LEARNING IRREDUCIBLE REPRESENTATIONS OF NONCOMMUTATIVE LIE GROUPS

Noah Shutty[*]
Department of Physics
Stanford University
Stanford, CA
noaj@stanford.edu

Casimir Wierzynski
Intel AI
San Diego, CA
casimir.wierzynski@intel.com

ABSTRACT

Recent work has constructed neural networks that are equivariant to continuous symmetry groups such as 2D and 3D rotations. This is accomplished using explicit group representations to derive the equivariant kernels and nonlinearities. We present two contributions motivated by frontier applications of equivariance beyond rotations and translations. First, we relax the requirement for explicit Lie group representations, presenting a novel algorithm that finds irreducible representations of noncommutative Lie groups given only the structure constants of the associated Lie algebra. Second, we demonstrate that Lorentz-equivariance is a useful prior for object-tracking tasks and construct the first object-tracking model equivariant to the Poincaré group.

1 INTRODUCTION

Many tasks in machine learning exactly or approximately obey a *continuous symmetry* such as 2D rotations. An ML model is said to be *equivariant* to such a symmetry if the model respects it automatically (without training). Equivariant models have been applied to tasks ranging from computer vision to molecular chemistry, leading to a generalization of equivariance techniques beyond 2D rotations to other symmetries such as 3D rotations. This is enabled by known mathematical results about each new set of symmetries. Specifically, explicit *group representation matrices* for each new symmetry group are required. For many important symmetries, formulae are readily available to produce these representations. For other symmetries we are not so lucky, and the representations may be difficult to find explicitly. In the worst cases, the classification of the group representations is an open problem in mathematics. For example, in the important case of the *homogeneous Galilean group*, which we define in section 2, the classification of the finite dimensional representations is a so-called "wild algebraic problem" for which we have only partial solutions (De Montigny et al., 2006; Niederle & Nikitin, 2006; Levy-Leblond, 1971).

To construct an equivariant network without prior knowledge of the group representations, novel approaches are needed. In this work, we propose an algorithm **LearnRep** that finds the representation matrices with high precision. We validate that LearnRep succeeds for the *Poincaré group*, a set of symmetries governing phenomena from particle physics to object tracking. We further validate LearnRep on two additional sets of symmetries where formulae are known. We apply the Poincaré group representations obtained by LearnRep to construct **SpacetimeNet**, a Poincaré-equivariant object-tracking model. As far as we are aware, LearnRep is the first automated solver which can find explicit representation matrices for sets of symmetries which form *noncompact, noncommutative Lie groups* Further, SpacetimeNet is the first object-tracking model with a rigorous guarantee of Poincaré group equivariance.

1.1 GROUP REPRESENTATIONS AND EQUIVARIANT MACHINE LEARNING

Group theory provides the mathematical framework for describing symmetries and building equivariant ML models. Informally, a symmetry group G is a set of invertible transformations $\alpha, \beta \in G$

[*]Work done during an internship at Intel AI.

which can be composed together using a product operation $\alpha\beta$. We are interested in continuous symmetries for which G is a *Lie group*. In prior constructions of Lie group-equivariant models, *group representations* are required. For a group G, an $n-$dimensional (real) group representation $\rho : G \to \mathbb{R}^{n\times n}$ is a mapping from each element $\alpha \in G$ to an $n \times n$-dimensional matrix $\rho(\alpha)$, such that for any two elements $\alpha, \beta \in G$, we have $\rho(\alpha)\rho(\beta) = \rho(\alpha\beta)$.

Two parallel techniques have been developed for implementing Lie group equivariant neural networks. The first approach was described in general by Cohen et al. (2019). For the latter approach taken by Thomas et al. (2018); Anderson et al. (2019); Bogatskiy et al. (2020), convolutions and nonlinearities are performed directly on the *irreducible* representations of the group, which we define in section 2.4. A common thread in these works has been to utilize existing formulas derived for the matrix elements of these irreducible representations. However, these formulas are only available for specific Lie groups where the representation theory is well-understood. A more convenient approach for extending equivariance to novel Lie groups would utilize an automated computational technique to obtain the required representations. The primary contribution of this work is such a technique.

1.2 Contributions

In this work, we automate the generation of explicit group representation matrices of Lie groups using an algorithm called **LearnRep**. LearnRep poses an optimization problem defined by the *Lie algebra* associated with a Lie group, whose solutions are the representations of the algebra. A penalty term is used to prevent the formation of *trivial representations*. Gradient descent of the resulting loss function produces nontrivial representations upon convergence. We apply LearnRep to three noncommutative Lie groups for which the finite-dimensional representations are well-understood, allowing us to verify that the representations produced are irreducible by computing their *Clebsch-Gordan coefficients* and applying *Schur's Lemma*.

One of the Lie groups where LearnRep performs well is the Lorentz group of special relativity. Prior work has applied Lorentz-equivariant models to particle physics. In this work we explain that the Lorentz group along with the larger Poincaré group also governs everyday object-tracking tasks. We construct a Poincaré-equivariant neural network architecture called **SpacetimeNet** and demonstrate that it can learn to solve a 3D object-tracking task subject to "motion equivariance," where the inputs are a time series of points in space.

In summary, our contributions are:

- **LearnRep**, an algorithm which can find irreducible representations of a noncompact and noncommutative Lie group.
- **SpacetimeNet**, a Poincaré group-equivariant neural network applied to object-tracking tasks.

Our work contributes towards a general technical framework and toolset for building neural networks equivariant to novel Lie groups, and motivates further study of Lorentz equivariance for object tracking.

1.3 Organization

We summarize all necessary background and terminology in section 2. We describe the LearnRep algorithm in section 3.1 and SpacetimeNet in section 3.2. We summarize related work in section 4. We present our experimental results in section 5: our experiments in learning irreducible Lie group representations with LearnRep in section 5.1 and the performance of our Poincaré-equivariant SpacetimeNet model on a 3D object tracking task in section 5.2.

2 Technical Background

We explain the most crucial concepts here and defer to Appendix A.1 for a derivation of the representation theory of the Lorentz group.

2.1 Symmetry Groups $\mathrm{SO}(n)$ and $\mathrm{SO}(m,n)$

A 3D rotation may be defined as a matrix $A :\in \mathbb{R}^{3\times 3}$ which satisfies the following properties, in which $\langle \vec{u}, \vec{v} \rangle = \sum_{i=1}^{3} u_i v_i$:

$$\text{(i)}\ \ \det A = 0 \qquad \text{(ii)}\ \ \forall \vec{u}, \vec{v} \in \mathbb{R}^3, \langle A\vec{u}, A\vec{v} \rangle = \langle \vec{u}, \vec{v} \rangle;$$

these imply the set of 3D rotations forms a group under matrix multiplication and this group is denoted $\mathrm{SO}(3)$. This definition directly generalizes to the $n-$dimensional rotation group $\mathrm{SO}(n)$. For $n \geq 3$, the group $\mathrm{SO}(n)$ is noncommutative, meaning there are elements $A, B \in \mathrm{SO}(n)$ such that $AB \neq BA$. Allowing for rotations and translations of n dimensional space gives the $n-$dimensional special Euclidean group $\mathrm{SE}(n)$.

