LEARNING IRREDUCIBLE REPRESENTATIONS OF NONCOMMUTATIVE LIE GROUPS

Anonymous authors
Paper under double-blind review

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 non-linearities. 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 Appendix B, 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 ranging 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 $\alpha, \beta \in G$ which can be composed together using the 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 $\rho: G \rightarrow \mathbb{R}^{n \times n}$ 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 B.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.

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 for which LearnRep performs well is the Lorentz group of special relativity. Prior work has applied Lorentz-equivariant models to particle physics. As we will explain, 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 which guarantees motion equivariance for object-tracking tasks.

We have provided complete software implementations of LearnRep and SpacetimeNet. Our work contributes towards a general framework and toolset for building neural networks equivariant to novel Lie groups, and motivates further study of Lorentz equivariance for object tracking.

2 Related Work

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. Here 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 SO(3), SO(2, 1), and SO(3, 1). Cohen & Welling (2014) proposed a 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 SO(3), SO(2, 1), and 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 other Lie groups where the representation theory is not well-understood.

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 SE(2) (Worrall et al., 2017; Weiler & Cesa, 2019), 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), 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 this work, Bogatskiy et al. (2020) constructed a Lorentz-equivariant model which operated on irreducible representations of the Lorentz group. That work obtained the representations and Clebsch-Gordan coefficients analytically and applied the resulting model to particle physics. That approach relies on the existing representation theory of the group SO(3, 1). In contrast, our techniques may be applied to Lie groups without such a theory. Another concurrent work by Finzi et al. (2020) proposed a framework for building models equivariant to arbitrary Lie groups. That 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, and demonstrating a novel application of Poincaré equivariance to object tracking.

3 EXPERIMENTS

3.1 CONVERGENCE OF LEARNREP TO IRREDUCIBLE REPRESENTATIONS

We apply LearnRep to SO(3), SO(2, 1), and SO(3, 1) to learn 3, 3, and 4 dimensional irreducible representations respectively. The loss function converges 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 Appendix C.1.1. 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.

3.2 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 SO(2, 1) and SO(3, 1)-equivariant SpacetimeNet models with 3 channels and batch size 16 on 4096 MNIST-Live examples and evaluate on a test set of 124 examples. We obtain test accuracy of 80 ± 5%. Further details and source code are available in the Appendix.

As we mention in Section 1, such Lie groups include important cases like HG($m,n$).
Figure 1: Tensor product structure of the learned group representations $\rho$ with several known (analytically-derived) group representations $\rho_1$ for the groups $SO(3)$, $SO(2,1)$, and $SO(3,1)$. Each column is for the group indicated at the bottom, each row is for a different choice of $\rho_1$ for that group, and the horizontal axis indicates the $\rho^{(i)}$ onto which we project the tensor product $\rho \otimes \rho_1 \cong \bigoplus_{i \in I} \rho^{(i)}$. The diagnostic $r$ (defined in Appendix B.3) is plotted on the $y$-axis with a log scale for each subfigure. The labelling of group representations is explained in Section 3.1, 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 $SO(3)$ group representation denoted $1'$ is isomorphic to the spin-1 irreducible group representation obtained from known formulae, i.e. $1'_{SO(3)} \cong 1_{SO(3)}$. The first row also indicates that $1'_{SO(2,1)} \cong 1_{SO(2,1)}$, and $(1/2',1/2')_{SO(3,1)} \cong (1/2,1/2)_{SO(3,1)}$. The remaining rows indicate that the tensor product structure of the learned group representations matches that of the known irreducible group representations.

3.3 Conclusion

We envision many applications of Poincaré-equivariant deep neural networks beyond the physics of particles and plasmas. SpacetimeNet can identify and track objects as they move through 3D space while maintaining equivariance to the motion and coordinate system of the observer. This suggests that Lorentz-equivariance may serve as a useful prior for object-tracking tasks, especially when the observer’s velocity is not known. 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 would enable the construction of networks equivariant to other 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


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


A APPENDIX

B TECHNICAL BACKGROUND

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

B.1 SYMMETRY GROUPS SO(n) AND 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 $\langle \vec{u}, \vec{v} \rangle = \sum_{i=1}^{3} u_i v_i$:

