the "raw" polynomial (1). We also show how algorithms developed for conventional tensor networks, e.g., for optimizing tensor parameters, tensor orthogonalization, etc., carry over to the equivariant case (Section 4).

With our ETN formalism, we develop a new class of MLIPs that we henceforth refer to as the class of **ETN potentials** (Section 5). In particular, we explicitly develop and implement an ETN potential that has atomic position and species features, and show that this ETN potential requires significantly fewer parameters than MTPs to reach the same level of accuracy for QM7, QM9, and several datasets for metallic alloys (Section 6).

Further, we outline how our formalism can be used to construct more general ETN potentials (Section 7). Examples include extensions to arbitrary ETN topologies, more general feature tensors, nonlocal atomic energies, and other symmetry groups.

The key advantage of our new formalism is its modularity due to the rather small set of required operations; in essence, tensor operations on up-to-order-3 tensors. This heavily simplifies generic implementations and the automation of operations other than contractions to be performed on the ETNs, such as automatic differentiation, factorization, etc. Our formalism for constructing ETN potentials can thus be, in some sense, considered as an interatomic potential modular building system ("interatomic potential Lego") that allows for a rapid construction of efficient representations of polynomial MLIPs with arbitrary order and physical complexity.

2 A Short Introduction to Tensor Networks

Here we introduce the notions and notations needed to present equivariant tensor networks. We do not do it with maximal generality for the sake of conciseness of the presentation. Nevertheless, many features present in the conventional tensor networks are also present in the equivariant tensor networks; we remark on this in Section 7.

tensor networks; we remark on this in Section 7. A tensor is a multidimensional array $T_{\langle d \rangle} \in \mathbb{R}^{N_1 \times N_2 \times ... \times N_d}$, where d will be called the number of dimensions or simply the order of the tensor (we reserve the term rank to speak about, e.g., the rank of a matrix, or its generalization to tensors). Since we will mostly work with tensors up to the order of three, we use a notation using underscores for those tensors, namely, $v_{\langle 1 \rangle} = \underline{v}$ for an order-1 tensor (or vector), $U_{\langle 2 \rangle} = \underline{U}$ for an order-2 tensor (or matrix), and $T_{\langle 3 \rangle} = \underline{T}$ for an order-3 tensor. We will also use the index notation where, e.g., \underline{v} , \underline{U} , and \underline{T} , are written as v_i , U_{ij} , and T_{ijk} . For tensor contractions we will generally adopt the Einstein notation. In case we do not sum over all duplicate indices we will specify the contracted indices explicitly.

2.1 Three is the magic tensor order

A vector \underline{u} can be used to model a dependence u_i on one discrete variable i, or one continuous variable upon choosing a (finite) basis v_i and expanding the one-dimensional function in this basis with coefficients u_i . The latter motivates us to consider a contraction $\alpha_{\underline{u}}(\underline{v}) = \underline{u} \cdot \underline{v}$ corresponding to \underline{u} and interpret it as a linear form over \underline{v} . This way, a tensor of order d, $T_{\langle d \rangle}$, can be thought

of describing a quantity dependening on d variables and can be associated with a multilinear form

$$\alpha = \alpha(\underline{v}^1, \dots, \underline{v}^d) = (\dots (T_{\langle d \rangle} \underline{v}^d) \underline{v}^{d-1} \dots) \underline{v}^1.$$
(4)

Of course, the simplest tensor is a scalar (zero-order tensor), but it is hardly sufficient to model anything of nontrivial complexity. A vector \underline{v} can be used to model a dependence on one variable, while an outer product of two vectors is an order-two tensor $\underline{u} \otimes \underline{v}$, however, it can only describe a two-dimensional function whose two variables are essentially independent of each other—again, we can conclude that a vector is too simple to describe complex dependencies. Of course, it is known that any dependence can be represented as a sum of $\underline{u}_k \otimes \underline{v}_k$, however, this brings us to contractions of higher-order tensors.

Namely, the sum $\underline{u}_k \otimes \underline{v}_k$ is more conveniently interpreted as a product of two matrices $\underline{\underline{U}}\underline{\underline{V}}$ —which is a matrix (order-two tensor) itself. Here, by writing AB we mean that the last dimension(s) of A is contracted with the first dimension(s) of B. By taking contractions of matrices one can only obtain tensors of order two or less which is, again, not sufficient to describe nontrivial multivariate dependencies.

Order-three tensors make a difference. With two order-three tensors $\underline{\underline{A}}$ and $\underline{\underline{B}}$ by contracting one dimension in them one can obtain a nontrivial (in the sense defined above) order-four tensor $\underline{\underline{A}}\underline{\underline{B}}$; with three tensors one can obtain a nontrivial order-five tensor $\underline{\underline{A}}\underline{\underline{B}}\underline{\underline{C}}$, etc. Thus, three is the minimal tensor order which allows representing nontrivial dependencies on any number of variables—in this sense we say that *three* is the magic number for the tensor order.

2.2 Tensor Diagrams

Tensorial operations on vectors and matrices can easily be represented with formulas (as a single string of products of multiple vectors and matrices), however, when the tensor order goes above two, it is much easier to represent the operations with diagrams. In Figure 1 we show the basic tensors and operations. The tensors are represented with boxes, their dimensions are represented with links, and the diagram represents the result of contractions of tensors with themselves. If two tensor dimensions are connected, it means that they are contracted in the result. If there are no "hanging" links then the result is scalar.

2.3 Tensor Trains

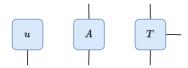
As follows from the above, a multilinear form can be realized by contractions with multiple order-two and order-three tensors:

$$\alpha(\underline{v}^1, \dots, \underline{v}^d) = \underline{\underline{T}}^1 \Big(\dots \underline{\underline{T}}^{d-1} \Big(\underline{\underline{T}}^d \underline{v}^d \Big) \underline{v}^{d-1} \dots \Big) \underline{v}^1, \tag{5}$$

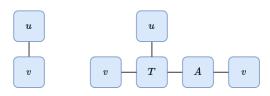
where $\underline{\underline{T}}^1$ and $\underline{\underline{T}}^d$ are order-2 tensors and $\underline{\underline{T}}^2, \ldots, \underline{\underline{T}}^{d-1}$ are order-3 tensors. Thus, the right-hand side of (5) is a product of two vectors and d-2 matrices yielding a scalar. The multilinear form (5) is nothing but the tensor train (TT) representation [20] of the associated order-d tensor which has been known in physics under the term matrix product state (MPS, see, e.g., [25], [26]). The graphical representation of (5) in terms of tensor network diagrams is

$$\alpha(\underline{v}^{1}, \dots, \underline{v}^{d}) = \begin{array}{cccc} v^{1} & v^{2} & v^{d} \\ & & & \\ j_{1} & j_{2} & j_{d} \\ \hline T^{1} & i_{2} & T^{2} & i_{3} & \cdots & i_{d} & T^{d} \end{array}$$

$$(6)$$



(a) Vector u, matrix A and order-3 tensor T can be distinguished by the number of links (=dimensions) attached to it



(b) Tensorial contractions, from simple like vector product (left) to more complex (right) can be illustrated by linking tensors, which corresponds to contracting over those dimensions



(c) The identity matrix δ_{ij} (left) is illustrated as a simple link, because, e.g., linking vectors u through the scalar product $\sum_i u_i v_i$ and v directly (middle), or through multiplying by the identity matrix (right) gives the same result

$$\begin{bmatrix} k \\ \vdots \\ i \end{bmatrix} = \begin{bmatrix} v \\ \end{bmatrix}$$

(d) We also need the "identity order-3 tensor" $\underbrace{I_{ijk}}_{\equiv ijk} = [i=j=k]$ (left) which can be used to express pointwise product, e.g., $u_i = v_i w_i$ (right)

Figure 1: Example of diagrams of basic tensors and operations on them.

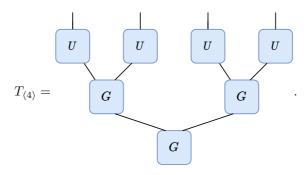
In these diagrams, a tensor is represented with a square block, with connections between blocks being contractions over the tensors' dimensions. Our notation of tensor network diagrams is inspired by tensor network diagrams used in quantum physics (e.g., [27]) and appears to us very convenient for representing tensors (and tensor operations) in our setting, in particular, in view of comparing different variants of equivariant tensor networks later on. To show the power of this notation, we remark that the order-d tensor $T_{\langle d \rangle}$ corresponding to (5) is not easy to write down in mathematical formulas, but very easy to represent as a visual diagram:

From this diagram one can see that an order-three tensor (like T^2) is a fundamental building block which allows to construct tensors of any order, which reiterates the point made in the previous subsection.

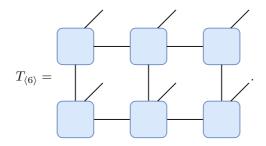
2.4 Other Tensor Networks

By utilizing the graphical representation of tensor contractions, let us give examples of other tensor networks. One prominent example is the Hierarchical Tucker representation [28] which was proposed in the field of tensor representations at about the same time as tensor train. On

a diagram it looks like



Another, more complex example is PEPS [29] which additionally features a two-dimensional arrangement of tensor indicies:



3 SO(3)-Equivariant Tensor Networks

Our motivation is to use (4) to approximate O(3)-invariant functionals (e.g., interatomic interaction energies) of local atomic environments (or, more generally, clouds of points). O(3) invariance means that, if the atomic environments are rotated or reflected in the three-dimensional space, then the functional α does not change. We separate the problem of O(3) invariance into SO(3) (rotation) invariance and reflection invariance. Mathematically, invariance of (4) with respect to a group G (for example, G = SO(3)), means that when \underline{v} changes under a group action (in our example, rotation) $\underline{v} \mapsto g \cdot \underline{v}$, then α remains constant. Note that the model parameters, $T_{\langle d \rangle}$, are not allowed to change with the group action (because the interatomic interaction model is independent of the frame of reference). Often, together with the invariance comes the notion of equivariance. For instance, for (5) to be invariant, the matrix $\underline{\underline{T}}$ does not only have to be SO(3)-invariant component-wise, but also SO(3)-equivariant, i.e., $\underline{g}^{-1} \cdot \underline{\underline{T}}(g \cdot \underline{v}) = \text{const}$ as \underline{v} rotates. We note that the equivariance of the operator $\underline{\underline{T}}$ is equivalent to the invariance of the quadratic form associated with \underline{T} , i.e., $(\underline{T}(g \cdot \underline{v})) \cdot (g \cdot \underline{v}') = \text{const}$.

We are hence interested in two kinds of objects: vectors (and sometimes tensors) rotating under the action of G, and tensors that remain constant but otherwise describe an equivariant transformation of the former vectors. The first kind of objects are typically the input features of a model (with an arbitrarily chosen frame of reference) or transformed features occurring in the middle of a calculation, while the second kind are model coefficients that "do not know" which frame of reference was chosen for the input. The first kind of objects are *covariant vectors*, i.e., vectors that change under rotations as $\underline{v}' = \underline{D}^\mathsf{T}\underline{v}$, where \underline{D} is a Wigner D-matrix, while the second kind of objects will be called *equivariant* tensors. In what follows, by "contraction" of tensors with anything else we mean invariant or equivariant contractions, depending on the context, unless it is explicitly mentioned otherwise.

Following the results of representation theory, for a covariant vector \underline{v} , we consider its decomposition into an irreducible representation. In the case of SO(3), this corresponds to spherical harmonics components with different angular momenta. Hence, we consider \underline{v} as a multi-index vector $v_{(\ell m n)}$, $\ell = 0, \ldots, L$, where L is the maximal angular momentum, $m \in \{-\ell, -\ell+1, \ldots, \ell\}$ is the phase, and $n = 1, \ldots, N(\ell)$ is the number of the components corresponding to ℓ . We remark that we deviate intentionally from the notation $v_{(n\ell m)}$ commonly used in quantum physics that puts the index n, usually referred to as "principal quantum number", in front of ℓm (cf., e.g., [30]). Our reasoning in doing so is due to the fact that we will later on work with higher-dimensional tensors, where each multi-index may depend on a different ℓ . Since m and n are always assumed to depend on ℓ , using the order $\ell m n$ appears more convenient to us for the sake of clarity. In the spirit of machine learning, we will call $N(\ell)$ as the number of angular momentum channels corresponding to each ℓ . We assume that the action of SO(3) on $v_{(\ell m n)}$ follows the action of SO(3) on the real spherical harmonic (RSH) function $\dot{Y}_{\ell m}$ defined as follows

$$\dot{Y}_{\ell m} = \begin{cases}
\frac{i}{\sqrt{2}} \left(Y_{\ell - |m|} - (-1)^m Y_{\ell |m|} \right) & m < 0, \\
Y_{\ell 0} & m = 0, \\
\frac{1}{\sqrt{2}} \left(Y_{\ell - |m|} + (-1)^m Y_{\ell |m|} \right) & m > 0,
\end{cases} \tag{8}$$

where the $Y_{\ell m}$'s are the (complex) spherical harmonics.

Using (8), we introduce the mapping $U_{\ell mm'}: Y_{\ell m'} \to \dot{Y}_{\ell m}$ [31], i.e.,

$$\underline{\underline{U}}_{\ell} = \frac{1}{\sqrt{2}} \begin{pmatrix} i & & & & & -(-1)^{\ell-1} i \\ & i & & & & -(-1)^{\ell-1} i \\ & & \ddots & & \ddots & & \\ & & & \sqrt{2} & & & \\ & & \ddots & & \ddots & & \\ & 1 & & & & (-1)^{\ell-1} & \\ 1 & & & & & & (-1)^{\ell} \end{pmatrix}.$$
(9)

Since $\underline{\underline{U}}_{\ell}$ is a unitary matrix, its inverse is its conjugate transpose $\underline{\underline{U}}_{\ell}^{-1} = \underline{\underline{U}}_{\ell}^{\mathsf{T}*}$. We then define the Wigner 3-j symbols (or just 3-j symbols) for RSHs as

$$\begin{cases}
\ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3
\end{cases} = \sum_{m'_1, m'_2, m'_3} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m'_1 & m'_2 & m'_3 \end{pmatrix} U_{\ell_1 m'_1 m_1}^{\mathsf{T}*} U_{\ell_2 m'_2 m_2}^{\mathsf{T}*} U_{\ell_3 m'_3 m_3}^{\mathsf{T}*}.$$
(10)

The 3-j symbols are nonzero only when the three ℓ 's and m's satisfy the usual two selection rules

[SR1]
$$\operatorname{TR}(\ell_1, \ell_2, \ell_3)$$
 if, by definition, $\max_{k}(\ell_k) \leq \min_{i \neq j}(\ell_i + \ell_j)$, (11)

[SR2]
$$m_1 + m_2 + m_3 = 0,$$
 (12)

where $TR(\ell_1, \ell_2, \ell_3)$ is the triangular inequality. The 3-j symbols for RSHs can also be nonzero for other linear combinations of m's. More precisely, for the 3-j symbols for RSHs, the selection rule [SR2] is replaced by the following two selection rules

[SR3]
$$\operatorname{sgn}(m_1)\operatorname{sgn}(m_2)\operatorname{sgn}(m_3) = \pm 1,$$
 (13)

where $\operatorname{sgn}(\bullet)$ is the sign function, and the + and - hold for even or odd $\ell_1 + \ell_2 + \ell_3$, respectively, and

$$[\mathbf{SR4}] \begin{cases} [\mathbf{SR4.1}] & m_1 + m_2 + m_3 = 0, \\ [\mathbf{SR4.2}] & m_1 + m_2 - m_3 = 0, \\ [\mathbf{SR4.3}] & m_1 - m_2 + m_3 = 0, \\ [\mathbf{SR4.4}] & -m_1 + m_2 + m_3 = 0. \end{cases}$$
(14)

The selection rule [SR4] holds true if (at least) one of the (sub-)selection rules [SR4.1]–[SR4.4] holds true.

The selection rules [SR3] and [SR4] follow from the properties of $\underline{\underline{U}}_{\ell}^{\mathsf{T}*}$ (see Appendix A.1). Although the additional selection rules suggest that the 3-j symbol for RSHs is much less sparse than the usual 3-j symbol, it is, in fact almost as sparse as the usual 3-j symbol and does not sacrifice computational efficiency (of tensor contractions) for convenience (undirected tensor networks) as shown in Appendix A.2.

Moreover, we refer to Appendix A.3 for the equivalence of the 3-j symbol for RSHs and the Clebsch-Gordan coefficients in adding angular momenta. However, we point out that the 3-j symbols for RSHs obey *almost the same symmetries* as the usual 3-j symbol (cf., Appendix A.4).