$\mathrm{SO}(n)$ is generalized by a family of groups denoted $\mathrm{SO}(m,n)$, with $\mathrm{SO}(n) = \mathrm{SO}(n,0)$. For integers $m, n \geq 0$, we define $\langle \vec{u}, \vec{v} \rangle_{m,n} = \sum_{i=1}^{m} u_i v_i - \sum_{i=m+1}^{m+n} u_i v_i$. The group $\mathrm{SO}(m,n)$ is the set of matrices $A \in \mathbb{R}^{(m+n)\times(m+n)}$ satisfying (i-ii) below:

$$\text{(i)}\ \ \det A = 1 \qquad \text{(ii)}\ \ \forall \vec{u}, \vec{v} \in \mathbb{R}^3, \langle A\vec{u}, A\vec{v} \rangle_{m,n} = \langle \vec{u}, \vec{v} \rangle_{m,n};$$

these imply that $\mathrm{SO}(m,n)$ is also a group under matrix multiplication. While the matrices in $\mathrm{SO}(n)$ can be seen to form a compact manifold for any n, the elements of $\mathrm{SO}(m,n)$ form a noncompact manifold whenever $n, m \geq 1$. For this reason $\mathrm{SO}(n)$ and $\mathrm{SO}(m,n)$ are called compact and noncompact Lie groups respectively. The representations of compact Lie groups are fairly well understood, see Bump (2004); Cartan (1930).

2.2 Action of $\mathrm{SO}(m,n)$ on Spacetime

We now explain the physical relevance of the groups $\mathrm{SO}(m,n)$ by reviewing *spacetime*. We refer to Feynman et al. (2011) (ch. 15) for a pedagogical overview. Two observers who are moving at different velocities will in general disagree on the coordinates $\{(t_i, \vec{u}_i)\} \subset \mathbb{R}^4$ of some events in spacetime. Newton and Galileo proposed that they could reconcile their coordinates by applying a spatial rotation and translation (i.e., an element of $\mathrm{SE}(3)$), a temporal translation (synchronizing their clocks), and finally applying a transformation of the following form:

$$t_i \mapsto t_i \qquad \vec{u}_i \mapsto \vec{u}_i + \vec{v} t_i, \tag{1}$$

in which $\vec{v}$ is the relative velocity of the observers. The transformation equation 1 is called a *Galilean boost*. The set of all Galilean boosts along with 3D rotations forms the homogeneous Galilean group denoted $\mathrm{HG}(1,3)$. Einstein argued that equation 1 must be corrected by adding terms dependent on $||\vec{v}||_2/c$, in which c is the speed of light and $||\vec{v}||_2$ is the ℓ_2 norm of $\vec{v}$. The resulting coordinate transformation is called a *Lorentz boost*, and an example of its effect is shown in figure 1. The set of 3D rotations along with Lorentz boosts is exactly the group $\mathrm{SO}(3,1)$. In the case of 2 spatial dimensions, the group is $\mathrm{SO}(2,1)$. Including spacetime translations along with the Lorentz group $\mathrm{SO}(n,1)$ gives the larger Poincaré group $\mathcal{P}_n$ with n spatial dimensions. The Poincaré group $\mathcal{P}_3$ is the group of coordinate transformations between different observers in special relativity.

Consider an object tracking task with input data consisting of a spacetime point cloud with n dimensions of space and 1 of time, and corresponding outputs consisting of object class along with location and velocity vectors. A perfectly accurate object tracking model must respect the action of $\mathcal{P}_n$ on the input. That is, given the spacetime points in *any* observer's coordinate system, the perfect model must give the correct outputs *in that coordinate system*. Therefore the model should be $\mathcal{P}_n$-equivariant. For low velocities the symmetries of the homogeneous Galilean groups $\mathrm{HG}(n,1)$ provide a good approximation to $\mathrm{SO}(n,1)$ symmetries, so Galilean-equivariance may be sufficient for some tasks. Unfortunately the representations of $\mathrm{HG}(n,1)$ are not entirely understood De Montigny et al. (2006); Niederle & Nikitin (2006); Levy-Leblond (1971).

2.3 Lie Groups and Lie Algebras

Here we give an intuitive summary of Lie groups and Lie algebras, deferring to Bump (2004) for a rigorous technical background. A Lie group G gives rise to a *Lie algebra* A as its *tangent space* at the identity. This is a vector space V along with a bilinear product called the *Lie bracket*: $[a, b]$ which

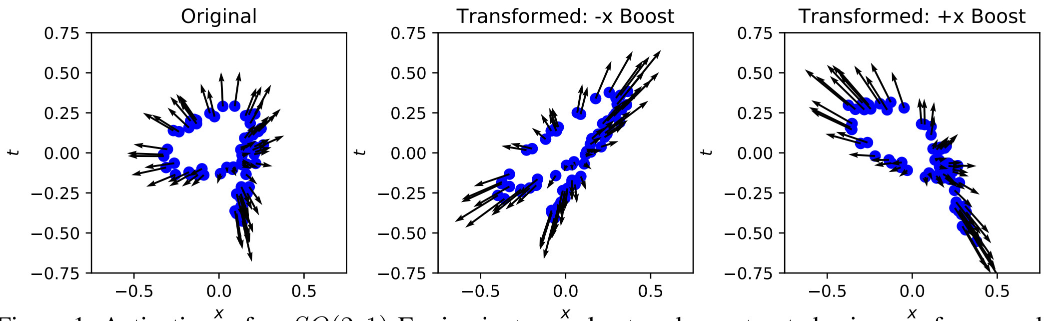

Figure 1: Activations of an $SO(2,1)$-Equivariant neural network constructed using our framework. The arrows depict the elements of the 3-dimensional representation space (arrows) and are embedded on their associated points within the point cloud. This point cloud is from the MNIST-Live dataset as generated with digits embedded in the $x-t$ plane. The y axis is suppressed. The left plot depicts the "original" activations (with the digit at rest). The right plots show what happens if we transform the point cloud with a Lorentz boost in the $\pm x$ direction before feeding it through the network. As dictated by Lorentz-equivariance, the activation vectors generated by the network transform in the same way as the input point cloud.

must behave like[1] the commutator for an associative ring R with multiplication operation $\times_R$:

$$[a,b] = a \times_R b - b \times_R a$$

The Lie algebra for $\mathrm{SO}(3)$, denoted $\mathfrak{so}(3)$, has a basis $\{J_1, J_2, J_3\}$ satisfying

$$[J_i, J_j] = \epsilon_{ijk} J_k, \tag{2}$$

in which $\epsilon_{ijk} \in \{\pm 1, 0\}$ is the totally antisymmetric Levi-Civita symbol.[2] Intuitively, the Lie bracket shows how group elements near the identity fail to commute. For example, the matrices R_x, R_y, R_z for rotations about the x and y axes by a small angle θ satisfy

$$R_x R_y - R_y R_x = R_z + O(\theta^2);$$

more generally the Lie bracket of equation 2 is satisfied to first order in θ. The Lia algebra $\mathfrak{so}(3,1)$ of the Lorentz Group $\mathrm{SO}(3,1)$ also satisfies equation 2 for the generators J_1, J_2, J_3 of its subalgebra isomorphic to $\mathfrak{so}(3)$. It has 3 additional generators denoted K_1, K_2, K_3, which satisfy:

$$[J_i, K_j] = \epsilon_{ijk} K_k \qquad [K_i, K_j] = -\epsilon_{ijk} J_k \tag{3}$$

These K_i correspond to the Lorentz boosts in the same way that the J_i correspond to the rotations. In general, if $\mathcal{A}$ is a t-dimensional Lie algebra with generators $T_1, ..., T_t$ such that

$$[T_i, T_j] = \sum_{k=1}^{t} A_{ijk} T_k, \tag{4}$$

we call the tensor A_{ijk} the *structure constants* of $\mathcal{A}$.