(i) $\det A = 1$ (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 $SO(3)$. This definition directly generalizes to the $n-$dimensional rotation group $SO(n)$. For $n \geq 3$, the group $SO(n)$ is noncommutative, meaning there are elements $A, B \in SO(n)$ such that $AB \neq BA$. Allowing for rotations and translations of $n$ dimensional space gives the $n-$dimensional special Euclidean group $SE(n)$. $SO(n)$ is generalized by a family of groups denoted $SO(m, n)$, with $SO(n) = SO(n, 0)$. 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 $SO(m, n)$ is the set of matrices $A \in \mathbb{R}^{(m+n) \times (m+n)}$ satisfying (i-ii) below:

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

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

B.2 ACTION OF SO(m, n) ON SPACETIME

We now explain the physical relevance of the groups $SO(m, n)$ by reviewing spacetime. We refer to [Feynman et al. (2011) (ch. 15)] for a pedagogical overview. Two observers who are moving at different velocities may disagree on the observed 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 $SE(3)$), a temporal translation (synchronizing their clocks), and finally applying a transformation of the following form:

$t_i \mapsto t_i \quad \vec{u}_i \mapsto \vec{u}_i + \vec{v} t_i$, \hspace{1cm} (1)

in which $\vec{v}$ is the relative velocity of the observers. The transformation equation is called a Galilean boost. The set of all Galilean boosts along with 3D rotations forms the homogeneous Galilean group denoted $HG(1, 3)$. Einstein argued that equation 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 $\ell_2$ norm of $\vec{v}$. The resulting coordinate transformation is called a Lorentz boost, and an example of its effect is shown in figure. The set of 3D rotations along with Lorentz boosts is exactly the group $SO(3, 1)$. In the case of 2 spatial dimensions, the group is $SO(2, 1)$. Including spacetime translations along with the Lorentz group $SO(n, 1)$ gives the larger Poincaré group $P_n$ with $n$ spatial dimensions. The Poincaré group $P_3$ 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 $P_n$ 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 $P_n$-equivariant. For low velocities the symmetries of the homogeneous Galilean groups $HG(n, 1)$ provide a good approximation to $SO(n, 1)$ symmetries, so Galilean-equivariance may be sufficient for some tasks. Unfortunately the representations of $HG(n, 1)$ are not entirely understood [De Montigny et al. (2006); Niederle & Nikitin (2006); Levy-Leblond (1971)].
Figure 2: 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 $±x$ 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.

B.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 $\mathfrak{g}$ as its tangent space at the identity. This is a vector space $V$ along with a bilinear product called the Lie bracket $\left[ a, b \right]$ which must behave like the commutator for an associative ring $R$ with multiplication operation $\times_R$:

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

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

$$\left[ J_i, J_j \right] = \epsilon_{ijk} J_k,$$  \hspace{1cm} (2)

in which $\epsilon_{ijk} \in \{±1, 0\}$ is the totally antisymmetric Levi-Civita symbol.\footnote{Specifically, the Lie bracket must satisfy the Jacobi identity and $\left[ a, a \right] = 0$.} 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 $\theta$ satisfy $R_x R_y - R_y R_x = R_z + O(\theta^2)$; more generally the Lie bracket of equation 8 is satisfied to first order in $\theta$. The Lie algebra $\mathfrak{so}(3,1)$ of the Lorentz Group $SO(3,1)$ also satisfies equation 8 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:

$$\left[ J_i, K_j \right] = \epsilon_{ijk} K_k \quad \left[ K_i, J_j \right] = -\epsilon_{ijk} J_k,$$  \hspace{1cm} (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 $T_1, \ldots, T_t$ such that

$$\left[ T_i, T_j \right] = \sum_{k=1}^{t} A_{ijk} T_k,$$  \hspace{1cm} (4)

we call the tensor $A_{ijk}$ the structure constants of $\mathcal{A}$. For connected matrix Lie groups such as $SO(m,n)$, the structure constants $A_{ijk}$ are easily obtained. For example, one may apply the matrix logarithm to several elements of the group to obtain elements of the algebra, then find a complete basis for the algebra and write the commutator of all basis elements in this basis.