In addition, we remark that the 3-j symbol for RSHs (10) is real whenever $\ell_1 + \ell_2 + \ell_3$ is even, and imaginary otherwise (cf., Appendix A.1 for further details). This feature of the 3-j symbol for RSHs will be important for the full O(3) (reflection + rotation) invariance.

With our definition of the 3-j symbol for RSHs, we now define the basic operations to be performed on covariant vectors that we require to build equivariant tensor networks.

Remark 1. At this point, we remark that the choice of the basis is not unique. One could likewise choose invariant polynomials (invariant with respect to rotation and reflection) as the basis set [9]. However, within the tensor network formalism we propose here, the operators that generate those invariant polynomials are not available off-the-shelf (such as for spherical harmonics). It could be nevertheless intersting to explore other basis sets that would lead to potentially more sparse representations.

3.1 Order-1 Tensors (Vectors)

The basic operations we require are the contraction of two covariant vectors

$$\underline{u} \cdot \underline{v} = \sum_{\ell,m,n} u_{(\ell m n)} v_{(\ell m n)},\tag{15}$$

and equivariant contractions of a vector \underline{c} with a covariant vector. The latter can only be performed with respect to the 0-th angular momenta such that

$$\alpha = \alpha(\underline{v}) = \langle \underline{c}, \underline{v} \rangle = \sum_{n} c_n v_{00n}. \tag{16}$$

In other words, (16) is the most general equivariant contraction.

3.2 Order-2 Tensors (Matrices)

Instead of defining the outer product of two covariant vectors as simply $u_{(\ell mn)}v_{(\ell'm'n')}$, we instead reinterpret it as a covariant vector

$$(\underline{u} \otimes \underline{v})_{(\ell m (\ell' \ell'' n' n''))} = \sum_{m', m''} \begin{cases} \ell' & \ell'' & \ell \\ m' & m'' & m \end{cases} u_{(\ell' m' n')} v_{(\ell'' m'' n'')}. \tag{17}$$

Here, the index on the left-hand side should be understood as follows: for each n', n'' and a pair of ℓ' and ℓ'' satisfying $\mathrm{TR}(\ell',\ell'',\ell)$ we have one angular momentum channel $n=(\ell'\ell''n'n'')$. Altogether, for each ℓ , the outer product has $\sum_{\ell',\ell'':\mathrm{TR}(\ell,\ell',\ell'')} N'(\ell') N''(\ell'')$ angular momenta channels. We chose the 3-j symbols rather than the Clebsch-Gordan coefficients to emphasize their symmetry with respect to permuting the tensorial dimensions.

We remark that we use the notation (17) instead of, e.g., the one used in quantum physics (cf., bipolar spherical harmonics [32]),

$$\left(\underline{u}_{\ell'} \otimes \underline{v}_{\ell''}\right)_{(\ell m n)} = \sum_{m', m''} \begin{Bmatrix} \ell' & \ell'' & \ell \\ m' & m'' & m \end{Bmatrix} u_{(\ell' m' n')} v_{(\ell'' m'' n'')}, \tag{18}$$

implying that this is nonzero only when $TR(\ell', \ell'', \ell)$, because we would like that the number of channels is explicitly featured in the notation.

An equivariant contraction of an arbitrary order-2 tensor $\underline{\underline{C}}$ and two covariant vectors is then given by

$$\alpha = \alpha(\underline{u}, \underline{v}) = \sum_{\ell, m, n, n'} C_{\ell n n'} u_{(\ell m n)} v_{(\ell m n')}. \tag{19}$$

Note that we write $C_{\ell nn'}$ with three indices, and, so, it appears to be an order-3 tensor—but we use it in the following as a shorthand notation for a block-diagonal order-2 tensor $C_{(\ell n)(\ell n')}$, with blocks $\underline{\underline{C}}_{\ell}$ corresponding to different ℓ being arbitrary, regular $N(\ell) \times N'(\ell)$ matrices. Hence, many operations that can be done on matrices, such as the QR- or SVD-decomposition, can be done in a block-wise fashion, as shown in Section 4.

The shortest (but not necessarily simplest) way to prove (19) is to consider the outer product of \underline{u} and \underline{v} (17) (only the $\ell = 0$ component matter²)

$$(\underline{u} \otimes \underline{v})_{(00nn')} = \sum_{m,m'} \begin{Bmatrix} \ell & \ell' & 0 \\ m & m' & 0 \end{Bmatrix} u_{(\ell m n)} v_{(\ell' m' n')}$$
(21)

$$\stackrel{\text{(A7)}}{=} \sum_{m} (-1)^m \begin{pmatrix} \ell & \ell & 0 \\ m & -m & 0 \end{pmatrix} u_{(\ell m n)} v_{(\ell m n')}$$
(22)

$$= \sum_{m} \frac{(-1)^{\ell}}{\sqrt{2\ell+1}} u_{(\ell m n)} v_{(\ell m n')}, \tag{23}$$

and then using the arbitrary vectorial contraction (16) noting that the scaling $\frac{(-1)^{\ell}}{\sqrt{2\ell+1}}$ can be adsorbed into $C_{\ell nn'}$.

3.3 Order-3 Tensors (where the Magic Happens)

In the previous subsection we have already encountered a special case of an order-3 equivariant tensor—the 3-j symbol for RSHs. If we consider an arbitrary order-3 tensor $\underline{\underline{T}}$ given by a trilinear form

$$\alpha = \alpha(\underline{u}, \underline{v}, \underline{w}) = T_{(\ell_1 m_1 n_1)(\ell_2 m_2 n_2)(\ell_3 m_3 n_3)} u_{(\ell_1 m_1 n_1)} v_{(\ell_2 m_2 n_2)} w_{(\ell_3 m_3 n_3)}.$$
(24)

$$\begin{pmatrix} \ell & \ell & 0 \\ m & -m & 0 \end{pmatrix} = \frac{(-1)^{\ell-m}}{\sqrt{2\ell+1}}.$$
 (20)

²We hence use the fact that [32]

and require that α stays invariant as SO(3) acts simultaneously on \underline{u} , \underline{v} , and \underline{w} , we will arrive at the following general representation of $\underline{\underline{T}}$,

$$T_{(\ell_1 m_1 n_1)(\ell_2 m_2 n_2)(\ell_3 m_3 n_3)} = C_{(\ell_1 n_1)(\ell_2 n_2)(\ell_3 n_3)} \left. \begin{cases} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{cases} \right\}, \tag{25}$$

where the coefficient tensor \subseteq can be arbitrary. Due to the properties of the 3-j symbols for RSHs, $C_{(\ell_1 n_1)(\ell_2 n_2)(\ell_3 n_3)}$ can be defined as nonzero only when the triangular inequality (11) is satisfied.

The bilinear and linear forms can be obtained by simply substituting $(\ell_i m_i n_i) = (0,0,0)$ for one or two indices of the above equations. In particular, a bilinear form can be written with $\underline{\underline{T}}$ in the following manner

$$\alpha = \alpha(\underline{u}, \underline{v}) = \langle \underline{u}, \underline{\underline{T}}\underline{v} \rangle = T_{(\ell_1 m_1 n_1)(\ell_2 m_2 n_2)(000)} u_{(\ell_1 m_1 n_1)} v_{(\ell_2 m_2 n_2)}$$

$$= T_{(\ell m n_1)(\ell m n_2)} u_{(\ell m n_1)} v_{(\ell m n_2)},$$
(26)

or

$$\alpha = \alpha(\underline{v}, \underline{w}) = \langle \underline{v}, \underline{\underline{T}}\underline{w} \rangle = T_{(000)(\ell_2 m_2 n_2)(\ell_3 m_3 n_3)} v_{(\ell_2 m_2 n_2)} w_{(\ell_3 m_3 n_3)}$$

$$= T_{(\ell m n_2)(\ell m n_3)} v_{(\ell m n_2)} w_{(\ell m n_3)}.$$
(27)

For completeness, linear forms can be written with $\underline{\underline{T}}$ as follows

$$\alpha = \alpha(\underline{u}) = \langle \underline{T}, \underline{v} \rangle = T_{(000)(\ell_2 m_2 n_2)(000)} v_{(\ell_2 m_2 n_2)} = T_{(00n_2)} v_{(00n_2)}.$$
(28)

In the following, we tacitly refer to $T_{(\ell_1 m_1 n_1)(\ell_2 m_2 n_2)(\ell_3 m_3 n_3)}$ (and also $\underline{\underline{T}}$) as equivariant tensors without explicitly mentioning the dimension.

It is known that equivariant contractions of higher-order (higher than three) covariant vectors can be expressed through products of the 3-j (or Clebsch-Gordan) coefficients, so we do not have to explicitly consider higher-order equivariant tensors. This turns into a very interesting coincidence: similarly to how order-3 tensors were sufficient to generate a tensor of any order by repeatedly contracting the order-3 tensors, a third-order equivariant tensor contain the needed information to encode equivariance in any higher-order tensors. The algorithmic combination of these two facts is the core of this work; everything else can be thought of as a corollary of this.

3.4 Construction of Equivariant Tensor Networks

Using our definition of equivariant tensors, we now exemplify the construction of equivariant tensor networks (ETNs) using the tensor train format [20]. We emphasize that, in principle, any topology of tensor networks tensors can be realized also with the equivariant tensors, including the already mentioned hierarchical Tucker decomposition [28], the ring tensor format [33], or more advanced networks like PEPs [29].

3.4.1 Equivariant Tensor Trains

In the tensor train format, a multilinear form is a product of equivariant tensors with covariant vectors, leading to the representation

$$\alpha(\underline{v}^{1}, \dots, \underline{v}^{d}) = \underline{\underline{T}}^{1} \left(\dots \underline{\underline{T}}^{d-1} \left(\underline{\underline{T}}^{d} \underline{v}^{d} \right) \underline{v}^{d-1} \dots \right) \underline{v}^{1}$$

$$= \left(T_{(\ell'_{1}m'_{1}n'_{1})(\ell_{1}m_{1}n_{1})}^{1} v_{(\ell'_{1}m'_{1}n'_{1})}^{1} \right) \left(T_{(\ell_{1}m_{1}n_{1})(\ell'_{2}m'_{2}n'_{2})(\ell_{2}m_{2}n_{2})}^{2} v_{(\ell'_{2}m'_{2}n'_{2})}^{2} \right) \dots \left(T_{(\ell_{d-1}m_{d-1}n_{d-1})(\ell'_{d}m'_{d}n'_{d})}^{d} v_{(\ell'_{d}m'_{d}n'_{d})}^{d} \right).$$

$$(29)$$

We denote this representation as the equivariant tensor train (ETT) representation of a contraction of a general (d-dimensional) equivariant tensor with d covariant input feature vectors. Within the ETT representation, every coefficient tensor $\underline{\underline{C}}^i$ corresponding to an equivariant tensor $\underline{\underline{T}}^i$ has $\sum_{\ell,\ell',\ell'':TR(\ell,\ell',\ell'')} N_{\text{ett}}^{\text{rank}}(i,\ell) N(i,\ell') N_{\text{ett}}^{\text{rank}}(i,\ell'')$ angular momenta channels. As a boundary condition for α being a scalar, we require that $N_{\text{ett}}^{\text{rank}}(0,0) = N_{\text{ett}}^{\text{rank}}(d+1,0) = 1$. The invariance of α with respect to SO(3) follows from the properties of the equivariant tensors as discussed in the previous sections.

3.4.2 Invariance under O(3)

For our application of interatomic potentials, we also require reflection invariance of the multilinear forms α (29). To that end, we recall that, under space inversions $\mathcal{P}v_{(\ell mn)}(\widehat{\underline{r}}) = v_{(\ell mn)}(-\widehat{\underline{r}})$, we have

$$v_{(\ell m n)}(\widehat{\underline{r}}) = \mathcal{P}^{-1} v_{(\ell m n)}(\widehat{\underline{r}}) = (-1)^{\ell} v_{(\ell m n)}(-\widehat{\underline{r}}). \tag{30}$$

To ensure the invariance of α with respect to O(3), we require, according to the previous section, that $\ell'_1 + \ell'_2 + \ldots + \ell'_d$ is even. Recalling the definition of the 3-j symbol for RSHs, this immediately implies that α is always a *real* quantity since, in this case, the sum over all ℓ 's

$$(\ell'_1 + \ell_1) + (\ell_1 + \ell'_2 + \ell_2) + \dots + (\ell_{d-1} + \ell'_d) = \sum_{i=1}^d \ell'_i + \sum_{i=1}^{d-1} 2\ell_i$$
(31)

is obviously also even. This also implies that we can neglect imaginary parts for ETTs with <3 cores since we only need to consider even $\ell_1+\ell_2'+\ell_2$. For >3 cores, we need to keep track of both real and imaginary parts. Algorithms 1 and 2 are possible implementations of these two cases (therein, a tensor with superscripted "re" denotes its real part, whereas a tensor with superscripted "im" denotes its imaginary part).

Algorithm 1: ETT contraction with three or less cores

```
Input: Input feature vectors \underline{v}^i

1 \underline{u}^{d-1,\text{re}} \leftarrow \underline{\underline{T}}^{d,\text{re}}\underline{v}^d;

2 for i = d-1,\ldots,1 do

3 \left| \underline{u}^{i-1,\text{re}} \leftarrow \left(\underline{\underline{T}}^{i,\text{re}}\underline{u}^{i,\text{re}}\right)\underline{v}^i;

4 end

Output: u_0^{0,\text{re}}
```

There is another way to understand the real/imaginary splitting of reflection symmetric/antisymmetric components by simply noting that for the conventional complex-valued harmonics, written in the x,y,z coordinates (such that $x^2 + y^2 + z^2 = 1$), it holds that $Y_{\ell m}(x,y,z) = (Y_{\ell m}(x,-y,z))^*$, tracking which through our definitions of RSHs and the 3-j symbols yields an alternative proof.

4 Equivariant Tensor Network Numerical Algebra

The power of conventional tensor networks lies in the rich family of algorithms working with those. In this section we show that all the key algorithms manipulating with conventional tensor networks can be adapted to work with the equivariant tensor networks as well.

Algorithm 2: ETT contraction with four or more cores

Input: Input feature vectors
$$\underline{v}^{i}$$

1 $\underline{u}^{d-1,\mathrm{re}} \leftarrow \underline{\underline{T}}^{d,\mathrm{re}} \underline{v}^{d}$;

2 $\underline{u}^{d-2,\mathrm{re}} \leftarrow \left(\underline{\underline{T}}^{d-1,\mathrm{re}} \underline{u}^{d-1,\mathrm{re}}\right) \underline{v}^{d-1}$;

3 $\underline{u}^{d-2,\mathrm{im}} \leftarrow \left(\underline{\underline{T}}^{d-1,\mathrm{im}} \underline{u}^{d-1,\mathrm{re}}\right) \underline{v}^{d-1}$;

4 for $i = d-2, \ldots, 3$ do

5 $\underline{u}^{i-1,\mathrm{re}} \leftarrow \left(\underline{\underline{T}}^{i,\mathrm{re}} \underline{u}^{i,\mathrm{re}} + \underline{\underline{T}}^{i,\mathrm{im}} \underline{u}^{i,\mathrm{im}}\right) \underline{v}^{i}$;

6 $\underline{u}^{i-1,\mathrm{im}} \leftarrow \left(\underline{\underline{T}}^{i,\mathrm{re}} \underline{u}^{i,\mathrm{im}} + \underline{\underline{T}}^{i,\mathrm{im}} \underline{u}^{i,\mathrm{re}}\right) \underline{v}^{i}$;

7 end

8 $\underline{u}^{1,\mathrm{re}} \leftarrow \left(\underline{\underline{T}}^{2,\mathrm{re}} \underline{u}^{2,\mathrm{re}} + \underline{\underline{T}}^{2,\mathrm{im}} \underline{u}^{2,\mathrm{im}}\right) \underline{v}^{2}$;

9 $u_{0}^{0,\mathrm{re}} \leftarrow \left(\underline{\underline{T}}^{1,\mathrm{re}} \underline{u}^{1,\mathrm{re}}\right) \underline{v}^{1}$;

Output: $u_{0}^{0,\mathrm{re}}$

4.1 Trivial algorithms: gradient descent and alternating least squares

Although having a nontrivial structure, the evaluation of tensor networks reduces to sequential multiplication and addition of numbers, which can be effectively differentiated with respect to tensor core parameters using back propagation—hence, the gradient descent algorithm of optimizing tensor parameters (weights) can trivially (from the mathematical point of view) be formulated. A nontrivial generalization would be Riemannian gradient descent (i.e., the one with Riemannian gradient), but we leave this to future work.