2.4 Group Representations and the Tensor Product

Let G be a Lie group and $\rho : G \to \mathbb{R}^{n\times n}$ be a represenation of G as defined in section 1.1. Then ρ defines a *group action* on $\mathbb{R}^n$: given a vector $\vec{u} \in \mathbb{R}^n$ and a group element $\alpha \in G$, we can define

$$\alpha *_\rho \vec{u} := \rho(\alpha)\vec{u}$$

using the matrix product. We then say that ρ is *irreducible* if it leaves no nontrivial subspace invariant – for every subspace $V \subset \mathbb{R}^n$ with $0 < \dim V < n$, there exists $\alpha \in G, \vec{v} \in V$ such that $\alpha \star_\rho \vec{v} \notin V$.

Given two G-representations $\rho_1 : G \to \mathbb{R}^{n_1\times n_1}$, $\rho_2 : G \to \mathbb{R}^{n_2 \times n_2}$, we define their *tensor product* as

$$\rho_1 \otimes \rho_2 : G \to \mathbb{R}^{n_1 n_2 \times n_1 n_2} \qquad (\rho_1 \otimes \rho_2)(\alpha) = \rho_1(\alpha) \otimes \rho_2(\alpha),$$

[1]Specifically, the Lie bracket must satisfy the *Jacobi identity* and $[a,a] = 0$.
[2]The symbol ϵ_{ijk} simply expresses in equation 2 that $[J_1, J_2] = J_3, [J_2, J_3] = J_1, [J_3, J_1] = J_2$.

in which $\otimes$ on the right hand side denotes the usual tensor product of matrices. It is easy to check that $\rho_1 \otimes \rho_2$ is also a representation of G using the fact that for matrices $A_1, A_2 \in \mathbb{R}^{n_1}$ and $B_1, B_2 \in \mathbb{R}^{n_2 \times n_2}$,

$$(A_1 \otimes B_1)(A_2 \otimes B_2) = (A_1 A_2) \otimes (B_1 B_2).$$

For ρ_1, ρ_2 as above we also define their *direct sum* as

$$(\rho_1 \oplus \rho_2)(\alpha) = \left(\begin{array}{c|c} \rho_1(\alpha) & \\ \hline & \rho_2(\alpha) \end{array}\right).$$

For two groups H, G we say that H is isomorphic to G and write $H \cong G$ if there exists a bijection $f : H \to G$ such that $f(\alpha\beta) = f(\alpha)f(\beta)$. For ρ_1, ρ_2 as above, their images $\rho_i(G)$ form groups and we say that ρ_1 and ρ_2 are *isomorphic* and write $\rho_1 \cong \rho_2$ if these groups are isomorphic, i.e. $\rho_1(G) \cong \rho_2(G)$. Some familiar representations of $\mathrm{SO}(3)$ act on scalars $\in \mathbb{R}$, vectors $\in \mathbb{R}^3$, and tensors (e.g., the Cauchy stress tensor) – these representations are all nonisomorphic.

For many Lie groups such as $\mathrm{SO}(n, 1)$ and $\mathrm{SO}(n)$, a property called *complete reducibility* guarantees that any representation is either irreducible, or isomorphic to a direct sum of irreducible representations. For such groups it suffices to identify the irreducible representations to understand all other representations and construct equivariant models.

2.5 Clebsch-Gordan Coefficients and Tensor-Product Nonlinearities

Clebsch-Gordan Coefficients: Let G be a completely reducible Lie group, and let ρ_1, ρ_2, ρ_3 be irreducible G-representations on the vector spaces $\mathbb{R}^{n_1}, \mathbb{R}^{n_2}, \mathbb{R}^{n_3}$. Consider the tensor product representation $\rho_1 \otimes \rho_2$. Since G is completely reducible, there exists a set S of irreducible representations such that $\rho_1 \otimes \rho_2 \cong \bigoplus_{\rho \in S} \rho$. Suppose that $\rho_3 \in S$. Then there exists a matrix $C \in \mathbb{R}^{n_3 \times (n_1 n_2)}$ which projects the space of the n_3-dimensional group representation ρ_3 from the tensor product space $\mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2}$. That is,

$$\begin{aligned} \forall(\alpha, \vec{u}, \vec{v}) \in G \times \mathbb{R}^{n_1} \times \mathbb{R}^{n_2},\ & C(\rho_1(\alpha) \otimes \rho_2(\alpha))(\vec{u} \otimes \vec{v}) = \rho_3(\alpha) C(\vec{u} \otimes \vec{v}) \\ & \Rightarrow C(\rho_1(\alpha) \otimes \rho_2(\alpha)) = \rho_3(\alpha) C. \end{aligned} \tag{5}$$

The matrices C satisfying equation 5 for various ρ_3 are called the *Clebsch-Gordan coefficients*. In equation 5 there are $n_1 n_2 n_3$ linear constraints on C, and therefore this is a well-posed homogeneous linear program (LP) for C. The entries of C may be found numerically by sampling several distinct $\alpha \in G$ and concatenating the linear constraints (equation 5) to form the final LP. The solutions for C form a linear subspace of $\mathbb{R}^{n_3 \times (n_1 n_2)}$ given by the nullspace of some matrix we denote $\mathcal{C}[\rho_1, \rho_2, \rho_3]$.

Tensor Product Nonlinearities: Tensor product nonlinearities, including norm nonlinearities, use the Clebsch-Gordan coefficients defined above to compute equivariant quadratic functions of multiple G-representations within the G-equivariant model. This was demonstrated for the case of $\mathrm{SE}(3)$ by Thomas et al. (2018); Kondor et al. (2018) and for $\mathrm{SO}(3, 1)$ by Bogatskiy et al. (2020).

3 Methods

3.1 Learning Lie Group Representations

For a matrix $M \in \mathbb{R}^{n \times n}$ we denote its Frobenius and L_1 norms by

$$|M|_F^2 = \sum_{1 \le i,j \le n} |M_{ij}|^2, \qquad |M|_1 = \sum_{1 \le i,j \le n} |M_{ij}|.$$

The approach of LearnRep is to first learn a Lie algebra representation and then obtain its corresponding group representation through the matrix exponential. Fix a t-dimensional Lie algebra $\mathcal{A}$ with structure constants A_{ijk} as defined in equation 4. Fix a positive integer n as the dimension of the representation of $\mathcal{A}$. Then let the matrices $T_1, ..., T_t \in \mathbb{R}^{n \times n}$ be optimization variables, and define the following loss function on the T_i:

$$\mathcal{L}[T_1, ..., T_t] = \underbrace{\max\left(1, \max_{1 \le i \le t} \frac{1}{|T_i|_F^2}\right)}_{N[T_i]^{-1}} \times \sum_{1 \le i \le j \le t} \left| [T_i, T_j] - \sum_k A_{ijk} T_k \right|_1. \tag{6}$$

This is the magnitude of violation of the structure constants of $\mathcal{A}$, multiplied by a norm penalty term $N[T_i]^{-1}$ (this penalty is plotted separately in figure 2). The purpose of the norm penalty is to avoid convergence to the trivial solution $(T_i)_{jk} = 0 \ \forall (i, j, k) \in [t] \times [n] \times [n]$. We pose the optimization problem:

$$\min_{T_1,...,T_t \in \mathbb{R}^{n\times n}} \mathcal{L}[T_1, ..., T_t].$$

The generators were initialized with entries from the standard normal distribution. Gradient descent was performed in PyTorch with the adam optimizer (Kingma & Ba, 2014) with initial learning rate 0.1. The learning rate is set to decrease exponentially when loss plateaus. The results are shown in figure 2.