B.4 GROUP REPRESENTATIONS AND THE TENSOR PRODUCT

Let $G$ be a Lie group and $\rho : G \rightarrow \mathbb{R}^{n×n}$ be a representation of $G$ as defined in Section 1.1. Then $\rho$ 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

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

\footnote{The symbol $\epsilon_{ijk}$ simply expresses in equation 8 that $\left[ J_1, J_2 \right] = J_3, \left[ J_2, J_3 \right] = J_1$, and $\left[ J_3, J_1 \right] = J_2$.}
using the matrix product. We then say that \( \rho \) is irreducible if it leaves no nontrivial subspace invariant – for every subspace \( V \subset \mathbb{R}^n \) with \( 0 < \dim V < n \), there exists \( \alpha \in G, \bar{v} \in V \) such that \( \alpha \ast \rho \bar{v} \notin V \).

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

\[
\rho_1 \otimes \rho_2 : G \rightarrow \mathbb{R}^{n_1 n_2 \times n_1 n_2}, \quad (\rho_1 \otimes \rho_2)(\alpha) = \rho_1(\alpha) \otimes \rho_2(\alpha),
\]

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 \( A_1, A_2 \in \mathbb{R}^{n_1 \times n_1} \) 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 \( \rho_1, \rho_2 \) as above we also define their direct sum as

\[
(\rho_1 \oplus \rho_2)(\alpha) = \begin{pmatrix} \rho_1(\alpha) \\ \rho_2(\alpha) \end{pmatrix}.
\]

For two groups \( H, G \) we say that \( H \) is isomorphic to \( G \) and write \( H \cong G \) if there exists a bijection \( f : H \rightarrow G \) such that \( f(\alpha \beta) = f(\alpha)f(\beta) \). For \( \rho_1, \rho_2 \) as above, their images \( \rho_1(G) \) form matrix groups and we say that \( \rho_1 \) and \( \rho_2 \) 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 \( \text{SO}(3) \) 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 \( \text{SO}(n, 1) \) and \( \text{SO}(n) \), 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.

### B.5 Clebsch-Gordan Coefficients and Tensor-Product Nonlinearities

**Clebsch-Gordan Coefficients:** Let \( G \) be a completely reducible Lie group and let \( \rho_1, \rho_2, \rho_3 \) 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 \( \rho_1 \otimes \rho_2 \). As \( G \) 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 \( \rho_3 \in S \). 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 \( \rho_3 \) from the tensor product space \( \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \). That is,

\[
\forall (\alpha, \bar{u}, \bar{v}) \in G \times \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}, \\
C(\rho_1(\alpha) \otimes \rho_2(\alpha))(\bar{u} \otimes \bar{v}) = \rho_3(\alpha)C(\bar{u} \otimes \bar{v})
\]

\[
\Rightarrow C(\rho_1(\alpha) \otimes \rho_2(\alpha)) = \rho_3(\alpha)C.
\]

The matrices \( C \) satisfying equation (5) for various \( \rho_3 \) 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 \( C[\rho_1, \rho_2, \rho_3] \).

**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 \( \text{SE}(3) \) by [Thomas et al., 2018], [Kondor et al., 2018] and for \( \text{SO}(3, 1) \) by [Bogatskiy et al., 2020].

### B.6 Equivariant Convolutions

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

\[
\bar{u}^\prime_i = \sum_j \kappa(\bar{x}_j - \bar{x}_i)\bar{u}_j
\]
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 \( \alpha \in \text{SO}(3, 1) \),

\[
\rho_v(\alpha) \sum_j \kappa(\vec{x}_j - \vec{x}_i) \vec{u}_j = \sum_j \kappa(\rho_0(\alpha)(\vec{x}_j - \vec{x}_i)) \rho_u(\alpha) \vec{u}_j
\]

\[
\Rightarrow \kappa(\Delta \vec{x}) = \rho_v(\alpha^{-1}) \kappa(\rho_0(\alpha) \Delta \vec{x}) \rho_u(\alpha) \quad (6)
\]

Therefore a single kernel matrix in \( \mathbb{R}^{n \times m} \) may be learned for each coset of spacetime under the action of \( \text{SO}(3, 1) \). 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 \( 6 \) by computing an \( \alpha \) 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 \( \text{SO}(3, 1) \) acting upon \( \mathbb{R}^4 \) is the set of points \( \{(t, 0, 0, 0) : t \in \mathbb{R}^+ \} \cup \{(0, x, 0, 0) : x \in \mathbb{R} \} \cup \{(t, ct, 0, 0) : t \in \mathbb{R}^+ \} \).