Another trivially transferable algorithm is alternating least squares (ALS)—the tensor depends linearly on each of the cores $C_{(\ell_1 n_1)(\ell_2 n_2)(\ell_3 n_3)}$ and they can be independently optimized, each time solving a simple quadratic optimization problem.

We next come to more advanced algorithms, generalization of which to the equivariant case is less trivial.

4.2 QR or SVD normalization of cores

In the course of optimization, it is beneficial to orthogonalize the cores to avoid, e.g., that the weights unnecessarily grow during the optimization.

4.2.1 Order-2 cores

To that end, we consider two order-2 cores (cf., Figure 2)

$$\begin{split} \tilde{A}_{(\ell_1 m_1 n_1)(\ell_2 n_2 m_2)} &= A_{(\ell_1 n_1)(\ell_1 n_2)} \delta_{\ell_1 - \ell_2} \delta_{m_1 - m_2}, \quad \text{and} \\ \tilde{B}_{(\ell_1 m_1 n_1)(\ell_2 n_2 m_2)} &= B_{(\ell_1 n_1)(\ell_1 n_2)} \delta_{\ell_1 - \ell_2} \delta_{m_1 - m_2}, \end{split}$$

where δ is the Kronecker delta symbol. Notice that we have explicitly kept all three indices (ℓmn) for each tensor dimension and used the Kronecker delta to indicate which indices are effectively equal. Assuming these tensors are adjacent in the tensor network, they form another order-2

product

$$\begin{split} \tilde{C}_{(\ell_1 m_1 n_1)(\ell_3 n_3 m_3)} &= \sum_{\ell_2, m_2, n_2} A_{(\ell_1 n_1)(\ell_2 n_2)} \delta_{\ell_1 - \ell_2} \delta_{m_1 - m_2} B_{(\ell_2 n_2)(\ell_3 n_3)} \delta_{\ell_2 - \ell_3} \delta_{m_2 - m_3} \\ &= \delta_{\ell_1 - \ell_3} \delta_{m_1 - m_3} \sum_{\ell_2, m_2, n_2} \delta_{\ell_1 - \ell_2} \delta_{m_1 - m_2} A_{(\ell_1 n_1)(\ell_2 n_2)} B_{(\ell_2 n_2)(\ell_3 n_3)}. \\ &= \delta_{\ell_1 - \ell_3} \delta_{m_1 - m_3} \sum_{\ell_2, n_2} \delta_{\ell_1 - \ell_2} A_{(\ell_1 n_1)(\ell_2 n_2)} B_{(\ell_2 n_2)(\ell_3 n_3)}. \end{split}$$

What we see inside the sum is, effectively, a product of two block-diagonal matrices where different values of ℓ comprise independent blocks. We therefore have that

$$C_{(\ell n_1)(\ell n_3)} = \sum_{n_2} A_{(\ell n_1)(\ell n_2)} B_{(\ell n_2)(\ell n_3)}$$

for each ℓ . Orthogonalization is thus simple: we can, e.g., get a QR decomposition for $A_{(\ell \bullet)(\ell \bullet)} = QR$ and reassign the cores $A'_{(\ell \bullet)(\ell \bullet)} := Q$ and $B'_{(\ell \bullet)(\ell \bullet)} = RB_{(\ell \bullet)(\ell \bullet)}$ for each ℓ . The new A' and B' make up the same product C as the old ones, but now the rows of A are orthogonalized. Likewise, for each ℓ we can apply the SVD (singular value decomposition) $C_{(\ell \bullet)(\ell \bullet)} = UD^{\mathsf{T}}$ and assign $A'_{(\ell \bullet)(\ell \bullet)} = UD^{1/2}$ and $B'_{(\ell \bullet)(\ell \bullet)} = D^{1/2}V^{\mathsf{T}}$. In this case both, rows of A' and columns of B', will be orthogonal.

(a) The order-2 tensor is split into a coefficient matrix indexed by (ℓ,n) and a Kronecker delta in m

(b) The (ℓ,n) -indexed matrix is decomposed into L diagonal blocks each indexed by n

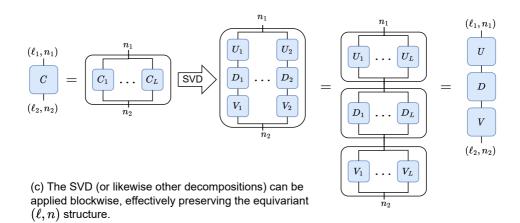


Figure 2: Diagrammatic illustration of concepts in and operations on order-2 tensors.

4.2.2 Order-3 cores

We now consider two adjacent order-3 cores. This is a harder case and requires some preparation. Consider an order-3 tensor

$$\tilde{A}_{(\ell_1 m_1 n_1)(\ell_2 m_2 n_2)(\ell_3 m_3 n_3)} = A_{(\ell_1 n_1)(\ell_2 n_2)(\ell_3 n_3)} \begin{cases} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{cases}$$

and the corresponding trilinear form

$$\alpha = A_{(\ell_1 n_1)(\ell_2 n_2)(\ell_3 n_3)} \left\{ \begin{matrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{matrix} \right\} u_{(\ell_1 m_1 n_1)} v_{(\ell_2 m_2 n_2)} w_{(\ell_3 m_3 n_3)}.$$

We later contract this tensor along the third dimension while the first two will remain uncontracted. This motivates us to reexpand $u_{(\ell_1 m_1 n_1)} v_{(\ell_2 m_2 n_2)}$ as

$$U_{(\ell m(\ell_1\ell_2n_1n_2))} = \sum_{m_1,m_2} \left\{ \begin{matrix} \ell_1 & \ell_2 & \ell \\ m_1 & m_2 & m \end{matrix} \right\} u_{(\ell_1m_1n_1)} v_{(\ell_2m_2n_2)}.$$

We fix ℓ_1 and ℓ_2 , and let ℓ go from $|\ell_1 - \ell_2|$ to $\ell_1 + \ell_2$. We then introduce an auxiliary ℓ_1, ℓ_2 -indexed family of matricies

$$(J_{\ell_1,\ell_2})_{(\ell m)(m_1m_2)} := \left\{ \begin{matrix} \ell_1 & \ell_2 & \ell \\ m_1 & m_2 & m \end{matrix} \right\}.$$

The first dimension of each of these matrices is ℓm and the second one is $m_1 m_2$. The size of each matrix is exactly $(2\ell_1+1)(2\ell_2+1)\times(2\ell_1+1)(2\ell_2+1)$. We denote its inverse as $(J_{\ell_1,\ell_2})^{-1}$ (it exists since the 3-j coefficients form a one-to-one mapping) and hence express

$$u_{(\ell_1 m_1 n_1)} v_{(\ell_2 m_2 n_2)} = (J_{\ell_1, \ell_2})_{(m_1 m_2)(\ell m)}^{-1} U_{(\ell m (\ell_1 \ell_2 n_1 n_2))}.$$

Our trilinear form α , after substitution of U instead of u and v becomes a bilinear form

$$\begin{split} \alpha &= A_{(\ell_1 n_1)(\ell_2 n_2)(\ell_3 n_3)} \left\{ \begin{aligned} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{aligned} \right\} (J_{\ell_1,\ell_2})_{(m_1 m_2)(\ell m)}^{-1} U_{(\ell m(\ell_1 \ell_2 n_1 n_2))} w_{(\ell_3 m_3 n_3)} \\ &= A_{(\ell_1 n_1)(\ell_2 n_2)(\ell_3 n_3)} (J_{\ell_1,\ell_2})_{(\ell_3 m_3)(m_1 m_2)} (J_{\ell_1,\ell_2})_{(m_1 m_2)(\ell m)}^{-1} U_{(\ell m(\ell_1 \ell_2 n_1 n_2))} w_{(\ell_3 m_3 n_3)} \\ &= A_{(\ell_1 n_1)(\ell_2 n_2)(\ell_3 n_3)} \delta_{\ell - \ell_3} \delta_{m - m_3} U_{(\ell m(\ell_1 \ell_2 n_1 n_2))} w_{(\ell_3 m_3 n_3)}. \end{split}$$

The last expression should remind us of the order-2 tensor that we have considered in the last subsection. In other words, if we denote

$$\hat{A}_{(\ell_3(\ell_1\ell_2n_1n_2))(\ell_3n_3)} := A_{(\ell_1n_1)(\ell_2n_2)(\ell_3n_3)},$$

it is the coefficient matrix of the order-2 tensor.

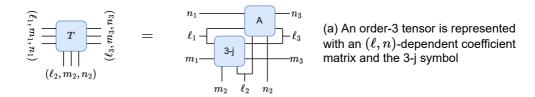
This motivates us to consider the following definition of reshaping an order-3 tensor to an order-2 tensor:

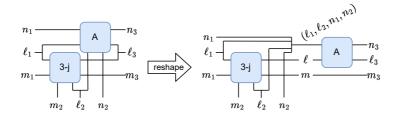
$$\left(\mathrm{reshape}_{((1,2),3)} A \right)_{(\ell(\ell_1\ell_2n_1n_2))\,(\ell n)} := A_{(\ell_1n_1)(\ell_2n_2)(\ell n)},$$

where the index ((1,2),3) indicates that we are merging dimension 1 and 2 together.

The rest is relatively straightforward. If the tensor $\tilde{A}_{(\ell_1 m_1 n_1)(\ell_2 m_2 n_2)(\ell_3 m_3 n_3)}$ is contracted with another tensor $\tilde{B}_{(\ell_3 m_3 n_3)(\ell_4 m_4 n_4)(\ell_5 m_5 n_5)}$, we reshape the latter's coefficients to $\hat{B}_{(\ell_3 n_3)(\ell_3 (\ell_4 \ell_5 n_4 n_5))}$ and proceed as in Section 4.2.1 for order-2 tensors. That is, for the QR decomposition we let, independently for each ℓ_3 , $\hat{A} = QR$, and assign $\hat{A}' := Q$ and $\hat{B}' := R\hat{B}$. Likewise, for SVD we decompose $\hat{A}\hat{B} = UDV^{\mathsf{T}}$ and assign $\hat{A}' := UD^{1/2}$ and $\hat{B}' := D^{1/2}V^{\mathsf{T}}$. After setting the new cores \hat{A}' and \hat{B}' we trivially reshape them back to the sought order-3 cores A' and B'.

The ideas presented above are illustration in Figure 3. The case of contracting an order-3 tensor with an order-2 tensor is, in the view of the above, is straightforward: we need to reshape the order-3 tensor and keep it as an order-2 tensor.





(b) The 3-j symbol can be interpreted as reducing the double $(\ell_1,m_1,n_1),(\ell_2,m_2,n_2)$ input into a single one (ℓ,m,n) with $n=(\ell_1,\ell_2,n_1,n_2)$, allowing for reshaping the coefficient matrix into an order-2 matrix

Figure 3: Diagrammatic illustration of concepts in and operations on order-3 tensors.

Remark 2. Assuming a QR decomposition of $\hat{A}\hat{B}$ such that $\hat{A} = Q$, an interesting property that follows directly from the orthogonality of \hat{A} is the unitarity (more precisely, left-unitarity) of the associated equivariant tensor A. To see this, let

$$\begin{split} A^{\mathsf{T}*}A &:= A_{(\ell(\ell_1\ell_2m_1m_2n_1n_2))(\ell mn)} A_{(\ell(\ell_1\ell_2m_1m_2n_1n_2))(\ell'm'n')} \\ &= \sum_{\substack{\ell_1, m_1, n_1, \\ \ell_2, m_2, n_2}} \hat{A}_{(\ell(\ell_1\ell_2n_1n_2))(\ell n)} \left\{ \begin{matrix} \ell_1 & \ell_2 & \ell \\ m_1 & m_2 & m \end{matrix} \right\}^* \hat{A}_{(\ell(\ell_1\ell_2n_1n_2))(\ell'n')} \left\{ \begin{matrix} \ell_1 & \ell_2 & \ell' \\ m_1 & m_2 & m' \end{matrix} \right\} \\ &= \sum_{\substack{\ell_1, n_1, \\ \ell_2, n_2}} \hat{A}_{(\ell(\ell_1\ell_2n_1n_2))(\ell n)} \hat{A}_{(\ell(\ell_1\ell_2n_1n_2))(\ell'n')} \sum_{m_1, m_2} \left\{ \begin{matrix} \ell_1 & \ell_2 & \ell \\ m_1 & m_2 & m \end{matrix} \right\}^* \left\{ \begin{matrix} \ell_1 & \ell_2 & \ell' \\ m_1 & m_2 & m' \end{matrix} \right\}, \\ \begin{pmatrix} (\mathsf{A}23) \\ = \delta_{\ell\ell'} \delta_{mm'} \sum_{\ell_1, n_1, \\ \ell_2, n_2} \hat{A}_{(\ell(\ell_1\ell_2n_1n_2))(\ell n)} \hat{A}_{(\ell(\ell_1\ell_2n_1n_2))(\ell'n')} \\ &= \delta_{\ell\ell'} \delta_{mm'} \delta_{nn'} \end{split}$$

where we have adsorbed the prefactor $(2\ell+1)$ from (A23) into \hat{A} . Hence, there appears to us no obstacle in extending convergence results for the usual TT-ALS algorithm to the equivariant case; the unitarity property (32) hints that an ETT-ALS algorithm that is stabilized using QR decompositions should be guaranteed to converge (cf., Theorem 1 in [34]).

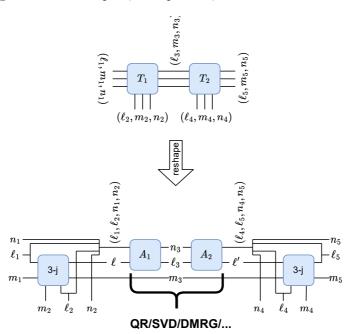
Remark 3 (a potentially more efficient version of SVD-based orthogonalization). The dimension $\ell_3(\ell_1\ell_2n_1n_2)$ of the reshaped matrix may be rather large. Instead of applying SVD to the product $\hat{A}\hat{B}$ which can be rather large, one can consider an SVD applied separately to $\hat{A} = U_1D_1V_1^\mathsf{T}$, $\hat{B} = U_2D_2V_2^\mathsf{T}$, in which case $\hat{A}\hat{B} = U_1(D_1V_1^\mathsf{T}U_2D_2)V_2^\mathsf{T}$. The expression in the parenthesis is a

square matrix with reduced dimension $(\ell_3 n_3)$ which itself can be decomposed as $(D_1 V_1^\mathsf{T} U_2 D_2) = U_3 D_3 V_3^\mathsf{T}$ so that the final SVD can be obtained as $\hat{A}\hat{B} = U_1 U_3 D_3 V_3^\mathsf{T} V_2^\mathsf{T}$.

4.3 DMRG algorithm

Reshaping equivariant tensors opens the way to applying a number of algorithms that are standard for the conventional tensor networks. For instance, the Density Matrix Renormalization Group (DMRG) algorithm [26], [35], applied to the problem of minimizing a quadratic functional of the multidimensional tensor, can now be easily formulated.

Indeed, if we have two adjacent cores A and B (like in Section 4.2.2), we can reshape and combine these cores into a large matrix $\hat{A}\hat{B}$, with respect to which, with all other cores fixed, the optimization problem is quadratic. We can hence solve this quadratic optimization problem, apply the SVD algorithm to decompose, reshape back, and this forms the new cores A' and B'.



Reshaping two adjacent order-3 tensors into order-2 tensors opens possibilities of transfering standard tensornetwork algorithms onto the equivariant tensor networks

Figure 4: Diagrammatic illustration of how one can conduct QR, SVD, DMRG, or other algorithms on two adjacent order-3 equivariant tensors. They are reduced (reshaped) to the product of two order-2 tensors, operations on which (e.g., matrix multiplication) can be done block-wise with respect to ℓ .