3.1.1 Verifying Irreducibility of Learned Representations

Suppose we have converged to $T_1, \ldots T_t$ such that $\mathcal{L}[T_i] = 0$. Then the $T_1, ..., T_t$ are a nonzero n-dimensional representation of the Lie *algebra* $\mathcal{A}$. The groups considered here are covered by the exponential map applied to their Lie algebras, so for each $\alpha \in G$ there exist $b_1, \ldots, b_t \in \mathbb{R}$ such that

$$\rho(\alpha) = \exp\left[\sum_{i=1}^{t} b_i T_i\right],$$

where ρ is any $n-$dimensional representation of G and $\exp$ is the matrix exponential. This $\rho : G \mapsto \mathbb{R}^{n\times n}$ is then a representation of the Lie *group*. Throughout this section, ρ denotes this representation. In general ρ may leave some nontrivial subspace invariant. In this case it is *reducible* and splits as the direct sum of lower-dimensional irreducible representations ρ_i as explained in 2.4:

$$\rho \cong \rho_1 \oplus \ldots \oplus \rho_\ell.$$

Recall that any representation may be obtained as such a direct sum of irreducible representations, so it is important to verify that ρ is indeed irreducible, corresponding to $\ell = 1$. To validate that ρ is irreducible, LearnRep computes its tensor product structure and compares with the expected structure. Specifically, it computes the Clebsch-Gordan coefficients for the direct-sum decomposition of the tensor product of the learned representation ρ with several other known representations $\rho_1, ..., \rho_r$. section 2.5 defines these coefficients and explains how they are computed from the nullspace of the matrix $\mathcal{C} = \mathcal{C}[\rho, \rho_1, \rho_2]$, in which ρ_2 appears in the decomposition of $\rho \otimes \rho_1$. Let ρ_1, ρ_2 denote two other known representations, and consider the Clebsch-Gordan coefficients C such that $C\rho\otimes\rho_1 = \rho_3 C$. The dimension of the nullspace of $\mathcal{C}$ indicates the number of unique nonzero matrices C of Clebsch-Gordan coefficients. The singular values of $\mathcal{C}$ are denoted $SV_1(\mathcal{C}) \leq ... \leq SV_\ell(\mathcal{C})$. The ratio

$$r(\mathcal{C}) := SV_2(\mathcal{C})/SV_1(\mathcal{C}) \tag{7}$$

diverges only if the nullspace is one dimensional which therefore corresponds to a unique solution for C. The number of expected solutions is known (e.g., it may be computed using the same technique from the formulae for the irreducible representations). Therefore if $r(\mathcal{C})$ diverges for exactly the choices of ρ_1, ρ_2 where the theory indicates that unique nonzero Clebsch-Gordan coefficients exist, then this is consistent with our having learned an irreducible representation of the group G.

Clearly the tensor product with the trivial representation $\rho_1 = 1$ is $\rho \otimes 1 = \rho$. In this case, the permissible C correspond to $G-$linear maps $\mathbb{R}^n \to \mathbb{R}^{n_2}$. By a result of Schur (1905) (Schur's Lemma), the only such (nonzero) maps are isomorphisms. Therefore a divergent value of $r(\mathcal{C})$ when $\rho_1 = 1$ indicates that $\rho \cong \rho_2$. This is shown in the top row of figure 3 and discussed further in section 5.1.

3.1.2 Stopping condition

Similar to (Rao & Ruderman, 1999), LearnRep restarts gradient descent several times starting from random initialization points. A restart is triggered if loss plateaus and the learning rate is smaller than the loss by a factor of at most 10^{-4}. The tensor product structure is computed upon convergence to loss under 10^{-9}, a restart is triggered if the divergences of $r(\mathcal{C})$ do not agree with the theoretical prediction, indicating a reducible representation.

3.2 SpacetimeNet Architecture

We obtain all Clebsch-Gordan coefficients through the procedure explained in section 2.5. We place them in a tensor: $C_{g,qr,ls,mt}$. This notation corresponds to taking the tensor product of an element of the l^{th} group representation space indexed by s with an element of the m^{th} group representation space indexed by t, and projecting it onto the q^{th} group representation space indexed by r. The space of possible Clebsch-Gordan coefficients can be multidimensional.[3] We use an index g to carry the dimension within the space of Clebsch-Gordan coefficients.

The trainable weights in SpacetimeNet are complex-valued filter weights denoted f^k_{qg} and channel-mixing weights denoted W^k_{qcgd}. Each layer builds a collection of equivariant convolutional filters F^k_{xijqr} from the geometry of the point cloud. Let q' denote the index of the group representation in which the points are embedded. Let X_{xir} denote the point coordinates, in which x indexes the batch dimension, i indexes the points, and r indexes the q' group representation space. Define the (globally) translation-invariant quantity $\Delta X_{xijr} := X_{xjr} - X_{xir}$. The equivariant filters at layer k are:

$$F^k_{xijqr} = \delta_{qq'} \Delta X_{xijr} + \sum_{s,t,g} C_{g,qr,q's,q't} f^k_{qg} \Delta X_{xijs} \Delta X_{xijt}. \tag{8}$$

The input and activations for the k^{th} layer of the network are defined on a tensor V^k_{ximct}, where x is the batch dimension, i indexes the points, m is the group representation index, c is the channel index, t indexes the group representation space. Our mixing weights are then defined for the k^{th} layer as W^k_{qcgd} with layer update rule:

$$V^{k+1}_{xiqcr} = \sum_{g,l,s,m,t,d,j} C_{g,qr,ls,mt} F^k_{xijls} V^k_{xjmdt} W^k_{qcgd}. \tag{9}$$

A proof that SpacetimeNet is $\mathcal{P}_n$-equivariant is given in Appendix A.2.

4 Related Work

4.1 Learning Lie Group Representations

Several authors have investigated automated means of identifying Lie group representations. (Rao & Ruderman, 1999) used gradient descent with several starting points to find the Lie group generators, given many examples of data which had been transformed by the group. Applying the technique requires knowledge of how the group acts on a representation space. In our case we know the Lie algebra structure but we do not know how to compute its representations. Tai et al. (2019) gave a closed-form solution for the canonical coordinates for Lie groups. But their formula only applies for Abelian one-parameter Lie groups, excluding $\mathrm{SO}(3), \mathrm{SO}(2,1)$, and $\mathrm{SO}(3,1)$. Cohen & Welling (2014) devised an interesting probabilistic model to learn representations of *compact, commutative* Lie groups from pairs of images related by group transformations. In the present work we demonstrate a new approach to handle *noncompact* and *noncommutative* groups such as $\mathrm{SO}(3), \mathrm{SO}(2,1)$, and $\mathrm{SO}(3,1)$. Computer algebra software such as the LiE package developed by Van Leeuwen et al. (1992) automates some calculations related to completely reducible Lie groups. Unfortunately this limits us when considering novel Lie groups where the representation theory is less well-understood.