**B.7 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 3 has a function “jmat” that readily gives the representation matrices. A formula to obtain real-valued representation matrices is given in 7 and a software implementation is available at 8. The three-dimensional Lie algebra \( \mathfrak{so}(2, 1) = \text{span}\{K_x, K_y, J_z\} \) has structure constants given by equation 7

\[
\begin{align*}
[J_i, J_j] &= \epsilon_{ijk} K_k & [K_i, K_j] &= -\epsilon_{ijk} J_k
\end{align*}
\]

(7)

In fact, these three generators \( K_x, K_y, J_z \) may be rescaled so that they satisfy equation 8 instead, the structure constants of \( \mathfrak{so}(3) \):

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

(8)

This is due to the isomorphism \( \mathfrak{so}(3) \cong \mathfrak{so}(2, 1) \). Specifically, let \( \{L_x, L_y, L_z\} \) denote a Lie algebra representation of \( \mathfrak{so}(3) \), defining

\[
K_x = -iL_x \quad K_y = -iL_y \quad J_z := L_z,
\]

it may be easily checked that \( K_x, K_y, J_z \) satisfy the applicable commutation relations from equation 7. 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 7, we define new generators \( A_i, B_i \) as

\[
A_i := \frac{1}{2}(J_i + iK_i), \quad B_i := \frac{1}{2}(J_i - iK_i),
\]

(9)

we see that the \( \mathfrak{so}(3, 1) \) commutators equation 8 become

\[
[A_i, A_j] = i\epsilon_{ijk} A_k, \quad [B_i, B_j] = i\epsilon_{ijk} B_k, \quad [A_i, B_j] = 0.
\]

(10)

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 \( \mathfrak{so}(3) \).

**C Methods**

**C.1 Learning Lie Group Representations**

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

\[
|M|_F^2 = \sum_{1 \leq i, j \leq n} |M_{ij}|^2, \quad |M|_1 = \sum_{1 \leq i, j \leq n} |M_{ij}|.
\]
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 C.1. 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 $T_i$:

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

This is the magnitude of violation of the structure constants of $\mathcal{A}$, multiplied by a norm penalty term $N[T_i]^{-1}$. The purpose of the norm penalty is to avoid convergence to a solution where $T_i = 0_{n \times n}$ for any $i$, which will act trivially when restricted to the nontrivial subgroup $\{e^{tT_i} : t \in \mathbb{R}\}$. 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 was decreased exponentially when loss plateaued. Plots of the loss and penalty during training are available in the Appendix.

C.1.1 Verifying Irreducibility of Learned Representations

Suppose LearnRep has converged to $T_1, ..., T_t$ 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 there exists an $n$-dimensional representation $\rho$ of the Lie group $G$ which is defined by

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

in which each $\alpha \in G$ is uniquely determined by constants $b_1^\alpha, ..., b_t^\alpha \in \mathbb{R}$, and $\exp$ denotes the matrix exponential. The constants $b_i^\alpha$ are independent of $n$ and may be obtained from a single representation using the matrix logarithm. Throughout this section, $\rho$ denotes the above representation. In general $\rho$ may leave some nontrivial subspace invariant. In this case it is reducible and splits as the direct sum of lower-dimensional irreducible representations $\rho_i$ as explained in [B4].

$$
\rho \cong \rho_1 \oplus \ldots \oplus \rho_t.
$$

Recall that any representation may be obtained as such a direct sum of irreducible representations with dimensions $n_1, ..., n_\ell$ satisfying $n = \sum_{i=1}^\ell n_i$. If $n$ is set to the minimum dimension of a nontrivial irreducible representation, the only permissible partitions of $n$ have $\ell = 1$ and $\ell = n$ – as the latter representation is trivial, equation C.11 diverges, so LearnRep can only converge to an irreducible $n$ dimensional representation $\rho$. It is important to verify that the learned $\rho$ is indeed irreducible with $\ell = 1$. To validate that $\rho$ 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 $\rho$ with several other known representations $\rho_1, ..., \rho_\ell$. Section B.5 defines these coefficients and explains how they are computed from the nullspace of the matrix $C = C[\rho, \rho_1, \rho_2]$, in which $\rho_2$ appears in the decomposition of $\rho \otimes \rho_1$. Let $\rho_3, \rho_2$ 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 $C$ indicates the number of linearly independent matrices $C$ of Clebsch-Gordan coefficients. The singular values of $C$ are denoted $SV_1(C) \leq \ldots \leq SV_t(C)$. The ratio

$$
r(C) := SV_2(C)/SV_1(C)
$$

This applies to our experiments learning $SO(3)$ representations, with $n = 3$. 