For example, for a tensor-train representation of $T_{\langle d \rangle}$ given by

$$T_{(\ell'_1m'_1n'_1)\dots(\ell'_dm'_dn'_d)} = \sum_{(\ell_1m_1n_1)\dots} \dots \tilde{T}^i_{(\ell_{i-1}m_{i-1}n_{i-1})(\ell'_im'_in'_i)(\ell_im_in_i)}$$

$$\tilde{T}^{i+1}_{(\ell_im_in_i)(\ell'_{i+1}m'_{i+1}n'_{i+1})(\ell_{i+1}m_{i+1}n_{i+1})} \dots,$$

(compared to (29) our (ℓmn) -dependent cores have tildes above them to be consistent with the rest of the notations in this section) the optimization problem $\mathcal{F}(T) \to \min$ when all cores except for $\tilde{T}^{(i-1)}$ and $\tilde{T}^{(i)}$ are fixed, reduces to the problem schematically written as

$$\mathcal{F}\big(\dots \tilde{T}^{i}_{(\ell_{i-1}m_{i-1}n_{i-1})(\ell'_{i}m'_{i}n'_{i})(\ell_{i}m_{i}n_{i})}\tilde{T}^{i+1}_{(\ell_{i}m_{i}n_{i})(\ell'_{i+1}m'_{i+1}n'_{i+1})(\ell_{i+1}m_{i+1}n_{i+1})}\dots\big).$$

The coefficients of these two tensors are reshaped to the matrices $\hat{T}^i_{(\ell'(\ell_{i-1}\ell_i n_{i-1}n_i))(\ell'n')}$ and $\hat{T}^{i+1}_{(\ell'n')(\ell'(\ell_i\ell_{i+1}n_i n_{i+1}))}$ and then \mathcal{F} is quadratic in terms of $C:=\hat{T}^i\hat{T}^{i+1}$. It is hence solved, decomposed into UDV^T , and then the updated cores are set as $\hat{T}^i=UD^{1/2}$ and $\hat{T}^{i+1}=D^{1/2}V^\mathsf{T}$.

The idea of how the DMRG (and other) algorithms can be applied to the two adjacent tensor network cores is diagrammatically illustrated in Figure 4.

5 Application to Interatomic Potentials

We now use ETNs to construct interatomic potentials. We present our ETN potentials with the classical way using formulas defining the features, then the functional form, etc., as well as the tensor network diagram notation that we have developed in the previous sections. This allows us to easier understand the proposed functional forms, but also to compare those of different potentials. In particular, we present tensor network diagrams for several state-of-the-art MLIPs in Appendix B, including MTP, SNAP, and ACE.

5.1 Equivariant Tensor Train Representation of Per-Atom Energies

Before applying our ETN formalism, we first fix some additional notation: in the following, we denote a configuration of atoms \underline{r}_i by $\{\underline{r}_i\}$, and a neighborhood of an atom \underline{r}_i by $\{\underline{r}_{ij}\}$. We then define the input feature vectors more explicitly as functions of atomic neighborhoods $\{\underline{r}_{ij}\}$

$$v_{(\ell mn)} = v_{(\ell mn)}(\{\underline{r}_{ij}\}) = \sum_{j} \dot{Y}_{lm}(\widehat{\underline{r}}_{ij}) f_n, \tag{33}$$

where $\hat{\underline{r}}_{ij}$ is the angular contribution to \underline{r}_{ij} , and f_n is a vector of pair-wise non-angular features that we will define momentarily. We can then write the per-atom energy as follows

$$\mathcal{E} = \mathcal{E}(\{\underline{r}_{ij}\}) = \underline{\underline{T}}^{1} \Big(\dots \underline{\underline{T}}^{d-1} \Big(\underline{\underline{T}}^{d} \underline{\underline{v}}\Big) \underline{\underline{v}} \dots \Big) \underline{\underline{v}}.$$
 (34)

The total energy of a configuration $\{\underline{r}_i\}$ is then given as follows

$$\Pi = \Pi(\{\underline{r}_i\}) = \sum_{i} \mathcal{E}(\{\underline{r}_{ij}\}). \tag{35}$$

Remark 4. In fact, we consider the vectors $\underline{v}' = (1, \underline{v}^{\mathsf{T}})^{\mathsf{T}}$ as input vectors to the ETT in order to generate complete polynomials. We omit this detail as it would require some tedious notation, which is not essential to explain the basic construction of the potentials.

5.2 Non-Angular Features

In the following, we consider multicomponent systems with $N_{\rm spec}$ atomic species. That is, the feature vectors do not only depend on the atomic positions in the neighborhood of an atom i but also on the species of the i-th atom \underline{z}^i and the species of all atoms \underline{z}^j in its neighborhood. We assume a lexicographical order of the atomic species $s = \{0, 1, 2, \ldots, N_{\rm spec} - 1\}$ and define

$$z_{\beta}^{i} = \delta_{\beta s_{i}}, \qquad \qquad z_{\gamma}^{j} = \delta_{\beta s_{i}}, \qquad (36)$$

where s_i and s_j are the species of the *i*-th and *j*-th atoms, respectively. We may then write the non-angular feature vectors as

$$f_n = f_{(\mu\beta\gamma)}(\|\underline{r}_{ij}\|, \underline{z}^i, \{\underline{z}^j\}) = Q_{\mu}(\|\underline{r}_{ij}\|) z_{\beta}^i z_{\gamma}^j, \tag{37}$$

where $Q_{\mu}(\|\underline{r}_{ij}\|)$ are radial basis functions. The corresponding feature vectors then read

$$v_{(\ell m \mu \beta \gamma)}(\{\underline{r}_{ij}\}, \underline{z}^i, \{\underline{z}^j\}) = \sum_{j} \dot{Y}_{lm}(\widehat{\underline{r}}_{ij}) Q_{\mu}(\|\underline{r}_{ij}\|) z_{\beta}^i z_{\gamma}^j.$$
(38)

A trilinear form of type (24) with such \underline{v} 's is then given by

$$\alpha = \alpha(\underline{u}, \underline{v}, \underline{w}) = T_{(\ell_1 m_1 n_1)(\ell_2 m_2 \mu \beta \gamma)(\ell_3 m_3 n_3)} u_{(\ell_1 m_1 n_1)} v_{(\ell_2 m_2 \mu \beta \gamma)} w_{(\ell_3 m_3 n_3)}. \tag{39}$$

5.3 Equivariant Tensor Network Potentials

One problem that arises when defining a potential energy as done in the previous sections is the growth of the parameter space when the number of features becomes large. Even considering only radial and species features can already lead to a very large parameter space, e.g., for systems with many components, such as high-entropy alloys. To address this problem, we propose in the following a contraction of features before entering the ETT, leading to equivariant tensor network (ETN) potentials.

In order to illustrate this idea, we consider a contraction of only non-angular features and leave the spherical harmonic basis untouched. To that end, we reshape the input feature vectors $v_{(\ell m \mu \beta \gamma)}$ into three-dimensional covariant tensors

$$F_{\ell m\mu(\beta\gamma)} = \sum_{i} \dot{Y}_{\ell m}(\widehat{r}_{ij}) Q_{\mu}(\|\underline{r}_{ij}\|) z_{\beta}^{i} z_{\gamma}^{j}. \tag{40}$$

We then contract μ and $(\beta \gamma)$ before entering the ETT, that is, we define the *contracted feature* vectors as

$$v_{(\ell m n)} = B_{\ell n \alpha \lambda} A_{\ell \lambda (\beta \gamma)} F_{\ell m \alpha (\beta \gamma)}$$

$$= \sum_{j} \dot{Y}_{\ell m}(\widehat{\underline{r}}_{ij}) \left(B_{\ell n \alpha \lambda} Q_{\alpha}(\|\underline{r}_{ij}\|) \left(A_{\ell \lambda (\beta \gamma)} z_{\beta}^{i} z_{\gamma}^{j} \right) \right), \tag{41}$$

with the ℓ -dependent coefficient tensors

$$\underline{\underline{\underline{A}}}_{\ell} \in \mathbb{R}^{N_{\text{spec}}^{\text{rank}}(\ell) \times N_{\text{spec}}^2} \quad \text{and} \quad \underline{\underline{\underline{B}}}_{\ell} \in \mathbb{R}^{N_{\text{rad}}^{\text{rank}}(\ell) \times N_{\text{rad}}(\ell) \times N_{\text{spec}}^{\text{rank}}(\ell)}. \tag{42}$$

We remark here that we never explicitly compute $A_{\lambda(\beta\gamma)}z^i_{\beta}z^j_{\gamma}$ but rather take the $(z^i_{s_i}z^j_{s_j})$ -th column of the matrix $\underline{\underline{A}}$, since this is the only nonzero entry in $\underline{z}^i \otimes \underline{z}^j$ according to (36). Algorithm 3 summarizes our implementation of computing \underline{v} .

Algorithm 3: Computation of the contracted feature vectors for the ETN potential (43)

```
Input: Neighborhood \{\underline{r}_{ij}\} of an atom \underline{r}_i, species features \underline{z}^i, \{\underline{z}^j\}

1 \underline{v} \leftarrow \underline{0};

2 for all \underline{r}_{ij} \in \{\underline{r}_{ij}\} do

3 | for \ell = 0, \dots, L do

4 | \underline{a} \leftarrow \underline{A}_{\ell(z^i_{s_i}z^j_{s_j})};

5 | \underline{b} \leftarrow (\underline{\underline{B}}_{\ell}\underline{a})\,\underline{Q}(||\underline{r}_{ij}||);

6 | \underline{v}_{\ell} \leftarrow \underline{v}_{\ell} + \mathrm{reshape}_{(1,2)}\,\big(\underline{Y}_{\ell}(\widehat{r}_{ij}) \otimes \underline{b}\big);

7 | end

8 end

Output: \underline{v}
```

In the tensor network diagram notation, the per-atom energy then reads

$$\mathcal{E}(\underline{\underline{F}}) = \begin{pmatrix} A_{\ell_{1}} & A_{\ell_{2}} & A_{\ell_{d}} \\ (\beta_{1}\gamma_{1}) \lambda_{1} & (\beta_{2}\gamma_{2}) \lambda_{2} & (\beta_{d}\gamma_{d}) \lambda_{d} \\ F_{\ell_{1}'} & \alpha_{1} - B_{\ell_{1}'} & F_{\ell_{2}'} & \alpha_{2} - B_{\ell_{2}'} \\ m'_{1} & n'_{1} & m'_{2} & n'_{2} & m'_{d} & n'_{d} \\ (\ell'_{1}m'_{1}n'_{1}) & (\ell'_{2}m'_{2}n'_{2}) & (\ell'_{d}m'_{d}n'_{d}) \\ \sum_{1} & \sum_{1} &$$

This is, of course, not the only possibility to construct a tensor network potential. For example, if the number of atomic species becomes large, one may envision a separation of \underline{z}^i and \underline{z}^j leading to

$$\mathcal{E}(F_{\langle 4 \rangle}) = \begin{array}{c} A_{\ell_{1}'} & A_{\ell_{2}'} \\ A_{\ell_{1}'} & A_{\ell_{2}'} \\ A_{\ell_{1}'} & A_{\ell_{2}'} \\ A_{\ell_{2}'} & A_{\ell_{d}'} \\ A_{\ell_{d}'} & A_{\ell_{$$

We call the T^i tensors in (43) and (44) ETT cores and the entire lower part the ETT.

We emphasize that we have used weight sharing between ETT cores for the two ETN potentials defined above, i.e., the parameter tensors $\underline{\underline{A}}, \underline{\underline{B}}$, and $\underline{\underline{C}}$, do not depend on d—as opposed to $\underline{\underline{T}}$. We could have likewise used a core-dependent feature contraction, which could simplify the training since the ETNs then linearly depend on each parameter tensor (refer to our remarks in Section 7).

5.4 Hyperparameters of Tensor Network Potentials

MLIPs contain a number of hyperparameters, such as the cut-off radius, the choice of the radial basis, etc. In the following, we list the hyperparameters that are specific to ETN potentials and do not occur in other MLIPs. The ETN-specific hyperparameters are related to the ranks of the tensors $\underline{\underline{A}}$, $\underline{\underline{B}}$, and $\underline{\underline{T}}$, as shown in Table 1. All tensorial dimensions are defined per core i and angular momentum ℓ . We remark that the ETT dimension d and the total number of angular momenta for the feature vectors and ETT ranks are also hyperparameters that define the body-order and the polynomial degree of the ETN potential.

Tensor	Dimensions
$\underline{\underline{A}}_{\ell'}$	$N_{ m spec}^{ m rank}(\ell') \times N_{ m spec} \times N_{ m spec}$
$\underline{\underline{B}}_{\ell'}$	$N_{\mathrm{rad}}^{\mathrm{rank}}(\ell') \times N_{\mathrm{rad}}(\ell') \times N_{\mathrm{spec}}^{\mathrm{rank}}(\ell')$
$\boxed{\frac{T^{i}}{\equiv}(\ell m)(\ell' m')(\ell'' m'')}$	$N_{ m ett}^{ m rank}(i,\ell) imes N_{ m rad}^{ m rank}(\ell') imes N_{ m ett}^{ m rank}(i,\ell'')$

Table 1: Tensorial dimensions for the ETN potential (43)

6 Performance of Equivariant Tensor Network Potentials

6.1 Optimization of the Hyperparameters

In order to identify the "best" hyperparameters for the ETN potentials, i.e., the hyperparameters that are close-to the Pareto front that minimizes some function with respect to the number of fitting parameters, we propose a *stochastic gradient descent algorithm*. The function to be minimized could be, e.g., the validation loss, the root-mean-square error of per-atom energies, or some other quantity of interest. Since the total number of hyperparameters becomes large when the ETT dimension and the total number of angular momenta become large, we first place some assumptions based on our "educated guess" for the selection of the hyperparameters:

- The total number of angular momenta of the ETT ranks is constant over all cores and, moreover, equal to the total number of angular momenta L of the feature vectors.
- The total number of angular momenta channels of the ETT ranks are constant over all cores, i.e., $N_{\rm ett}^{\rm rank}(i,\ell) = N_{\rm ett}^{\rm rank}(\ell)$.
- Any number of angular momentum channels $N(\ell)$ decays exponentially with increasing ℓ according to the function

$$N(\ell) = (N(0) - 1)e^{-\ell/a} + 1 \tag{45}$$

rounded to the nearest integer, where a specifies the decay. Specifically, we set a=1 for $N_{\rm rad}(\ell)$, and a=5 for $N_{\rm spec}^{\rm rank}(\ell)$, $N_{\rm ett}^{\rm inp}(\ell)$, and $N_{\rm ett}^{\rm rank}(\ell)$. So, the number of radial functions is assumed to decay slowly with ℓ , while all the ranks decay fast with increasing ℓ .

Based on the previous assumptions we are left with the following six hyperparameters as degrees of freedom:

- the maximum number of radial functions $N_{\rm rad} = N_{\rm rad}(0)$,
- the order/dimension d of the ETT,
- the maximum number of angular momenta L,
- the maximum species rank $N_{\mathrm{spec}}^{\mathrm{rank}} = N_{\mathrm{spec}}^{\mathrm{rank}}(0)$,
- the maximum radial rank $N_{\mathrm{rad}}^{\mathrm{rank}} = N_{\mathrm{rad}}^{\mathrm{rank}}(0)$,
- the maximum ETT rank $N_{\mathrm{ett}}^{\mathrm{rank}} = N_{\mathrm{ett}}^{\mathrm{rank}}(0)$.

We thus define the set of hyperparameters as

$$\mathcal{H} = \left\{ N_{\text{rad}}, d, L, N_{\text{spec}}^{\text{rank}}, N_{\text{rad}}^{\text{rank}}, N_{\text{ett}}^{\text{rank}} \right\}. \tag{46}$$

Given an ETN potential with hyperparameters \mathcal{H} , we then define the loss function

$$\mathscr{L}(\mathscr{H}, \underline{\theta}, \mathscr{T}) = \frac{1}{\#\mathscr{T}} \sum_{\{\underline{r}_i\} \in \mathscr{T}} \left(w_{\mathrm{e}} \Big(\Pi(\{\underline{r}_i\}) - \Pi^{\mathrm{qm}}(\{\underline{r}_i\}) \Big)^2 + w_{\mathrm{f}} \sum_{\underline{r}_i \in \{\underline{r}_i\}} \|\underline{f}_{r_i} - \underline{f}_{r_i}^{\mathrm{qm}}\|^2 \right), \quad (47)$$

where $\#\mathscr{T}$ is the size of the training set, $\underline{\theta}$ is the vector of fitting parameters, \mathscr{T} is the dataset set which may contain quantum-mechanical energies $\Pi^{\rm qm}$ and forces $\underline{f}^{\rm qm}$ of a configuration, and $w_{\rm e}$ and $w_{\rm f}$ are weighting parameters. We assume in the following a training set $\mathscr{T}^{\rm train}$ and a validation set $\mathscr{T}^{\rm valid}$. We exemplify the optimization of the hyperparameters by minimizing the mean loss of $\mathscr{T}^{\rm valid}$ with respect to \mathscr{H} . Our algorithm is then as follows:

- [S1] We start with a pair potential with five radial basis functions and minimal ranks, i.e., $\mathcal{H} = \{5, 1, 1, 1, 1, 1\}.$
- [S2] We then minimize the loss of $\mathscr{T}^{\text{train}}$ N times using different random initializations of the parameters $\underline{\theta}$ and compute the mean of the validation loss

$$\bar{\mathcal{L}}^{\text{valid}}(\mathcal{H}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathcal{H}, \underline{\theta}_i, \mathcal{T}^{\text{valid}}). \tag{48}$$

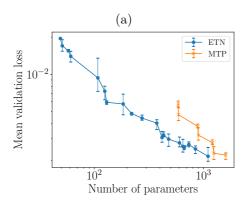
- [S3] We then increase each of the hyperparameters by 1 individually and denote these updates by $\mathcal{H} + \Delta_i \mathcal{H}$ (i = 1, ..., 6). For each hyperparameter update we minimize the training loss N times and compute the mean validation losses $\mathcal{L}^{\text{valid}}(\mathcal{H} + \Delta_i \mathcal{H})$.
- [S4] For each hyperparameter update, we compute the stochastic gradient of the mean validation loss as follows

$$\nabla_{i}\bar{\mathcal{L}}^{\text{valid}} = -\frac{\log \bar{\mathcal{L}}^{\text{valid}}(\mathcal{H} + \Delta_{i}\mathcal{H}) - \log \bar{\mathcal{L}}^{\text{valid}}(\mathcal{H})}{\log \#\underline{\theta}(\mathcal{H} + \Delta_{i}\mathcal{H}) - \log \#\underline{\theta}(\mathcal{H})},\tag{49}$$

where $\#\underline{\theta}(\mathcal{H})$ and $\#\underline{\theta}(\mathcal{H} + \Delta_i \mathcal{H})$ are the number of parameters before and after the *i*-th hyperparameter update. Note that we use logarithms to adjust the order of $\mathcal{L}^{\text{valid}}$ and $\#\underline{\theta}$.