4.2 Equivariant Neural Networks

Beginning with the success of (approximately) translation-equivariant CNNs introduced by LeCun et al. (1989) for image recognition, a line of work has extended equivariance to additional continuous symmetry groups. Most relevant are the architectures for groups $\mathrm{SE}(2)$ (Worrall et al., 2017; Weiler & Cesa, 2019), $\mathrm{SE}(3)$ (Weiler et al., 2018; Cohen et al., 2019; Kondor et al., 2018; Thomas et al., 2018; Cohen et al., 2018; Kondor, 2018; Gao et al., 2020; Anderson et al., 2019; Fuchs et al., 2020; Eismann et al., 2020), and the group of Galilean boosts (Zhu et al., 2019).

The work by Thomas et al. (2018); Kondor et al. (2018); Anderson et al. (2019); Bogatskiy et al. (2020) used Clebsch-Gordan coefficients in their equivariant neural networks. Weiler et al. (2018),

[3]This is common if a group representation is itself obtained via tensor product.

generalized by Cohen et al. (2019) showed all equivariant linear maps are convolutions whose kernels satisfy some linear constraints. In our work we obtain Clebsch-Gordan coefficients from similar linear constraints (equation 5) and use them to show that the learned representations are irreducible. We also use them in SpacetimeNet. Griffiths & Griffiths (2005) provide an introductory exposition of Clebsch-Gordan coefficients and Gurarie (1992) provides a more general exposition.

One of the first constructions that addressed spatiotemporal symmetries was by Zhu et al. (2019). They introduce *motion-equivariant* networks to handle linear optical flow of an observer moving at a fixed speed. They use a canonical coordinate system in which optical flow manifests as a translation, as described for general one dimensional Lie groups by Tai et al. (2019). This allows them to use the translation equivariance of CNNs to produce Galilean boost-equivariance. However, this gives up equivariance to translation in the original coordinate system. To maintain approximate translation-equivariance, the authors apply a spatial transformer network (Jaderberg et al., 2015) to predict a landmark position in each example. This is similar to the work of Esteves et al. (2018), which achieved equivariance to 2D rotation and scale, and approximate equivariance to translation.

The first mention of Poincaré-equivariant networks appears to be a work by Cheng et al. (2019) on the link between covariance in ML and physics. Concurrently to our work, Bogatskiy et al. (2020) constructed a Lorentz-equivariant model which operated on irreducible representations of the Lorentz group, derived similarly to Appendix A.1. This work also made use of the Clebsch-Gordan coefficients, and the model was applied to experimental particle physics rather than object-tracking. Another work by Finzi et al. (2020) concurrent to our own proposed a framework for building models equivariant to arbitrary Lie groups. This work also made use of the exponential and logarithm maps between Lie algebra and group. It does not provide a technique for identifying the Lie algebra representations. Our ideas complement this line of work by providing an algorithm (LearnRep) that solves for the representations numerically.

5 Experiments

5.1 Convergence of LearnRep to Irreducible Representations

We apply LearnRep to $\mathrm{SO}(3), \mathrm{SO}(2,1)$, and $\mathrm{SO}(3,1)$ to learn $3, 3$, and 4 dimensional irreducible representations respectively. The loss function converges arbitrarily close to 0 with the penalty term bounded above by a constant. We exponentiate the resulting algebra representation matrices to obtain group representations and calculate the tensor product structure as described in section 3.1.1 The details of this calculation are in Appendix A.4 and shown in figure 3. The results indicate that the learned representations are irreducible representations of the associated Lie algebras to within numerical error of about 10^{-6}. Schur's Lemma in the special case of the tensor product with the trivial representation indicates the isomorphism class of each learned group representation.

5.2 Poincaré-Equivariant Object-Tracking Networks

We created MNIST-Live, a benchmark dataset of spacetime point clouds sampled from digits from the MNIST dataset moving uniformly through space. Each sample consists of 64 points with uniformly random times $t \in [-1/2, 1/2]$, and spatial coordinates sampled from a 2D probability density function proportional to the pixel intensity. Using instances of the 0 and 9 classes, we train on examples with zero velocity and evaluate on examples with random velocity and orientation. This dataset is analogous to data from an event camera (see (Orchard et al., 2015)) or LIDAR system. We train 3 layer $\mathrm{SO}(2,1)$ and $\mathrm{SO}(3,1)$-equivariant SpacetimeNet models with 3 channels and batch size 16 on 4096 MNIST-Live examples and evaluate on a dev set of 124 examples. We obtain dev accuracy of $80 \pm 5\%$ as shown in figure 4 of the Appendix.

5.3 Conclusion

We envision many applications of Poincaré-equivariant deep neural networks beyond the physics of particles and plasmas. SpacetimeNet can identify and track simple objects as they move through 3D space. This suggests that Lorentz-equivariance is a useful prior for object-tracking tasks. With a treatment of bandlimiting and resampling as in Worrall et al. (2017); Weiler et al. (2018), our work could be extended to build Poincaré-equivariant networks for volumetric data. More broadly, understanding the representations of noncompact and noncommutative Lie groups may enable the construction of networks equivariant to new sets of symmetries such as the Galilean group. Since the representation theory of these groups is not entirely understood, automated techniques such as LearnRep could play a beneficial role.

REFERENCES

Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 14510–14519. Curran Associates, Inc., 2019.

Alexander Bogatskiy, Brandon Anderson, Jan T Offermann, Marwah Roussi, David W Miller, and Risi Kondor. Lorentz group equivariant neural network for particle physics. *arXiv preprint arXiv:2006.04780*, 2020.

Daniel Bump. *Lie groups*. Springer, 2004.

Élie Cartan. *La théorie des groupes finis et continus et l'analysis situs*. Mémorial Sc. Math., 1930.

Miranda CN Cheng, Vassilis Anagiannis, Maurice Weiler, Pim de Haan, Taco S Cohen, and Max Welling. Covariance in physics and convolutional neural networks. *arXiv preprint arXiv:1906.02481*, 2019.

Taco Cohen and Max Welling. Learning the irreducible representations of commutative lie groups. In *International Conference on Machine Learning*, pp. 1755–1763, 2014.

Taco Cohen, Mario Geiger, Jonas K¨ohler, Pim de Haan, K. T. Sch¨utt, and Benjamin K. Miller. Lie learn, February 2020. URL `https://github.com/AMLab-Amsterdam/lie_learn/releases/tag/v1.0_b`.

Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Hkbd5xZRb`.

Taco S Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 9142–9153. Curran Associates, Inc., 2019.

M De Montigny, J Niederle, and AG Nikitin. Galilei invariant theories: I. constructions of indecomposable finite-dimensional representations of the homogeneous galilei group: directly and via contractions. *Journal of Physics A: Mathematical and General*, 39(29):9365, 2006.