\[11\]
We now prove that SpacetimeNet is $\mathcal{P}_n$-equivariant. Consider an arbitrary Poincarë group transformation $\alpha \in \mathcal{P}_n$, and write $\alpha = \beta t$ in which $\beta \in \text{SO}(n, 1)$ and $t$ is a translation. Suppose we apply this $\alpha$ to the inputs of equation (13) through the representations indexed by $q$: $\rho_q(\alpha)_{st}$, in which $s, t$ index the representation matrices. Then since the translation $t$ leaves $\Delta X$ invariant, the resulting filters will be

\[ F_{xijqr}^{k+1} = \sum_{g,l,s,m,t,d,j} C_{g,qr,ls,mt} F_{xijls}^{k} V_{xjmdt}^{k} W_{qcgd}^{k}. \]
\[ F_{xijqr}^k = \delta_{qq'} \sum_{r'} \rho_{q'}(\beta)_{rr'} \Delta X_{xijr'} + \]

\[ \sum_{s,t,g} C_{g,qr,qs,qt} f_{qq}^k \sum_{s',t'} \rho_{q'}(\beta)_{ss'} \Delta X_{x js's'} \rho_{q'}(\beta)_{tt'} \Delta X_{x jt't'} \]

\[ = \delta_{qq'} \sum_{r'} \rho_{q'}(\beta)_{rr'} \Delta X_{xijr'} + \]

\[ \sum_{s,t,g,r'} \left( \sum_{s,t} C_{g,qr,qs,qt} \rho_{q'}(\beta)_{ss'} \rho_{q'}(\beta)_{tt'} \right) f_{qq}^k \Delta X_{x js's'} \Delta X_{x jt't'} \]

\[ = \delta_{qq'} \sum_{r'} \rho_{q'}(\beta)_{rr'} \Delta X_{xijr'} + \]

\[ \sum_{s,t,g,r'} (\rho_{q}(\beta)_{rr'} C_{g,qr',qs,qt}) f_{qq}^k \Delta X_{x js's'} \Delta X_{x jt't'} \]

\[ = \sum_{r'} \rho_{q'}(\beta)_{rr'} F_{xijqr'}^k, \]

where we have used equation 5. The network will be equivariant if each layer update is equivariant.

Recall the layer update rule of equation 14:

\[ V_{xijqr}^{k+1} = \sum_{g,l,s,m,t,d,j} C_{g,qr,ls,mt} F_{xijls}^k V_{xjmdt}^k W_{qcgd} W_{xijqr}. \]

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

\[ V_{xijqr}^{k+1} = \sum_{g,l,s,m,t,d,j} C_{g,qr,ls,mt} \sum_{s,t} \rho_m(\beta)_{ss'} F_{xijls'}^k \times \]

\[ \sum_{r'} \rho_m(\beta)_{tt'} V_{xjmdt}^k W_{qcgd}^k \]

\[ = \sum_{s',t'} \sum_{g,l,s,m,t,d,j} (C_{g,qr,ls,mt} \rho_m(\beta)_{ss'} \rho_m(\beta)_{tt'}) \times \]

\[ F_{xijls'}^k V_{xjmdt}^k W_{qcgd}^k \]

\[ = \sum_{g,l,s,m,t,d,j} (\rho_m(\beta)_{rr'} C_{g,qr',ls,mt}) F_{xijls'}^k V_{xjmdt}^k W_{qcgd}^k \]

\[ = \sum_{r'} \rho_m(\beta)_{rr'} V_{xijqr'}^{k+1}, \]

where again we applied equation 5.

C.4 Supplementary figures
Figure 3: Convergence to arbitrary precision group representations of three Lie groups: SO(3), SO(2, 1), and 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.
Figure 4: (Left) 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) 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.

```python
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.