[S5] We now increase \mathcal{H} by the $\Delta_i \mathcal{H}$ that maximizes (49) and return to [S3].

We repeat the steps [S3]-[S5] until the mean validation loss is as small as we want it to be.



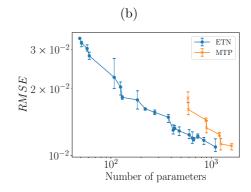


Figure 5: (a) Mean validation loss, and (b) RMSE, as a function of the number of parameters for QM7. The horizontal bars correspond to the minimum and maximum values

6.2 Numerical Examples

We test the algorithm on various datasets for multicomponent systems from the literature. For the optimization of the loss function, we use python's plain BFGS solver. During optimization, we monitor the validation loss. After 1000 iterations, we stop the optimization and select the minimal validation loss from all the iterations. We repeat the optimization three times to compute the mean validation loss. For the weights we choose

$$w_{\rm e} = 1 \,\text{eV}^2, \qquad w_{\rm f} = 0.01 \,(\text{eV/Å})^2.$$
 (50)

Further, for all computational results, presented in the following, we use a cut-off radius of 5 Å, if not specified otherwise.

We test our algorithm on various datasets from the literature for multicomponent systems and compare the performance of the ETN potentials to moment tensor potentials (MTPs) in terms of the required number of parameters; we refer to Appendix B.1 for the functional form of MTPs. We will analyze the error in the validation loss and the root-mean-square error for the per-atom energies and forces, referred to as RMSE and RMSF in the following.

6.2.1 QM7

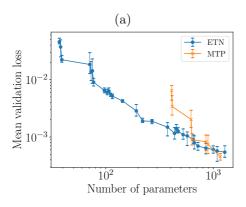
The QM7 dataset [36], [37] contains 7211 small molecules with up to 23 atoms and six atomic species (C, N, O, S, Cl, and H) in the their equilibrium position. We select 6186 configurations for the training set, and 1025 configurations for the validation set. We fit to total energies.

The decay of the mean validation loss, computed using the stochastic gradient descent algorithm, is shown in Figure 5 (a). In general, the ETN potentials require much fewer parameters than MTPs for a comparable level of accuracy, especially the ETN potentials with lower complexity.

6.2.2 QM9

The QM9 dataset [38], [39] contains 134k small organic molecules with up to five atomic species (C, H, O, N, F) in the their equilibrium position. We select a random subset of 20k configurations from the full dataset. From these 20k configurations, we use 17k configurations for the training set and 3k configurations for the validation set. We fit to total energies.

As for QM7, ETN potentials with lower complexity require much fewer parameters than MTPs. However, when approaching the realm of higher accuracy (RMSE < 10 meV/atom, cf.,



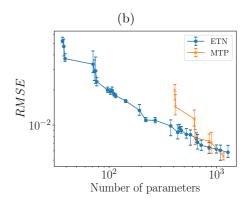


Figure 6: (a) Mean validation loss, and (b) RMSE, as a function of the number of parameters for QM9. The horizontal bars correspond to the minimum and maximum values

Figure 6 (b)), there is no difference anymore between ETN potentials and MTPs. Presently, we attribute this to python's BFGS solver not being the optimal choice for training ETN potentials. Indeed, we have observed premature convergence (after ~ 50 iterations) for approximately 50 % of the training runs per iteration when the ETN potentials reach a higher complexity (i.e., from $\sim 400-500$ coefficients). On the other hand, python's BFGS is also not optimal for MTPs, but MTPs already have a more mature functional form that appears to perform better on QM9.

6.2.3 MoNbTaW Medium-Entropy Alloy

This dataset has been developed to construct a spectral neighbor analysis potential (SNAP) for the MoNbTaW medium-entropy alloy [40]. Hence, we refer to this dataset as MoNbTaW-MEA in the following. The dataset contains 5529 configurations containing various configuration types, e.g., configurations with free surfaces, snapshots from molecular dynamics simulations at finite temperature, etc. We use 4983 configurations for the training set and 546 for the validation set (the validation set is constructed such that every configuration type is represented). We fit to total energies and forces.

Over the entire range of potentials, the amount of coefficients is universally lower than the amount of coefficients for MTPs when both potentials give similar accuracy, as shown in Figure 7 (a) and (b). Even when approaching the realm of high accuracy $(RMSE \sim 5\text{--}6\,\text{meV/atom}, RMSF \sim 150\text{--}180\,\text{meV/Å})$, ETN potentials require 2–3 times less parameters than MTPs. We attribute this excellent performance to the ability of ETN potentials to learn similarities between atomic species by varying the species rank $N_{\text{spec}}^{\text{rank}}$ since all atoms belong to the group of transition metals.

6.2.4 MoNbTaVW High-Entropy Alloy

This dataset, henceforth referred to as MoNbTaVW-HEA, appears to us the first dataset generated for a high-entropy alloy (five components or more) and is more general than the MoNbTaW-MEA in the sense that it contains a more diverse set of configurations, including various defects (vacancies, di- and tri-vacancies, self-interstitial atoms, stacking faults) and liquid states [41]. The whole dataset contains 2859 configurations. The dataset also contains many out-of-equilibrium configurations with very large forces (up to $\sim 150\,{\rm eV/\mathring{A}}$), such as dimers. We have removed configurations that contain forces $> 5\,{\rm eV/\mathring{A}}$ leading to a subset of the full dataset containing 1423 configurations. From this subset we use 1210 configurations for training, and 213 for validation. We fit to total energies and forces.

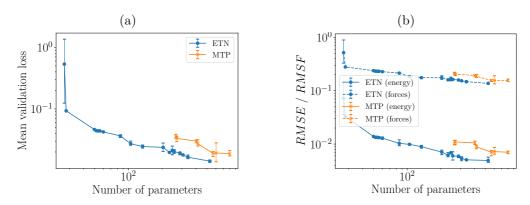


Figure 7: (a) Mean validation loss, and (b) RMSE/RMSF, as a function of the number of parameters for MoNbTaW-MEA. The horizontal bars correspond to the minimum and maximum values

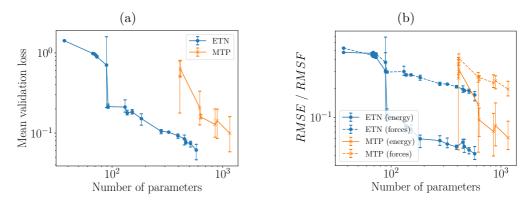


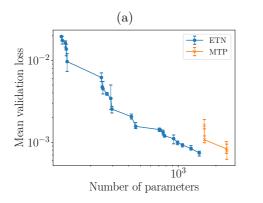
Figure 8: (a) Mean validation loss, and (b) RMSE/RMSF, as a function of the number of parameters for MoNbTaVW-HEA. The horizontal bars correspond to the minimum and maximum values

From Figure 8, we observe that the performance of ETN potentials is similar to MoNbTaW-MEA. In comparison to MTPs, ETN potentials require many times less coefficients to reach a comparable mean validation loss. We yet remark that, for MoNbTaVW-HEA, it now appears to be more difficult to fit MTPs as the fluctuation in the loss function is much larger than for ETN potentials. Nevertheless, even when comparing the *mean* validation loss of ETN potentials and the *minimum* validation loss of MTPs, ETN potentials still require 2–3 times less coefficients when both potentials give comparable accuracy.

6.2.5 Binary Alloys with 10 different Species

We now test the performance of ETN potentials on a dataset containing 10 binary alloys [42], with 10 different atomic species in total (AgCu, AlFe, AlMg, AlNi, AlTi, CoNi, CuFe, CuNi, FeV, and NbNi). We refer to it as BA10 in the following. BA10 contains all possible fcc, bcc, and hcp structures, with up to eight atoms in the unit cell. In total, BA10 contains 15950 configurations. We select 13558 configurations that serve as the training set, the remaining 2392 configurations serve as the validation set. Since BA10 only contains equilibrium structures we only fit to total energies. Further, we choose a cut-off radius of 7 Å.

As for the previous alloy datasets MoNbTaW-MEA and MoNbTaVW-HEA, ETN poten-



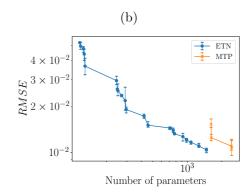


Figure 9: (a) Mean validation loss, and (b) RMSE, as a function of the number of parameters for BA10. The horizontal bars correspond to the minimum and maximum values

tials generally require less coefficienst to reach the accuracy of MTPs, as shown in Figure 9, although the difference is less pronounced (~ 1.5 –1.8, as compared to > 2 for MoNbTaW-MEA and MoNbTaVW-HEA). Since BA10 only contains equilibrium structures, it also appears not too difficult to fit an MTP to it. We yet remark that, for an ETN potential with 10 atomic species, the structure of the ETN (43) may not be optimal since the size of the tensor \underline{A}_{ℓ} is already $100 \times N_{\rm spec}^{\rm rank}(\ell)$. For example, for the ETN potential with L=5 and $N_{\rm spec}^{\rm rank}=3$ (cf., Figure 10) the size of all \underline{A}_{ℓ} 's is 800. We think that an ETN of type (44) that also splits the species features could substantially reduce this amount. We leave this for exploration in future work.

6.2.6 Evolution of the Hyperparameters

We now analyze how the hyperparameters evolve during the optimization procedure. For each dataset, the ETN hyperparameters \mathscr{H} are shown in Figure 10 as a function of the number of coefficients/iteration corresponding to Figure 5–9.

The number radial basis functions $N_{\rm rad}$ grows approximately similar for all datasets, up to 8–11, except for QM9, where $N_{\rm rad}$ grows up to 15.

For the ETT order d and the maximum angular momentum L, we observe an interesting difference between the molecule datasets and the alloy datasets: while d dominates over L for the molecule datasets throughout the optimization procedure, L starts dominating over d for the alloy datasets with an increasing complexity of the potential. This indicates that the body-order is more important for molecules than for alloys, and, vice-versa, the polynomial degree is more important for alloys than for molecules. In addition, we observe that the order does not increase beyond 3 for the alloy datasets for interatomic potentials, i.e., for MoNbTaW-MEA and MoNbTaVW-HEA. Order-3 ETN potentials are more sparse since we can neglect the imaginary parts of the equivariant tensors (cf., Section 3.4.2). This implies that we can construct highly efficient ETN potentials for many-component alloys.

For the ranks, we observe that the species rank $N_{\rm spec}^{\rm rank}$ increases up to 5 and 4, respectively, for the molecule datasets, and up to 3 for the alloy datasets. This is not surprising since QM7 and QM9 contain atoms from different groups (nonmetals and halogens), while the alloy datasets only contain atoms from the group of transition metals, except BA10, but BA10 does not contain all possible pair-wise combinations of atomic species. Moreover, we observe that the radial rank $N_{\rm ran}^{\rm rank}$ always settles at some value that is $\sim 2-3$ times lower than $N_{\rm rad}$. This indicates that our assumption of letting the ranks decay with increasing angular momentum ℓ appears to be a reasonable choice. The ETT rank $N_{\rm ext}^{\rm rank}$ is generally lower than $N_{\rm rad}^{\rm rank}$, which is is expected

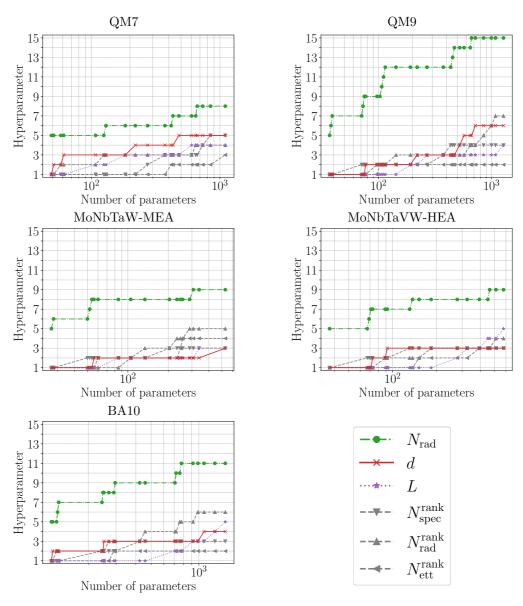


Figure 10: Evolution of the ETN hyperparameters corresponding to the each iteration of the optimization algorithm for the five datasets from Figure 5–9

since a $N_{\rm ett}^{\rm rank} > N_{\rm rad}^{\rm rank}$ may "overparameterize" the potential. The observed $N_{\rm rad}^{\rm rank} > N_{\rm ett}^{\rm rank}$ thus further confirms that our setup of constructing ETN potentials is indeed suitable and validates the correct functioning of the optimization algorithm.

6.3 Comparison with non-polynomial MLIPs

From the two other popular classes of MLIPs, neural network potentials, and Gaussian process regression-based potentials, neural networks achieve state-of-the-art performance for molecule datasets. However, we have found in Section 6.2.2 that converging ETN potentials on molecules with the python BFGS solver is difficult, in particular for ETN potentials with higher body-order.

Name	Hyperparameters	Coefficients
ETN1	$\{7, 3, 1, 2, 2, 1\}$	92
ETN2	$\{8, 3, 2, 3, 3, 2\}$	278
ETN3	$\{9,3,5,3,4,3\}$	577

Table 2: Hyperparameters $\left\{N_{\mathrm{rad}}, d, L, N_{\mathrm{spec}}^{\mathrm{rank}}, N_{\mathrm{rad}}^{\mathrm{rank}}, N_{\mathrm{ett}}^{\mathrm{rank}}\right\}$ and number of coefficients of the three ETN potentials used for the comparison in Section 6.3, where N_{rad} is the maximum number of radial basis functions, d is the order of the ETT, L is the maximum number of angular momenta, and $N_{\mathrm{spec}}^{\mathrm{rank}}$, $N_{\mathrm{rad}}^{\mathrm{rank}}$, and $N_{\mathrm{ett}}^{\mathrm{rank}}$, are the ranks of $\underline{\underline{A}}$, $\underline{\underline{B}}$, and $\underline{\underline{T}}$, respectively

Error measure	ETN1	ETN2	ETN3	GAP	cGAP2	trGAP
RMSE [eV/atom]	0.084	0.025	0.018	0.032	0.019	0.17
RMSF [eV/Å]	0.56	0.3	0.24	0.41	0.33	0.33

Table 3: MoNbTaVW-HEA test errors for the three ETN potentials from Table 2 and the best GAPs reported in [17], [18] (the values are deduced from Figure 8 in [17], and from Figure 3 & 8 in [18]). The ETN test errors are the average errors over five training runs with different parameter initializations

While we have shown in Section 4 that alternating least-squares solvers are in principle stable for ETNs, they still need to be implemented and we thus postpone a comparison for molecules to future work.

In the case of alloys, the—in our view—presently most interesting dataset is MoNbTaVW-HEA because it contains configurations with lattice defects and liquid configurations that are all relevant for predicting mechanical properties. For this dataset, several Gaussian approximation potentials (GAPs) have been proposed in recent works by Darby, Kermode, and Csányi [17] and Darby, Kovács, Batatia, et al. [18], which also introduce compression schemes comparable to our tensor network approach. The GAP in [17] (in the following cGAP) uses a compression scheme that exploits symmetries in the power spectrum; the GAP in [18] (in the following trGAP) introduces a tensor-reduced parameter tensor using the canonical polyadic decomposition.