Stephan Eismann, Raphael JL Townshend, Nathaniel Thomas, Milind Jagota, Bowen Jing, and Ron Dror. Hierarchical, rotation-equivariant neural networks to predict the structure of protein complexes. *arXiv preprint arXiv:2006.09275*, 2020.

Carlos Esteves, Christine Allen-Blanchette, Xiaowei Zhou, and Kostas Daniilidis. Polar transformer networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=HktRlUlAZ`.

Richard P Feynman, Robert B Leighton, and Matthew Sands. *The Feynman lectures on physics, Vol. I: The new millennium edition: mainly mechanics, radiation, and heat*, volume 1. Basic books, 2011.

Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. *arXiv preprint arXiv:2002.12880*, 2020.

Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020.

Liyao Gao, Yifan Du, Hongshan Li, and Guang Lin. Roteqnet: Rotation-equivariant network for fluid systems with symmetric high-order tensors. *arXiv preprint arXiv:2005.04286*, 2020.

D.J. Griffiths and P.D.J. Griffiths. *Introduction to Quantum Mechanics*. Pearson international edition. Pearson Prentice Hall, 2005. ISBN 9780131118928. URL `https://books.google.com/books?id=z4fwAAAAMAAJ`.

D Gurarie. Symmetries and laplacians. introduction to harmonic analysis, group representations and applications. *North-Holland mathematics studies*, 174:1–448, 1992.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 2017–2025. Curran Associates, Inc., 2015. URL `http://papers.nips.cc/paper/5854-spatial-transformer-networks.pdf`.

J Robert Johansson, Paul D Nation, and Franco Nori. Qutip 2: A python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 184(4):1234–1240, 2013.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Risi Kondor. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials. *arXiv preprint arXiv:1803.01588*, 2018.

Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. In *Advances in Neural Information Processing Systems*, pp. 10117–10126, 2018.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

Jean-Marc Levy-Leblond. Galilei group and galilean invariance. In *Group theory and its applications*, pp. 221–299. Elsevier, 1971.

J Niederle and AG Nikitin. Construction and classification of indecomposable finite-dimensional representations of the homogeneous galilei group. *Czechoslovak Journal of Physics*, 56(10-11): 1243–1250, 2006.

Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015.

Didier Pinchon and Philip E Hoggan. Rotation matrices for real spherical harmonics: general rotations of atomic orbitals in space-fixed axes. *Journal of Physics A: Mathematical and Theoretical*, 40(7): 1597, 2007.

Rajesh PN Rao and Daniel L Ruderman. Learning lie groups for invariant visual perception. In *Advances in neural information processing systems*, pp. 810–816, 1999.

Issai Schur. *Neue begründung der theorie der gruppencharaktere*. 1905.

Kai Sheng Tai, Peter Bailis, and Gregory Valiant. Equivariant transformer networks. *arXiv preprint arXiv:1901.11399*, 2019.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Marc AA Van Leeuwen, Arjeh Marcel Cohen, and Bert Lisser. Lie: A package for lie group computations. 1992.

Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. In *Advances in Neural Information Processing Systems*, pp. 14334–14345, 2019.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In *Advances in Neural Information Processing Systems*, pp. 10381–10392, 2018.

Steven Weinberg. *The quantum theory of fields. Vol. 1: Foundations*. Cambridge University Press, 1995.

Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5028–5037, 2017.

Alex Zihao Zhu, Ziyun Wang, and Kostas Daniilidis. Motion equivariant networks for event cameras with the temporal normalization transform. *arXiv preprint arXiv:1902.06820*, 2019.

A Appendix

A.1 Analytic Derivation of Lorentz Group Representations

To compare our learned group representations with those obtained through prior methods, we require analytical formulae for the Lie algebra representations for the algebras $\mathfrak{so}(3), \mathfrak{so}(3,1)$, and $\mathfrak{so}(2,1)$. The case of $\mathfrak{so}(3)$ has a well-known solution (see Griffiths & Griffiths (2005)). If complex matrices are permissible the library QuTiP Johansson et al. (2013) has a function "jmat" that readily gives the representation matrices. A formulae to obtain real-valued representation matrices is given in Pinchon & Hoggan (2007) and a software implementation is available at Cohen et al. (2020). The three-dimensional Lie algebra $\mathfrak{so}(2,1) = \mathrm{span}\{K_x, K_y, J_z\}$ has structure constants given by equation 3. In fact, these three generators K_x, K_y, J_z may be rescaled so that they satisfy equation 2 instead. This is due to the isomorphism $\mathfrak{so}(3) \cong \mathfrak{so}(2,1)$. Specifically, leting $\{L_x, L_y, L_z\}$ denote a Lie algebra representation of $\mathfrak{so}(3)$, defining

$$K_x = -iL_x \qquad K_y = -iL_y \qquad J_z := L_z,$$

it may be easily checked that K_x, K_y, J_z satisfy the applicable commutation relations from Equation equation 3. This reflects the physical intuition that time behaves like an imaginary dimension of space.

The final Lie algebra for which we require explicit representation matrix formulas is $\mathfrak{so}(3,1)$. Following Weinberg (1995), we define new generators A_i, B_i as

$$A_i := \frac{1}{2}(J_i + iK_i) \qquad B_i := \frac{1}{2}(J_i - iK_i), \tag{10}$$

we see that the $\mathfrak{so}(3,1)$ commutators equation 2, equation 3 become

$$[A_i, A_j] = i\epsilon_{ijk}A_k, \qquad [B_i, B_j] = i\epsilon_{ijk}B_k, \qquad [A_i, B_j] = 0. \tag{11}$$

Therefore $\mathfrak{so}(3,1) \cong \mathfrak{so}(3) \oplus \mathfrak{so}(3)$, and the irreducible algebra representations of $\mathfrak{so}(3,1)$ may be obtained as the direct sum of two irreducible algebra representations of $\mathfrak{so}(3)$.

A.2 Proof that SpacetimeNet is Poincaré-Equivariant

Consider an arbitrary Poincaré group transformation $\alpha \in \mathcal{P}_n$, and write $\alpha = \beta t$ in which $\beta \in \mathrm{SO}(n,1)$ and t is a translation. Suppose we apply this α to the inputs of equation 8 through the representations indexed by q: $\rho_q(\alpha)_{st}$, in which s, t index the representation matrices. Then since the translation t leaves ΔX invariant, the resulting filters will be

$$
\begin{aligned}
F^k_{xijqr} &= \delta_{qq'}\sum_{r'}\rho_{q'}(\beta)_{rr'}\Delta X_{xijr'} + \sum_{s,t,g} C_{g,qr,q's,q't} f^k_{qg} \sum_{s',t'} \rho_{q'}(\beta)_{ss'}\Delta X_{xijs'}\rho_{q'}(\beta)_{tt'}\Delta X_{xijt'} \\
&= \delta_{qq'}\sum_{r'}\rho_{q'}(\beta)_{rr'}\Delta X_{xijr'} + \sum_{g,s',t'}\left(\sum_{s,t} C_{g,qr,q's,q't}\rho_{q'}(\beta)_{ss'}\rho_{q'}(\beta)_{tt'}\right) f^k_{qg}\Delta X_{xijs'}\Delta X_{xijt'} \\
&= \delta_{qq'}\sum_{r'}\rho_{q'}(\beta)_{rr'}\Delta X_{xijr'} + \sum_{s,t,g,r'}\left(\rho_q(\beta)_{rr'} C_{g,qr',q's,q't}\right) f^k_{qg}\Delta X_{xijs}\Delta X_{xijt} \\
&= \sum_{r'}\rho_{q'}(\beta)_{rr'}\left(\delta_{qq'}\Delta X_{xijr'} + \sum_{s,t,g,r'} C_{g,qr',q's,q't} f^k_{qg}\Delta X_{xijs}\Delta X_{xijt}\right) \\
&= \sum_{r'}\rho_{q'}(\beta)_{rr'} F^k_{xijqr'},
\end{aligned}
$$

where we have used equation 5. The network will be equivariant if each layer update is equivariant. Recall the layer update rule of equation 9:

$$
V^{k+1}_{xiqcr} = \sum_{g,l,s,m,t,d,j} C_{g,qr,ls,mt} F^k_{xijls} V^k_{xjmdt} W^k_{qcgd}.
$$