In order to compare those GAPs with ETN potentials, we have picked the three ETN potentials shown in Table 2 that were found during the optimization (Figure 8). All potentials use four-body interactions (d=3), but a different number of radial basis functions, angular momenta, and ranks. As in [17], [18], we have taken the reduced subset of 2329 configurations from MoNbTaVW-HEA for the training, and benchmark the potentials on the test set from [41]. Moreover, we now use 5000 BFGS iterations; all other parameters remain the same.

We then compare these three ETN potentials with the best (non-compressed) GAP, cGAP, and trGAP, which use three-body interactions. The GAP and cGAP use 12 radial basis functions and a maximum angular momenta of 9. The trGAP uses 8 radial basis functions and and a maximum angular momentum of 4, as for the ETN3 potential. Hence, the complexity of contracting radial basis functions and spherical harmonics is similar or lower for the ETN potentials.

Interestingly, an ETN potential with a maximum angular momentum of only 1 (ETN2) achieves a comparable accuracy to that of the three GAPs (Table 3). The energy error for the best ETN potential (ETN3) is the same as for the best GAP (trGAP), while the force error is lower, which might come from the higher body-order. However, we remark that increasing the body-order in GAPs is computationally expensive, while ETNs can achieve a linear scaling (at least theoretically). In addition, a GAP with higher body-order likely requires more sparse points/training data to achieve the same accuracy.

Our findings and those of Darby et al. thus indicate that, with a suitable compression of

the feature space, MLIPs with higher body-order may require less training data than without any compression. Our optimization algorithm provides an efficient means for finding a suitable compression, something that would have been difficult to achieve with expensive grid searches.

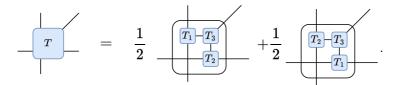
7 Generalizations to Other Tensor Networks

We think of the above construction as the simplest construction incorporating our main contribution—a way of incorporating equivariance in tensor networks. As hinted in Section 4, many ideas developed for the conventional tensor networks are transferable to the equivariant tensor networks. Below we list some of the ideas that we find can be immediately applied to the equivariant tensor networks.

7.1 Arbitrary topology of tensor network cores

The method of reshaping of two adjacent order-3 tensors, as illustrated Figure 4, as well as noting that it would also trivially work with one or two order-2 tensors, can be applied to construct an arbitrary topology involving order-3 tensors. For instance the hierarchical Tucker format, as sketched in Section 2.4 can readily be realized with our construction.

To implement PEPs or other tensor formats with tensors with order higher than three, one either has to use higher-order 3-j (i.e., "4-j", "5-j", etc.) tensors along with learnable coefficients or split the large order-5 tensor (as features in the two-dimensional PEPs) into three order-3 tensors which can then be symmetrized to avoid bias in the x vs. y coordinate. A possible way of doing it for the two-dimensional PEPs is



7.2 Weight sharing

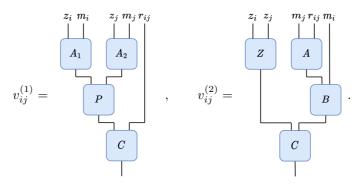
The ETNs as introduced in Section 4 depend linearly on the learnable weights in each of the tensors, however, already in our numerical experiments in Section 6.2 we used weight sharing for the tensors that compress the input features. This resulted in a more parameter-efficient model, however, some of the (multi-)linear algebra algorithms (like DMRG or ALS) would not apply.

7.3 Richer atomic features

The flexible nature of equivariant tensor networks allows for equivariantly incorporating more atomic features like magnetic moments, charge, dipole moments, etc. [43], [44], and at the same time compress them in an efficient and *physically interpretable* manner.

For instance, two possible way of treating magnetic moments (m_i on the central atom and

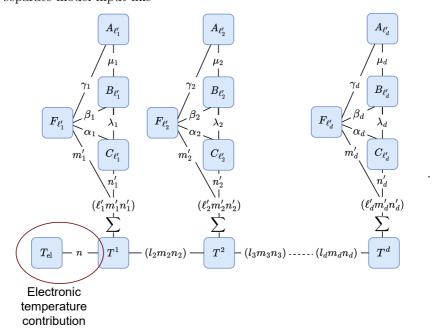
 m_i on the neighboring atom) are



Here in $v_{ij}^{(1)}$ we mix atomic species with magnetic moments into combined features with tensors A_1 and A_2 . Physically, we would say that we construct features that depend on the number (z) and magnetic state (m) of the core electrons of the i-th and j-th atoms. Also, from the physical point of view it may be reasonable for them to share weights, i.e., to fix $A_1 = A_2$. Then, these combined features are mixed together to produce information how the (z,m) pair for the central atom should interact with another (z,m) of the neighboring atom, which is then contracted with the relative positions r_{ij} of these atoms to form the final feature vector of interaction of the i-th and j-th atoms. A possible computational advantage of this method is that combining the m and z features with the A_2 vector can be precomputed and reused while processing different neighborhoods.

In $v^{(2)}$, instead, z_i and z_j are contracted using the learnable Z tensor—this operation is essentially a look-up for the (z_i, z_j) -index feature vector which is then contracted with a similarly formed feature vector obtained from m_i , m_j , and r_{ij} . Which of these two (or many other possible) variants is better in terms of accuracy or efficiency is yet to be tested.

Another example is adding non-atomic features like electronic temperature $T_{\rm el}$ to the potential [45]–[47]. A way to do it could be adding an atomic feature to all the feature vectors F or, instead, make it a separate model input like



7.4 Message passing/graph convolutions, and pooling operations

To introduce the message passing (also known as graph convolution operations) [7], [48], [49] we recall that each of the input features $F_{i,(\ell mn)}$ also depend on the particular atom i which forms another tensor dimension. Instead of just summing all the features over the neighborhood of the i-th atom, we explicitly introduce the convolution operator K as

$$(K)_{ij} = \underbrace{\begin{bmatrix} i \\ K \\ j \end{bmatrix}}_{j} = k(r_{ij}),$$

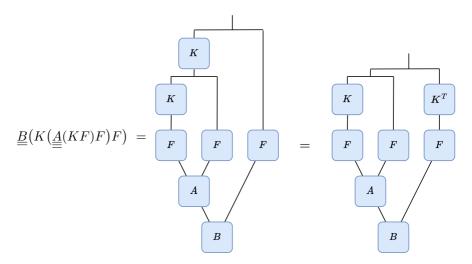
where k is some smooth cut-off function that serves as the convolution kernel. The operation itself acts on features

$$F_{i,(\ell mn)} = \begin{array}{c} \stackrel{i}{\underset{(\ell,m,n)}{|}} \\ \stackrel{i}{\underset{(\ell,m,n)}{|}} \end{array} ,$$

and distributes them over to the neighboring atoms:

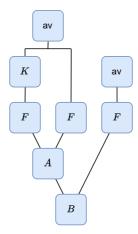
$$G_{i,(\ell mn)} = \sum_{j} (K)_{ij} F_{j,(\ell mn)}, \quad \text{or} \quad \begin{array}{c} i \\ G \\ \vdots \\ (\ell,m,n) \end{array} = \begin{array}{c} i \\ K \\ \vdots \\ F \\ \vdots \\ (\ell,m,n) \end{array}$$

Then, for example, a model in which features F are passed to the neighbors, convolved with F, the result of which is passed to the neighbors and again convolved with F, is given by $\underline{B}(K(\underline{A}(KF)F)F)$ and diagrammatically expressed as

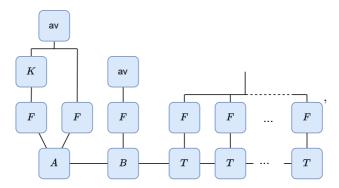


Again, operations on adjacent kernels, like $\underline{\underline{A}}$ and $\underline{\underline{B}}$ in this example, are possible as they are independent of the convolution operations K. The upper "hanging" link indicates that the result is the per-atom energy that should then be summed over this dimension (i.e., over i).

Pooling operations require simply defining an averaging operator over the atoms i. This is generally considered not useful for interatomic potentials because they destroy the locality (short-sightedness) principle, but may be good for cheminformatics models like the prediction of the band gap. For example, a model that takes atomic features F, passes it over to the neighbors, contracts with the original features F using the tensor A, averages, and contracts these averaged features with the averaging over the original features F would be



(here av is averaging over the atoms). If one wants to use this block to inform a regular interatomic-potential-like model (leading to something resembling the attention mechanism), one could design something like



where in the diagram for conciseness we did not illustrate the part with compressing radial, species, etc., information into a single feature vector like it is done with tensors A, B, and C in (44).

8 Concluding Remarks

In this work, we have developed a formalism for constructing equivariant tensor networks (ETNs), i.e., tensor networks that remain invariant under group actions of SO(3). Our main findings is that—analogous to conventional tensor networks—we only require up-to-order-3 equivariant tensors in order to construct arbitrary ETNs. In our view, most importantly, our formalism of

constructing ETNs can therefore lead to a drastic simplification of constructing machine-learning interatomic potentials of arbitrary complexity, which is our main target of application. Our confidence grounds on the fact that many algorithms developed for conventional tensor networks translate almost verbatim to the equivariant case, e.g., automatic differentiation, or tensor orthogonalization (cf. Section 4). So, once these algorithms are implemented, the only thing that would be left is to code the tensor network diagram (and this might be even possible using visual programming environments at some point)—all the rest would literally "fall from the tree".

Representing interatomic interaction energies using ETNs can be considered as a high-performance representation of polynomial machine-learning interatomic potentials that naturally allows for basis- and feature-compression. The numerical results presented in Section 6 indicates that ETNs can reach excellent performance for complex systems. This ETN potential outperformed moment tensor potentials on several databases for multicomponent systems in terms of the required number of parameters by factor of about 2 to 3. Interestingly, we have observed that the optimal hyperparameters do not evolve comparably for different species; for example, molecules benefit from a higher body-order, while metallic alloys benefit from a higher polynomial degree. Future research may thus be devoted to developing improved algorithms that find the best ETN hyperparameters for a given problem. The fact that we could significantly compress the feature space for an ETN potential with a still rather low number of features (positional and species) gives us hope that we can then reach even much better compression rates when adding new features, e.g., magnetic moments. We are planning to explore this in future work, alongside with extending ETNs in many other directions some of which we have outlined in Section 7.

Also, unlike linearly parameterized potentials like MTP or ACE in which the construction of different basis functions requires different operations, the ETN potentials are based on tensorial contractions, which have been easier to parallelize on massively parallel architectures such as GPUs—this could also become another advantage of the ETN models.

9 Acknowledgments

M.H. gratefully acknowledges the financial support under the scope of the COMET program within the K2 Center "Integrated Computational Material, Process and Product Engineering (IC-MPPE)" (Project No 886385). This program is supported by the Austrian Federal Ministries for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) and for Labour and Economy (BMAW), represented by the Austrian Research Promotion Agency (FFG), and the federal states of Styria, Upper Austria and Tyrol. A.S. was supported by the Russian Science Foundation (Grant No. 23-13-00332).

Appendix

A Wigner 3-j Symbol for Real Spherical Harmonics

A.1 Selection Rules

We first express the 3-j symbols for RSHs in terms of the usual 3-j symbols. Therefore, we distinguish between between three cases

- (i) $m_1 = m_2 = m_3 = 0$,
- (ii) $\exists ! m = 0 \in \{m_i\},\$
- (iii) $m_1, m_2, m_3 \neq 0$.

Case (i) is trivial since

$$\begin{cases} \ell_1 & \ell_2 & \ell_3 \\ 0 & 0 & 0 \end{cases} = \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ 0 & 0 & 0 \end{pmatrix}. \tag{A1}$$

For case (ii), we first exemplify our derivation for $m_3 = 0$; the other two cases follow analogously. From the conjugate transpose of \underline{U}_{ℓ} (eq. (9))

$$\underline{\underline{U}}_{\ell}^{\mathsf{T}*} = \frac{1}{\sqrt{2}} \begin{pmatrix}
-i & & & & & 1 \\
& -i & & & & 1 \\
& & \ddots & & \ddots & \\
& & & \sqrt{2} & & \\
& & & \ddots & \ddots & \\
& & & & (-1)^{\ell-1} i & & & (-1)^{\ell-1} \\
(-1)^{\ell} i & & & & & (-1)^{\ell-1}
\end{pmatrix}, (A2)$$

we deduce

$$\begin{cases}
\ell_1 & \ell_2 & \ell_3 \\
m_1 & m_2 & 0
\end{cases} = \begin{pmatrix}
\ell_1 & \ell_2 & \ell_3 \\
m_1 & m_2 & 0
\end{pmatrix} U_{\ell_1 m_1 m_1}^{\mathsf{T}*} U_{\ell_2 m_2 m_2}^{\mathsf{T}*} \\
+ \begin{pmatrix}
\ell_1 & \ell_2 & \ell_3 \\
-m_1 & -m_2 & 0
\end{pmatrix} U_{\ell_1 - m_1 m_1}^{\mathsf{T}*} U_{\ell_2 - m_2 m_2}^{\mathsf{T}*} \\
+ \begin{pmatrix}
\ell_1 & \ell_2 & \ell_3 \\
m_1 & -m_2 & 0
\end{pmatrix} U_{\ell_1 m_1 m_1}^{\mathsf{T}*} U_{\ell_2 - m_2 m_2}^{\mathsf{T}*} \\
+ \begin{pmatrix}
\ell_1 & \ell_2 & \ell_3 \\
m_1 & -m_2 & 0
\end{pmatrix} U_{\ell_1 - m_1 m_1}^{\mathsf{T}*} U_{\ell_2 m_2 m_2}^{\mathsf{T}*}.$$
(A3)

To simplify the latter expression, we use the symmetry property of the Wigner 3-j symbol (cf., [32])

$$\begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = (-1)^{\ell_1 + \ell_2 + \ell_3} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ -m_1 & -m_2 & -m_3 \end{pmatrix}$$
(A4)

and the relations

$$U_{\ell_{i}-m_{i}m_{i}}^{\mathsf{T}*} = \operatorname{sgn}(m_{i})(-1)^{m_{i}}U_{\ell_{i},m_{i}m_{i}}^{\mathsf{T}*} \qquad U_{\ell_{i},m_{i}m_{i}}^{\mathsf{T}*} = \operatorname{sgn}(m_{i})(-1)^{m_{i}}U_{\ell_{i}-m_{i}m_{i}}^{\mathsf{T}*}, \tag{A5}$$

where

$$\operatorname{sgn}(m_i) = \begin{cases} 1 & m_i \ge 0, \\ -1 & m_i < 0 \end{cases}$$
 (A6)

is the sign function. Then,

$$\begin{cases}
\ell_1 & \ell_2 & \ell_3 \\
m_1 & m_2 & 0
\end{cases} = U_{\ell_1 m_1 m_1}^{\mathsf{T}*} U_{\ell_2 m_2 m_2}^{\mathsf{T}*} \left(1 \pm \operatorname{sgn}(m_1) \operatorname{sgn}(m_2)\right) \left(\begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & 0 \end{pmatrix} + \operatorname{sgn}(m_2)(-1)^{m_2} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & -m_2 & 0 \end{pmatrix} \right), \tag{A7}$$

where the + and - in the second parenthesis hold for even or odd $\ell_1 + \ell_2 + \ell_3$, respectively. We omit the derivations for $m_1, m_2 = 0$ as they follow trivially from the previous one and just state the final results: for $m_2 = 0$, we have

$$\begin{cases}
\ell_1 & \ell_2 & \ell_3 \\
m_1 & 0 & m_3
\end{cases} = U_{\ell_1 m_1 m_1}^{\mathsf{T}*} U_{\ell_3 m_3 m_3}^{\mathsf{T}*} \left(1 \pm \operatorname{sgn}(m_1) \operatorname{sgn}(m_3)\right) \left(\begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & 0 & m_3 \end{pmatrix} + \operatorname{sgn}(m_3)(-1)^{m_3} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & 0 & -m_3 \end{pmatrix} \right), \tag{A8}$$

and for $m_1 = 0$

$$\begin{cases}
\ell_1 & \ell_2 & \ell_3 \\
0 & m_2 & m_3
\end{cases} = U_{\ell_2 m_2 m_2}^{\mathsf{T}*} U_{\ell_3 m_3 m_3}^{\mathsf{T}*} \left(1 \pm \operatorname{sgn}(m_2) \operatorname{sgn}(m_3)\right) \left(\begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ 0 & m_2 & m_3 \end{pmatrix} + \operatorname{sgn}(m_3)(-1)^{m_3} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ 0 & m_2 & -m_3 \end{pmatrix} \right).$$
(A9)

For case (iii), by using (A2) in (10), we have

$$\begin{cases} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{cases} = \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{pmatrix} U_{\ell_1 m_1 m_1}^{\mathsf{T*}} U_{\ell_2 m_2 m_2}^{\mathsf{T*}} U_{\ell_3 m_3 m_3}^{\mathsf{T*}} \\ & + \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ -m_1 & -m_2 & -m_3 \end{pmatrix} U_{\ell_1 -m_1 m_1}^{\mathsf{T*}} U_{\ell_2 -m_2 m_2}^{\mathsf{T*}} U_{\ell_3 -m_3 m_3}^{\mathsf{T*}} \\ & + \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & -m_3 \end{pmatrix} U_{\ell_1 m_1 m_1}^{\mathsf{T*}} U_{\ell_2 m_2 m_2}^{\mathsf{T*}} U_{\ell_3 -m_3 m_3}^{\mathsf{T*}} \\ & + \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ -m_1 & -m_2 & m_3 \end{pmatrix} U_{\ell_1 -m_1 m_1}^{\mathsf{T*}} U_{\ell_2 -m_2 m_2}^{\mathsf{T*}} U_{\ell_3 m_3 m_3}^{\mathsf{T*}} \\ & + \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & -m_2 & m_3 \end{pmatrix} U_{\ell_1 m_1 m_1}^{\mathsf{T*}} U_{\ell_2 -m_2 m_2}^{\mathsf{T*}} U_{\ell_3 m_3 m_3}^{\mathsf{T*}} \\ & + \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ -m_1 & m_2 & -m_3 \end{pmatrix} U_{\ell_1 -m_1 m_1}^{\mathsf{T*}} U_{\ell_2 m_2 m_2}^{\mathsf{T*}} U_{\ell_3 -m_3 m_3}^{\mathsf{T*}} \\ & + \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ -m_1 & m_2 & m_3 \end{pmatrix} U_{\ell_1 -m_1 m_1}^{\mathsf{T*}} U_{\ell_2 m_2 m_2}^{\mathsf{T*}} U_{\ell_3 m_3 m_3}^{\mathsf{T*}} \\ & + \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ -m_1 & m_2 & m_3 \end{pmatrix} U_{\ell_1 -m_1 m_1}^{\mathsf{T*}} U_{\ell_2 m_2 m_2}^{\mathsf{T*}} U_{\ell_3 -m_3 m_3}^{\mathsf{T*}} \\ & + \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ -m_1 & m_2 & m_3 \end{pmatrix} U_{\ell_1 -m_1 m_1}^{\mathsf{T*}} U_{\ell_2 -m_2 m_2}^{\mathsf{T*}} U_{\ell_3 -m_3 m_3}^{\mathsf{T*}}. \end{cases}$$

Using (A4) and (A5), we have

$$\begin{cases}
\ell_{1} & \ell_{2} & \ell_{3} \\
m_{1} & m_{2} & m_{3}
\end{cases} = U_{\ell_{1}m_{1}m_{1}}^{\dagger *} U_{\ell_{2}m_{2}m_{2}}^{\dagger *} U_{\ell_{3}m_{3}m_{3}}^{\dagger *} \left(1 \pm \operatorname{sgn}(m_{1}) \operatorname{sgn}(m_{2}) \operatorname{sgn}(m_{3})\right) \left(\begin{pmatrix} \ell_{1} & \ell_{2} & \ell_{3} \\
m_{1} & m_{2} & m_{3} \end{pmatrix} + \operatorname{sgn}(m_{3})(-1)^{m_{3}} \begin{pmatrix} \ell_{1} & \ell_{2} & \ell_{3} \\
m_{1} & m_{2} & -m_{3} \end{pmatrix} + \operatorname{sgn}(m_{2})(-1)^{m_{2}} \begin{pmatrix} \ell_{1} & \ell_{2} & \ell_{3} \\
m_{1} & -m_{2} & m_{3} \end{pmatrix} + \operatorname{sgn}(m_{1})(-1)^{m_{1}} \begin{pmatrix} \ell_{1} & \ell_{2} & \ell_{3} \\
-m_{1} & m_{2} & m_{3} \end{pmatrix} \right), \tag{A11}$$

where, again, the + and - in the second parenthesis hold for even or odd $\ell_1 + \ell_2 + \ell_3$, respectively. From (A1), (A7)–(A9), and (A11), it follows that the 3-j symbol for RSHs obeys the selection rules [SR3] and [SR4], in addition to the triangular inequality [SR1] for the three ℓ 's.

Further, for the 3-j symbol for RSHs being a real number $\neq 0$, one of the m_i 's must be positive which follows from the definition of $U_{\ell_i m_i m_i}^{\mathsf{T*}}$ (A2). Vice versa, for being a complex number with pure imaginary part $\neq 0$, one of the m_i 's must be negative. Therefore, the 3-j symbol for RSHs is always real whenever $\ell_1 + \ell_2 + \ell_3$ is even—and pure imaginary whenever $\ell_1 + \ell_2 + \ell_3$ is odd.

A.2 Sparsity

To analyze the sparsity, i.e., the number of nonzero entries divided by the size of the tensor, we consider 3-j symbols with some fixed maximum angular momentum. The sparsity of the 3-j

symbols for real and complex spherical harmonics is shown in Figure 11 (a).

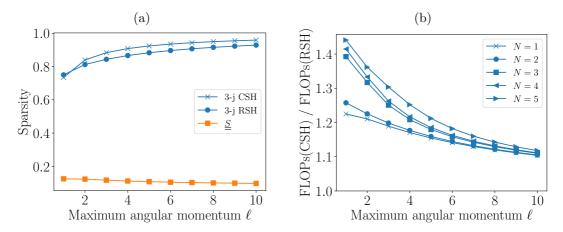


Figure 11: (a) Sparsity of the 3-j symbols for real and complex spherical harmonics, and the order-2 tensor \underline{S} from (A12). (b) Ratio of required FLOPs for the contractions $u_i = T_{ijk}v_jw_k$ with a different maximum number of channels N when using a complex and a real spherical harmonics basis, respectively

Interestingly, even though the 3-j symbol for RSHs is slightly less sparse, using an RSH basis requires slightly less floating point operations (FLOPs) for tensor contractions than using a complex spherical harmonics basis. To see this, let $\underline{\underline{T}}$ be an equivariant tensor of size $M \times M \times M$ and let its channels decay with increasing ℓ according to eq. (45). The maximum number of channels is denoted by $N = N(\ell = 0)$.

We exemplify our analysis by means of the contraction operation $u_i = T_{ijk}v_jw_k$, that is most relevant for contracting our ETNs, where u_i and w_i are two covariant but otherwise arbitrary vectors, and v_j is the input feature vector. To that end, first suppose that we are using a complex spherical harmonics basis. Then, T_{ijk} is a real-valued tensor with $N_{\rm nz}^{\rm csh}$ nonzero entries, and the feature vector v_j is complex-valued. The contraction can thus be split into the following operations

$$\begin{split} S_{ik}^{\text{re}} &= T_{ijk} v_{j}^{\text{re}} & (N_{\text{nz}}^{\text{csh}} \text{ FLOPs}) \\ S_{ik}^{\text{im}} &= T_{ijk} v_{j}^{\text{im}} & (N_{\text{nz}}^{\text{csh}} \text{ FLOPs}) \\ u_{i}^{\text{re}} &= S_{ik}^{\text{re}} w_{k}^{\text{re}} + S_{ik}^{\text{im}} w_{k}^{\text{im}} & (2M^{2} \text{ FLOPs}) \\ u_{i}^{\text{im}} &= S_{ik}^{\text{re}} w_{k}^{\text{im}} + S_{ik}^{\text{im}} w_{k}^{\text{re}} & (2M^{2} \text{ FLOPs}) \end{split}$$

$$(A12)$$

with the corresponding FLOPs given in parenthesis; we have assumed fused multiply-add instructions and that the intermediate tensor S_{ik} is stored in dense format as it is not very sparse (cf., Figure 11 (a)).

Now assume a real basis. Then, T_{ijk} is a complex-valued tensor with N_{nz}^{rsh} nonzero entries, and the feature vector v_j is real-valued. The first two operations from (A12) therefore change to

$$S_{ik}^{\text{re}} = T_{ijk}^{\text{re}} v_j \quad (N_{\text{nz}}^{\text{rsh/re}} \text{ FLOPs})$$

$$S_{ik}^{\text{im}} = T_{ijk}^{\text{im}} v_j \quad (N_{\text{nz}}^{\text{rsh/im}} \text{ FLOPs})'$$
(A13)

where $N_{\rm nz}^{\rm rsh/re}$ and $N_{\rm nz}^{\rm rsh/im}$ are the numbers of nonzeros of the real and imaginary part of $\underline{\underline{T}}$, respectively. Due to the fact that an entry of $\underline{\underline{T}}$ is either pure real or pure imaginary, we

have $N_{\rm nz}^{\rm rsh}=N_{\rm nz}^{\rm rsh/re}+N_{\rm nz}^{\rm rsh/im}$. Hence, using RSHs requires less FLOPs since $N_{\rm nz}^{\rm rsh}+4M^2<2N_{\rm nz}^{\rm csh}+4M^2$ for all tensor sizes shown in Figure 11 (b).

A.3 Contraction Properties

Generating a new covariant vector by contracting the usual 3-j symbol with two covariant vectors, $u_{\ell_1 m_1}$ and $u_{\ell_2 m_2}$, yields another covariant vector $(-1)^{m_3} w_{\ell_3 - m_3}$ with a phase shift. To see this, consider the 3-j symbol in terms of the Clebsch-Gordan coefficients $C_{\ell_1 m_1 \ell_2 m_2}^{\ell_3 m_3}$

$$\begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = \frac{(-1)^{\ell_1 - \ell_2 - m_3}}{\sqrt{2\ell_3 + 1}} C_{\ell_1 m_1 \ell_2 m_2}^{\ell_3 - m_3}, \tag{A14}$$

and recall that the Clebsch-Gordan coefficients expand the basis $|\ell_3 m_3\rangle$ in terms of $|\ell_1 m_1\rangle$ and $|\ell_2 m_2\rangle$.

This phase shift is naturally reversed when using the 3-j symbol for RSHs. To understand why this is the case, consider the contraction

$$\sum_{\substack{\ell_3, m_3, \\ m_2 \ m'_2}} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m'_3 \end{pmatrix} U_{\ell_3 m'_3 m_3}^{\mathsf{T}*} \begin{pmatrix} \ell_3 & \ell_4 & \ell_5 \\ m''_3 & m_4 & m_5 \end{pmatrix} U_{\ell_3 m''_3 m_3}^{\mathsf{T}*}. \tag{A15}$$

Since $U_{\ell_3m_3'm_3}^{\mathsf{T}*}U_{\ell_3m_3''m_3}^{\mathsf{T}*}=U_{\ell_3m_3'm_3}^{\mathsf{T}*}U_{\ell_3m_3m_3''}^*$ is the anti-diagonal matrix

we have $v_{\ell m_3'} U_{\ell_3 m_3' m_3}^{\mathsf{T}*} U_{\ell_3 m_3' m_3}^{\mathsf{T}*} = w_{\ell m_3''},$ with $w_{\ell m_3} = (-1)^{-m_3} v_{\ell - m_3}.$

A.4 Symmetry Properties

The 3-j symbols for RSHs obey the same symmetry properties for even and odd permutations than the usual 3-j symbol, i.e.,

$$\begin{cases}
\ell_1 & \ell_2 & \ell_3 \\
m_1 & m_2 & m_3
\end{cases} = \begin{cases}
\ell_3 & \ell_1 & \ell_2 \\
m_3 & m_1 & m_2
\end{cases} = \begin{cases}
\ell_2 & \ell_3 & \ell_1 \\
m_2 & m_3 & m_1
\end{cases},$$
(A17)

and

$$(-1)^{\ell_1+\ell_2+\ell_3} \left\{ \begin{matrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{matrix} \right\} = \left\{ \begin{matrix} \ell_2 & \ell_1 & \ell_3 \\ m_2 & m_1 & m_3 \end{matrix} \right\} = \left\{ \begin{matrix} \ell_1 & \ell_3 & \ell_2 \\ m_1 & m_3 & m_2 \end{matrix} \right\} = \left\{ \begin{matrix} \ell_3 & \ell_2 & \ell_1 \\ m_3 & m_2 & m_1 \end{matrix} \right\}. \tag{A18}$$

On the other hand, changing the sign of the m's gives, under the assumption that [SR3] holds,

$$\begin{cases} \ell_1 & \ell_2 & \ell_3 \\ -m_1 & -m_2 & -m_3 \end{cases} = 0, \tag{A19}$$

as opposed to

$$\begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ -m_1 & -m_2 & -m_3 \end{pmatrix} = (-1)^{\ell_1 + \ell_2 + \ell_3} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{pmatrix}. \tag{A20}$$

A.5 Orthogonality Properties

Recall that the usual 3-j symbol has the following orthogonality property [32]

$$(2\ell+1) \sum_{m_1, m_2} \begin{pmatrix} \ell_1 & \ell_2 & \ell \\ m_1 & m_2 & m \end{pmatrix} \begin{pmatrix} \ell_1 & \ell_2 & \ell' \\ m_1 & m_2 & m' \end{pmatrix} = \delta_{\ell\ell'} \delta_{mm'}. \tag{A21}$$

For the 3-j symbols for RSHs, we obtain almost the same property, namely,

$$(-1)^{\ell_1+\ell_2+\ell}(2\ell+1)\sum_{m_1,m_2} \begin{Bmatrix} \ell_1 & \ell_2 & \ell \\ m_1 & m_2 & m \end{Bmatrix} \begin{Bmatrix} \ell_1 & \ell_2 & \ell' \\ m_1 & m_2 & m' \end{Bmatrix} = \delta_{\ell\ell'}\delta_{mm'}, \tag{A22}$$

which can be derived from the expressions of the 3-j symbols for RSHs (A1), (A7)–(A9), and (A11), in terms of the usual 3-j symbol, and the orthogonality property (A21). Since the 3-j symbol for RSHs is always real whenever $\ell_1 + \ell_2 + \ell_3$ is even, and complex whenever $\ell_1 + \ell_2 + \ell_3$ is odd, the latter is in fact the same as (A21) in the sense of unitarity, i.e.,

$$(2\ell+1)\sum_{m_1,m_2} \begin{cases} \ell_1 & \ell_2 & \ell \\ m_1 & m_2 & m \end{cases} \begin{cases} \ell_1 & \ell_2 & \ell' \\ m_1 & m_2 & m' \end{cases}^* = \delta_{\ell\ell'} \delta_{mm'}, \tag{A23}$$

where $\{\bullet\}^*$ denotes the complex conjugate of the 3-j symbol for RSHs.

B Functional forms of other Machine-Learning Interatomic Potentials

B.1 Moment Tensor Potentials

For MTPs, the per-atom energies are defined as a linear combination of parameters θ_{α} and scalar basis functions B_{α}

$$\mathcal{E}(\{\underline{r}_{ij}\}) = \sum_{\alpha} \theta_{\alpha} B_{\alpha}(\{\underline{r}_{ij}\}). \tag{B1}$$

The basis functions themselves are obtained from scalar contractions of the moment tensor descriptors

$$M^{\mu}_{\langle \nu \rangle} = \sum_{j} C_{\mu n \alpha \beta} Q_{n}(\|\underline{r}_{ij}\|) z^{i}_{\beta} z^{j}_{\gamma} \left(\underbrace{\underline{r}_{ij} \otimes \ldots \otimes \underline{r}_{ij}}_{\nu \text{ times}} \right), \tag{B2}$$

where $C_{\mu n \alpha \beta}$ is another parameter tensor. The numbers μ and ν define the level of $M^{\mu}_{\langle \nu \rangle}$, lev $M_{\mu,\nu} = 2 + 4\mu + \nu$. The polynomial degree and body-order of MTPs are defined by the level lev_{MTP}. Following Gubaev, Podryabinkin, Hart, et al. [50], for a given MTP level, all possible scalar contractions of $M^{\mu}_{\langle \nu \rangle}$'s that satisfy lev_{MTP} $\geq \sum_i \text{lev} M^{\mu_i}_{\langle \nu_i \rangle}$ are included as basis functions.