Suppose for the same transformation $\alpha = \beta t$ above, that V^k and ΔX are transformed by α. Then because the activations associated with each point are representations of $\mathrm{SO}(n,1)$, they are invariant to the global translation t of the point cloud and we have

$$
\begin{aligned}
V^{k+1}_{xiqcr} &= \sum_{g,l,s,m,t,d,j} C_{g,qr,ls,mt}\sum_{s'}\rho_m(\beta)_{ss'}F^k_{xijls'}\sum_{t'}\rho_m(\beta)_{tt'}V^k_{xjmdt'}W^k_{qcgd} \\
&= \sum_{s',t'}\sum_{g,l,s,m,t,d,j}\left(C_{g,qr,ls,mt}\rho_m(\beta)_{ss'}\rho_m(\beta)_{tt'}\right) F^k_{xijls'}V^k_{xjmdt'}W^k_{qcgd} \\
&= \sum_{g,l,s,m,t,d,j,r'}\left(\rho_m(\beta)_{rr'}C_{g,qr',ls,mt}\right) F^k_{xijls}V^k_{xjmdt}W^k_{qcgd} \\
&= \sum_{r'}\rho_m(\beta)_{rr'}V^{k+1}_{xiqcr'},
\end{aligned}
$$

where again we applied equation 5.

A.3 Equivariant Convolutions

Consider data on a point cloud consisting of a finite set of spacetime points $\{\vec{x}_i\} \subset \mathbb{R}^4$, a representation $\rho_0 : \mathrm{SO}(3,1) \to \mathbb{R}^{4\times4}$ of the Lorentz group defining its action upon the spacetime, and feature maps $\{\vec{u}_i\}, \{\vec{v}_i\} \subset \mathbb{R}^n$ associated with representations $\rho_u : \mathrm{SO}(3,1) \to \mathbb{R}^{m\times m}$ and $\rho_v : \mathrm{SO}(3,1) \to \mathbb{R}^{n\times n}$. A convolution of this feature map can be written as

$$
\vec{u}'_i = \sum_j \kappa(\vec{x}_j - \vec{x}_i)\vec{u}_j
$$

in which $\kappa(\vec{x}) : \mathbb{R}^4 \to \mathbb{R}^{n\times m}$, a matrix-valued function of spacetime, is the filter kernel.

$\mathcal{P}_3$-equivariance dictates that for any $\alpha \in \mathrm{SO}(3,1)$,

$$
\rho_v(\alpha)\sum_j \kappa(\vec{x}_j - \vec{x}_i)\vec{u}_j = \sum_j \kappa(\rho_1(\alpha)(\vec{x}_j - \vec{x}_i))\rho_u(\alpha)\vec{u}_j
$$

$$
\Rightarrow \kappa(\Delta\vec{x}) = \rho_v(\alpha^{-1})\kappa(\rho_0(\alpha)\Delta\vec{x})\rho_u(\alpha) \quad (12)
$$

Therefore a single kernel matrix in $\mathbb{R}^{n\times m}$ may be learned for each coset of spacetime under the action of $\mathrm{SO}(3,1)$. The cosets are indexed by the invariant

$$
t^2 - x^2 - y^2 - z^2.
$$

The kernel may then be obtained at an arbitrary point $\vec{x} \in \mathbb{R}^4$ from equation 12 by computing an α that relates it to the coset representative $\vec{x}_0$: $\vec{x} = \rho_0(\alpha)\vec{x}_0$. A natural choice of coset representatives for $\mathrm{SO}(3,1)$ acting upon $\mathbb{R}^4$ is the set of points $\{(t,0,0,0) : t \in \mathbb{R}^+\} \cup \{(0,x,0,0) : x \in \mathbb{R}^+\} \cup \{(t,ct,0,0) : t \in \mathbb{R}^+\}$.

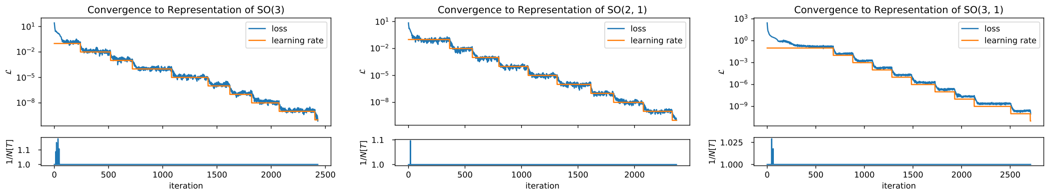

Figure 2: Convergence to arbitrary precision group representations of three Lie groups: $\mathrm{SO}(3), \mathrm{SO}(2,1)$, and $\mathrm{SO}(3,1)$. The multiplicative norm penalty is plotted in each lower subplot, and demonstrates that this penalty is important early on in preventing the learning of a trivial representation, but for later iterations stays at its clipped value of 1. Loss is plotted on each upper subplot.

A.4 Tensor Product Structure of Learned $\mathrm{SO}(3), \mathrm{SO}(2,1), \mathrm{SO}(3,1)$ Group Representations

We quantify the uniqueness of each set of Clebsch-Gordan coefficients in terms of the diagnostic ratio $r(\mathcal{C})$ defined in equation 7. Recall that the value of r becomes large only if there is a nondegenerate nullspace corresponding to a unique set of Clebsch- For $\mathrm{SO}(3)$ and $\mathrm{SO}(2,1)$, the irreducible group representations are labeled by an integer which is sometimes called the *spin*. We label learned group representations with a primed (i') integer. For the case of $\mathrm{SO}(3,1)$ the irreducible group representations are obtained from two irreducible group representations of $\mathfrak{so}(3)$ as explained in section A.1 and we label these representations with both spins i.e. (s_1, s_2). We again label the learned group representations of $\mathrm{SO}(3,1)$ with primed spins, i.e. (s_1', s_2'). The tensor product structures of the representations is shown in figure 3.

We have produced a software library titled *Lie Algebraic Networks* (LAN) built on PyTorch, which derives all Clebsch-Gordan coefficients and computes the forward pass of Lie group equivariant neural networks. LAN also deals with Lie algebra representations, allowing for operations such as taking the tensor product of mutliple group representations. figure 5 demonstrates the LAN library. Starting from several representations for a Lie algebra, LAN can automatically construct a neural network equivariant to the associated Lie group with the desired number of layers and channels. We present our experimental results training $\mathrm{SO}(2,1)$ and $\mathrm{SO}(3,1)$-equivariant object-tracking networks in section 5.2.

A.5 Supplementary figures

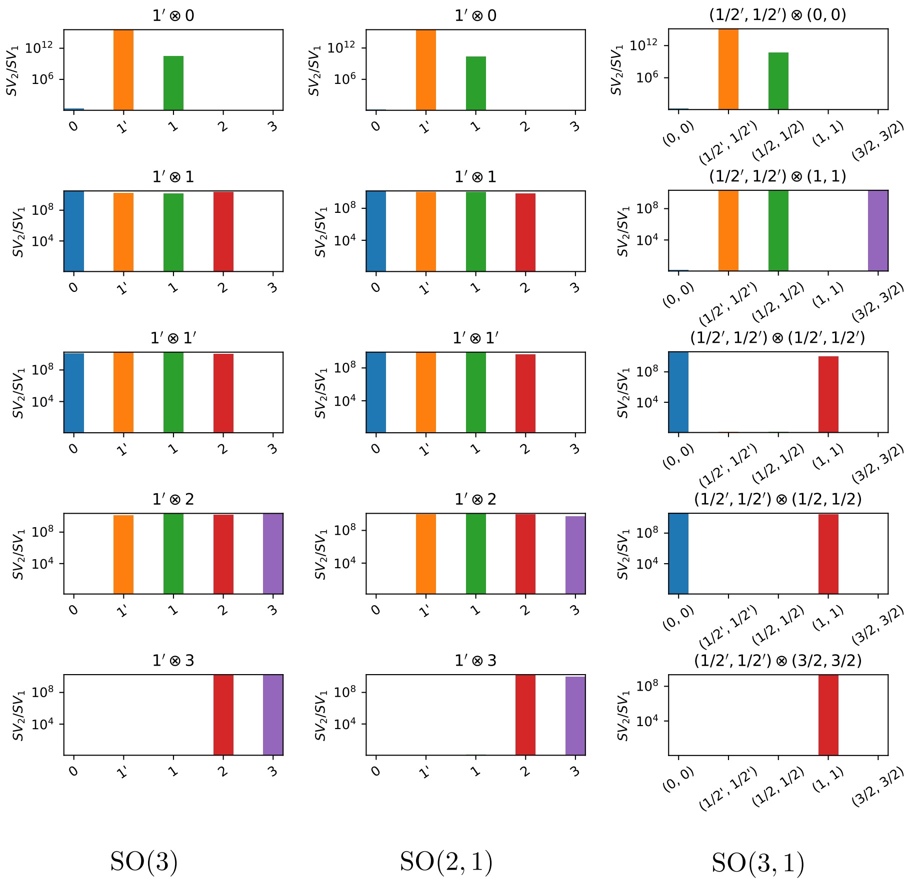

Figure 3: Tensor product structure of the learned group representations ρ with several known (analytically-derived) group representations ρ_1 for the groups $\mathrm{SO}(3), \mathrm{SO}(2,1)$, and $\mathrm{SO}(3,1)$. Each column is for the group indicated at the bottom, each row is for a different choice of ρ_1 for that group, and the horizontal axis indicates the $\rho^{(i)}$ onto which we project the tensor product $\rho \otimes \rho_1 \cong \oplus_{i \in I} \rho^{(i)}$. The diagnostic r (defined in section 2.5) is plotted on the y-axis with a log scale for each subfigure. The labelling of group representations is explained in section 5.1, recall that the primed integers indicate learned representations. The first row demonstrates by Schur's Lemma that to within numerical error of about $\sim 10^{-6}$ the learned $\mathrm{SO}(3)$ group representation denoted $1'$ is isomorphic to the spin-1 irreducible group representation obtained from known formulae, i.e. $1'_{\mathrm{SO}(3)} \cong 1_{\mathrm{SO}(3)}$. The first row also indicates that $1'_{\mathrm{SO}(2,1)} \cong 1_{\mathrm{SO}(2,1)}$, and $(1/2', 1/2')_{\mathrm{SO}(3,1)} \cong (1/2, 1/2)_{\mathrm{SO}(3,1)}$. The remaining rows indicate that the tensor product structure of the learned group representations matches that of the known irreducible group representations.

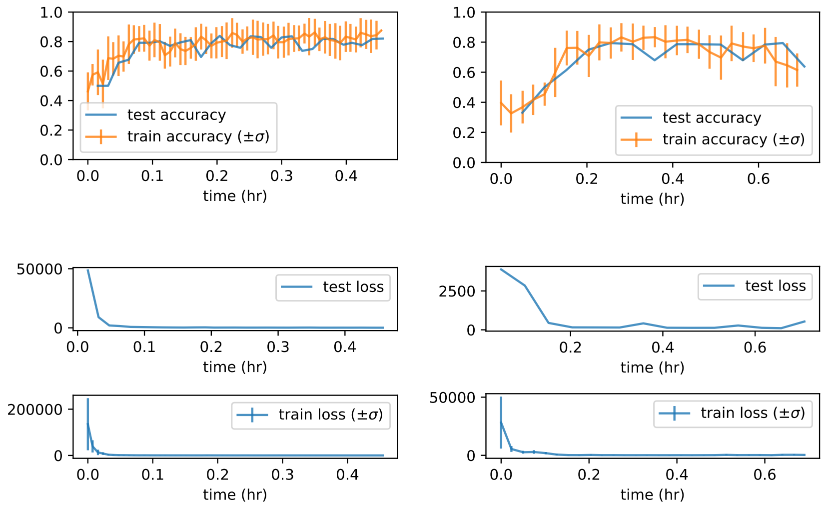

Figure 4: (Left) $\mathrm{SO}(2,1)$-equivariant neural network learning to recognize digits from the MNIST-Live dataset in 2 spatial dimensions. Error bars for train accuracy and loss are computed as the mean and standard deviation across a sliding window of 15 batches. (Right) $\mathrm{SO}(3,1)$-equivariant neural network training to recognize digits from the MNIST-Live dataset in 3 spatial dimensions. Error bars for train accuracy and loss are computed as the mean and standard deviation across a sliding window of 15 batches.

```
from lan import LieAlgebraRepresentation, \
    LieAlgebraRepresentationDirectSum, \
    LieAlgebraTensorProductRepresentation, \
    LieGroupEquivariantNeuralNetwork

learned_generators = [...]
known_generators = [...]

learned_irrep = LieAlgebraRepresentation(learned_generators)
scalar_irrep = LieAlgebraRepresentation(
    numpy.zeros((
        learned_irrep.algebra.dim, 1, 1
    ))
)
known_irrep = LieAlgebraRepresentation(known_generators)

representations = LieAlgebraRepresentationDirectSum([
    scalar_irrep,
    known_irrep
    learned_irrep,
    LieAlgebraTensorProductRepresentation(
        [learned_irrep, learned_irrep])
])

model = LieGroupEquivariantNeuralNetwork(
    representations, num_layers=10, num_channels=32)
```

Figure 5: Our Lie Algebraic Networks (lan) module handles Lie algebra and Lie group representations, derives Clebsch-Gordan coefficients for the equivariant layer update, and computes the forward pass. This makes it simple to build an equivariant point cloud network once the representations are obtained.