In order to cast (B1) into a tensor network structure, we slightly re-define the formalism above. To that end, we define the feature tensor

$$F_{n\beta\gamma(\tilde{\ell}\tilde{m})} = Q_n(\|\underline{r}_{ij}\|) z_{\beta}^i z_{\gamma}^j \left(\underbrace{\underline{r}_{ij} \otimes \ldots \otimes \underline{r}_{ij}}_{\tilde{\ell} \text{ times}}\right)_{(\tilde{\ell}\tilde{m})}, \tag{B3}$$

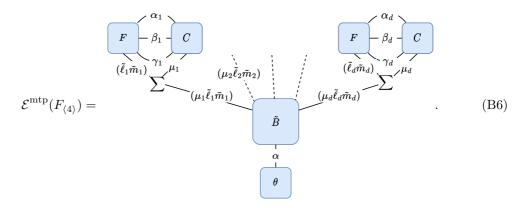
which includes all $M^{\mu}_{\langle\nu\rangle}$'s for a given MTP level, reshaped to a vector; we have chosen its index to be $(\tilde{\ell}\tilde{m})$ to highlight the similarity with respect to the spherical harmonics basis. The per-atom energy can then be written as

$$\mathcal{E}(\{\underline{r}_{ij}\}) = \sum_{\alpha} \theta_{\alpha} \tilde{B}_{\alpha(\mu_1 \tilde{\ell}_1 \tilde{m}_1) \dots (\mu_d \tilde{\ell}_d \tilde{m}_d)} v_{(\mu_1 \tilde{\ell}_1 \tilde{m}_1)} \dots v_{(\mu_d \tilde{\ell}_d \tilde{m}_d)}, \tag{B4}$$

with

$$v_{(\mu_i\tilde{\ell}_i\tilde{m}_i)} = \sum_i C_{\mu_i n_i \beta_i \gamma_i} F_{n_i \beta_i \gamma_i (\tilde{\ell}_i \tilde{m}_i)}. \tag{B5}$$

Here, \tilde{B}_{\bullet} is an integer tensor that encodes all O(3)-invariant contractions of the features for a given level. MTPs can then be written in the tensor network diagram notation as follows



This diagram should be understood that, unlike using the 3-j (or for that matter Clebsch-Gordan tensor) multiple times to get higher (than three) order polynomials, MTPs use an alternative way of contracting the moments to produce a complete set of basis functions.

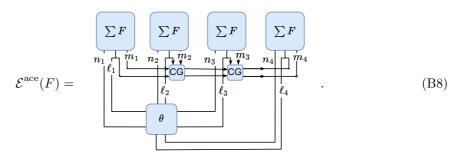
B.2 Spectral Neighbor Analysis and Atomic Cluster Expansion Potentials

The Spectral Neighbor Analysis (SNAP) [8] potentials are four-body potentials (i.e., that multiply the features F with each other three times) and have the following diagram:

Here we simplify the diagram by not explicitly distinguishing the summation-over-neighbors operations and not considering separately radial from chemical features (thus not distinguishing the original SNAP from explicit multielement SNAP [51]). These potentials use Clebsch-Gordan coefficients to ensure the rotational symmetry and have the order-3 (ℓn)-parametrized tensor θ with learnable coefficients. If we replace the Clebsch-Gordan coefficients with the 3-j coefficients the way they are introduced in this paper, then SNAP can be considered as simply contracting the three feature vectors with a single equivariant order-3 tensor as introduced in Figure 3(a).

The Atomic Cluster Expansion (ACE) potentials [10] replicate the construction (B7) to a chain of Clebsch-Gordan tensors and a large-order tensor of coefficients θ , the example of which

being



Compared to MTP, ACE uses the procedure based on the Clebsch-Gordan coefficients to produce the set of basis functions (also complete) that is indexed by (ℓ_i, m_i) and whose coefficients form a tensor. In a very recent work a canonical tensor decomposition was applied to ACE to compress the coefficient tensor [18].

Thus, comparing ACE and ETN to SNAP (all three three potentials use spherical harmonics), one can say that ACE generalizes SNAP by chaining the Clebsch-Gordan tensors and forming a large coefficient tensor preserving linearity of the model, while ETN generalizes SNAP by chaining order-3 equivariant tensors, moving from linear SNAP to a multilinear model.

References

- [1] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," *Physical Review Letters*, vol. 98, no. 14, p. 146 401, Apr. 2, 2007. DOI: 10.1103/PhysRevLett.98.146401.
- [2] K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller, "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, event-place: Long Beach, California, USA, Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 992–1002.
- [3] J. S. Smith, O. Isayev, and A. E. Roitberg, "ANI-1: An extensible neural network potential with DFT accuracy at force field computational cost," *Chemical Science*, vol. 8, no. 4, pp. 3192–3203, 2017. DOI: 10.1039/C6SC05720A.
- [4] H. Wang, L. Zhang, J. Han, and E. Weinan, "Deepmd-kit: A deep learning package for many-body potential energy representation and molecular dynamics," *Computer Physics Communications*, vol. 228, pp. 178–184, 2018.
- [5] G. P. Pun, R. Batra, R. Ramprasad, and Y. Mishin, "Physically informed artificial neural networks for atomistic modeling of materials," *Nature Communications*, vol. 10, no. 1, p. 2339, Dec. 2019. DOI: 10.1038/s41467-019-10343-5.
- [6] S. Takamoto, S. Izumi, and J. Li, "TeaNet: Universal neural network interatomic potential inspired by iterative electronic relaxations," *Computational Materials Science*, vol. 207, p. 111 280, May 2022. DOI: 10.1016/j.commatsci.2022.111280.
- [7] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, "E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials," *Nature Communications*, vol. 13, no. 1, p. 2453, May 4, 2022. DOI: 10.1038/s41467-022-29939-5.

- [8] A. Thompson, L. Swiler, C. Trott, S. Foiles, and G. Tucker, "Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials," *Journal of Computational Physics*, vol. 285, pp. 316–330, Mar. 2015. DOI: 10.1016/j.jcp.2014.12.018.
- [9] A. V. Shapeev, "Moment tensor potentials: A class of systematically improvable interatomic potentials," *Multiscale Modeling & Simulation*, vol. 14, no. 3, pp. 1153–1173, Jan. 2016. DOI: 10.1137/15M1054183.
- [10] R. Drautz, "Atomic cluster expansion for accurate and transferable interatomic potentials," *Physical Review B*, vol. 99, no. 1, p. 014104, Jan. 8, 2019. DOI: 10.1103/PhysRevB.99.014104.
- [11] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," *Physical Review Letters*, vol. 104, no. 13, p. 136 403, Apr. 1, 2010. DOI: 10.1103/PhysRevLett.104.136403.
- [12] R. Jinnouchi, K. Miwa, F. Karsai, G. Kresse, and R. Asahi, "On-the-fly active learning of interatomic potentials for large-scale atomistic simulations," *The Journal of Physical Chemistry Letters*, vol. 11, no. 17, pp. 6946–6955, 2020.
- [13] J. Vandermause, S. B. Torrisi, S. Batzner, Y. Xie, L. Sun, A. M. Kolpak, and B. Kozinsky, "On-the-fly active learning of interpretable bayesian force fields for atomistic rare events," npj Computational Materials, vol. 6, no. 1, p. 20, 2020.
- [14] R. Kondor, Z. Lin, and S. Trivedi, "Clebsch—gordan nets: A fully fourier space spherical convolutional neural network," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc., 2018.
- [15] Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen, J. Behler, G. Csányi, A. V. Shapeev, A. P. Thompson, M. A. Wood, and S. P. Ong, "Performance and cost assessment of machine learning interatomic potentials," *The Journal of Physical Chemistry A*, vol. 124, no. 4, pp. 731–745, Jan. 30, 2020. DOI: 10.1021/acs.jpca.9b08723.
- [16] N. Artrith, A. Urban, and G. Ceder, "Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species," *Physical Review B*, vol. 96, no. 1, p. 014112, Jul. 21, 2017. DOI: 10.1103/PhysRevB.96.014112.
- [17] J. P. Darby, J. R. Kermode, and G. Csányi, "Compressing local atomic neighbourhood descriptors," npj Computational Materials, vol. 8, no. 1, p. 166, Aug. 11, 2022. DOI: 10.1038/s41524-022-00847-y.
- [18] J. P. Darby, D. P. Kovács, I. Batatia, M. A. Caro, G. L. W. Hart, C. Ortner, and G. Csányi, "Tensor-reduced atomic density representations," 2022, Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.2210.01705.
- [19] N. Lopanitsyna, G. Fraux, M. A. Springer, S. De, and M. Ceriotti, "Modeling high-entropy transition-metal alloys with alchemical compression," 2022, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2212.13254.
- [20] I. V. Oseledets, "Tensor-train decomposition," SIAM Journal on Scientific Computing, vol. 33, no. 5, pp. 2295–2317, Jan. 2011. DOI: 10.1137/090752286.
- [21] A. Cichocki, A.-H. Phan, Q. Zhao, N. Lee, I. V. Oseledets, M. Sugiyama, and D. Mandic, "Tensor networks for dimensionality reduction and large-scale optimizations. part 2 applications and future perspectives," Foundations and Trends® in Machine Learning, vol. 9, no. 6, pp. 249–429, 2017. DOI: 10.1561/2200000067. arXiv: 1708.09165.

- [22] T. Kostiuchenko, F. Körmann, J. Neugebauer, and A. Shapeev, "Impact of lattice relaxations on phase transitions in a high-entropy alloy studied by machine-learning potentials," npj Computational Materials, vol. 5, no. 1, p. 55, 2019.
- [23] R. Jana and M. A. Caro, "Searching for iron nanoparticles with a general-purpose gaussian approximation potential," arXiv preprint arXiv:2302.13722, 2023.
- [24] R. Orús, "Tensor networks for complex quantum systems," *Nature Reviews Physics*, vol. 1, no. 9, pp. 538–550, Aug. 5, 2019. DOI: 10.1038/s42254-019-0086-7.
- [25] M. Fannes, B. Nachtergaele, and R. F. Werner, "Finitely correlated states on quantum spin chains," *Communications in Mathematical Physics*, vol. 144, no. 3, pp. 443–490, Mar. 1992. DOI: 10.1007/BF02099178.
- [26] D. Perez-Garcia, F. Verstraete, M. Wolf, and J. Cirac, "Matrix product state representations," *Quantum Information and Computation*, vol. 7, no. 5, pp. 401–430, Jul. 2007. DOI: 10.26421/QIC7.5-6-1.
- [27] J. C. Bridgeman and C. T. Chubb, "Hand-waving and interpretive dance: An introductory course on tensor networks," *Journal of Physics A: Mathematical and Theoretical*, vol. 50, no. 22, p. 223 001, Jun. 2, 2017. DOI: 10.1088/1751-8121/aa6dc3.
- [28] L. Grasedyck, "Hierarchical singular value decomposition of tensors," SIAM Journal on Matrix Analysis and Applications, vol. 31, no. 4, pp. 2029–2054, Jan. 2010. DOI: 10.1137/090764189.
- [29] F. Verstraete and J. I. Cirac, "Renormalization algorithms for quantum-many body systems in two and higher dimensions," 2004, Publisher: arXiv Version Number: 1. DOI: 10.48550/ ARXIV.COND-MAT/0407066.
- [30] J. J. Sakurai and J. Napolitano, *Modern Quantum Mechanics*, 3rd ed. Cambridge University Press, Sep. 17, 2020. DOI: 10.1017/9781108587280.
- [31] M. A. Blanco, M. Flórez, and M. Bermejo, "Evaluation of the rotation matrices in the basis of real spherical harmonics," *Journal of Molecular Structure: THEOCHEM*, vol. 419, no. 1, pp. 19–27, Dec. 1997. DOI: 10.1016/S0166-1280(97)00185-1.
- [32] D. A. Varshalovich, A. N. Moskalev, and V. K. Khersonskii, Quantum Theory of Angular Momentum. WORLD SCIENTIFIC, Oct. 1988. DOI: 10.1142/0270.
- [33] Q. Zhao, M. Sugiyama, L. Yuan, and A. Cichocki, "Learning efficient tensor representations with ring-structured networks," in *ICASSP 2019 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom: IEEE, May 2019, pp. 8608–8612. DOI: 10.1109/ICASSP.2019.8682231.
- [34] S. Holtz, T. Rohwedder, and R. Schneider, "The alternating linear scheme for tensor optimization in the tensor train format," *SIAM Journal on Scientific Computing*, vol. 34, no. 2, A683–A713, Jan. 2012. DOI: 10.1137/100818893.
- [35] S. R. White, "Density matrix formulation for quantum renormalization groups," *Physical Review Letters*, vol. 69, no. 19, pp. 2863–2866, Nov. 9, 1992. DOI: 10.1103/PhysRevLett. 69.2863.
- [36] L. C. Blum and J.-L. Reymond, "970 million druglike small molecules for virtual screening in the chemical universe database GDB-13," *J. Am. Chem. Soc.*, vol. 131, p. 8732, 2009.
- [37] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. v. Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," *Physical Review Letters*, vol. 108, p. 058 301, 2012.

- [38] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, "Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17," *Journal of Chemical Information and Modeling*, vol. 52, no. 11, pp. 2864–2875, Nov. 26, 2012. DOI: 10.1021/ci300415d.
- [39] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," *Scientific Data*, vol. 1, 2014, Publisher: Nature Publishing Group.
- [40] X.-G. Li, C. Chen, H. Zheng, Y. Zuo, and S. P. Ong, "Complex strengthening mechanisms in the NbMoTaW multi-principal element alloy," npj Computational Materials, vol. 6, no. 1, p. 70, Dec. 2020. DOI: 10.1038/s41524-020-0339-0.
- [41] J. Byggmästar, K. Nordlund, and F. Djurabekova, "Modeling refractory high-entropy alloys with efficient machine-learned interatomic potentials: Defects and segregation," *Physical Review B*, vol. 104, no. 10, p. 104 101, Sep. 3, 2021. DOI: 10.1103/PhysRevB.104.104101. arXiv: 2106.03369[cond-mat, physics:physics].
- [42] C. Nyshadham, M. Rupp, B. Bekker, A. V. Shapeev, T. Mueller, C. W. Rosenbrock, G. Csányi, D. W. Wingate, and G. L. W. Hart, "Machine-learned multi-system surrogate models for materials prediction," *npj Computational Materials*, vol. 5, no. 1, p. 51, Dec. 2019. DOI: 10.1038/s41524-019-0189-9.
- [43] R. Drautz, "Atomic cluster expansion of scalar, vectorial, and tensorial properties including magnetism and charge transfer," *Physical Review B*, vol. 102, no. 2, p. 024104, Jul. 10, 2020. DOI: 10.1103/PhysRevB.102.024104.
- [44] I. Novikov, B. Grabowski, F. Körmann, and A. Shapeev, "Magnetic moment tensor potentials for collinear spin-polarized materials reproduce different magnetic states of bcc fe," npj Computational Materials, vol. 8, no. 1, p. 13, Jan. 25, 2022. DOI: 10.1038/s41524-022-00696-9.
- [45] Y. Zhang, C. Gao, Q. Liu, L. Zhang, H. Wang, and M. Chen, "Warm dense matter simulation via electron temperature dependent deep potential molecular dynamics," *Physics of Plasmas*, vol. 27, no. 12, p. 122704, 2020.
- [46] N. Lopanitsyna, C. B. Mahmoud, and M. Ceriotti, "Finite-temperature materials modeling from the quantum nuclei to the hot electron regime," *Physical Review Materials*, vol. 5, no. 4, p. 043 802, 2021.
- [47] J. A. Ellis, L. Fiedler, G. A. Popoola, N. A. Modine, J. A. Stephens, A. P. Thompson, A. Cangi, and S. Rajamanickam, "Accelerating finite-temperature kohn-sham density functional theory with deep neural networks," *Physical Review B*, vol. 104, no. 3, p. 035120, 2021.
- [48] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nature Communications*, vol. 8, no. 1, p. 13890, Jan. 9, 2017. DOI: 10.1038/ncomms13890.
- [49] I. Batatia, D. P. Kovács, G. N. C. Simm, C. Ortner, and G. Csányi, "MACE: Higher order equivariant message passing neural networks for fast and accurate force fields," 2022, Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.2206.07697.
- [50] K. Gubaev, E. V. Podryabinkin, G. L. Hart, and A. V. Shapeev, "Accelerating high-throughput searches for new alloys with active learning of interatomic potentials," *Computational Materials Science*, vol. 156, pp. 148–156, Jan. 2019. DOI: 10.1016/j.commatsci. 2018.09.031.

[51] M. A. Cusentino, M. A. Wood, and A. P. Thompson, "Explicit multielement extension of the spectral neighbor analysis potential for chemically complex systems," *The Journal of Physical Chemistry A*, vol. 124, no. 26, pp. 5456–5464, Jul. 2, 2020. DOI: 10.1021/acs.jpca.0c02450.