A Practical Method for Constructing Equivariant Multilayer Perceptrons for Arbitrary Matrix Groups

Marc Finzi, Max Welling, Andrew Gordon Wilson

ICML-2021

Presented by Ozgur Guldogan

Presentation Outline

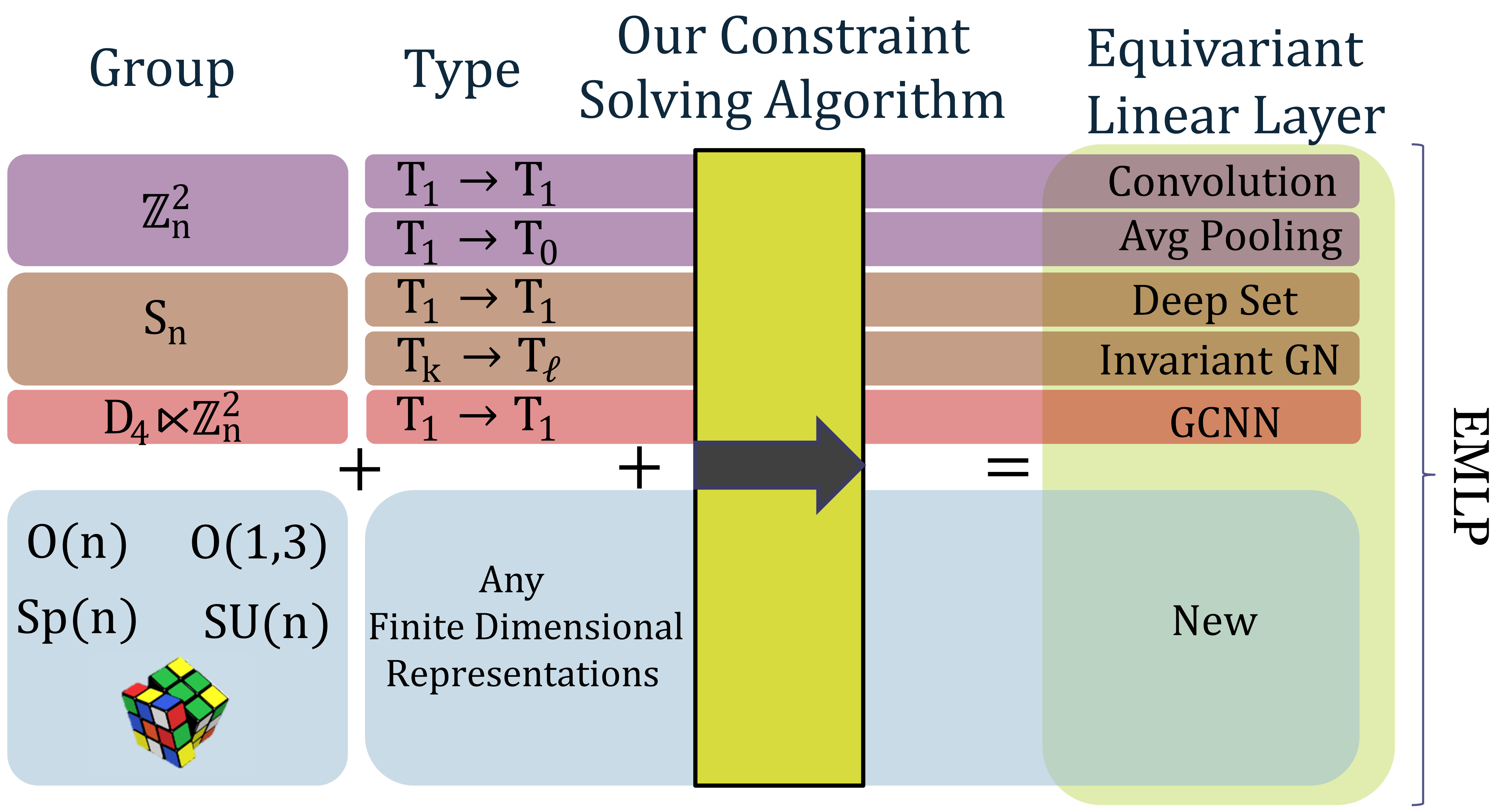
- Introduction and Motivation
- Related Works
- Background
- Methods
- Results
- Conclusion

Introduction and Motivation

- Symmetries and equivariance are crucial for neural network generalization
- Current focus: limited groups (translation, rotation, permutation)
- Challenge: Extend equivariance to a broader set of matrix groups and propose a general method for arbitrary matrix groups

Contributions

- Provided a completely general algorithm for solving for the equivariant layers of matrix groups, including continuous groups.



Contributions

- Established that equivariance conditions for matrix groups with arbitrary linear representations can be expressed as M + D constraints, where M is the number of discrete generators, and D is the dimension of the group.

- Introduced a polynomial time algorithm for solving these constraints in the case of finite-dimensional representations, with an emphasis on acceleration through structure exploitation and reformulation as an optimization problem.

- Developed the Equivariant MultiLayer Perceptron (EMLP) with a bilinear layer, showcasing its versatility by successfully applying it to previously challenging groups such as O(5), O(1, 3), Sp(n), and the Rubik's cube group, consistently outperforming non-equivariant baselines.

Related Works

- **Early Approaches to Equivariance:**
    - Initial attempts at equivariance focused on translation equivariance in convolutional neural networks (LeCun et al., 1989).
    - Cohen & Welling (2016a) extended the concept to discrete groups with GCNNs, paving the way for more general group equivariant neural networks.
- **Generalization to Rotation Groups:**
    - Works by Cohen & Welling (2016b), Esteves et al. (2017), Marcos et al. (2017), and others generalized the approach for continuous rotation groups like SO(2), O(2), SO(3), O(3), and their discrete subgroups.
    - Limitations arose from the complexity of working with irreducible representations, restricting the applicability to a few rotation groups and the Lorentz group $SO^{+}(1, 3)$ (Bogatskiy et al., 2020).
- **Alternative Approaches for Equivariance:**
    - Some researchers explored alternative approaches, such as group FFTs (Cohen et al., 2018b) and regular group convolution (Worrall & Welling, 2019; Bekkers, 2019; Finzi et al., 2020), providing greater flexibility.

Related Works

- **Advancements in Equivariance to Permutation Groups:**
    - Equivariance to the permutation group S_n advanced significantly for applications to sets, graphs, and related objects (Zaheer et al., 2017; Maron et al., 2018; Serviansky et al., 2020).
    - Instances of equivariant networks were shown to be universal, approximating equivariant functions for a given group with arbitrary accuracy (Maron et al., 2019; Ravanbakhsh, 2020; Dym & Maron, 2020).
- **Current Challenges and Attempts at Generality:**
    - Noteworthy efforts by Lang & Weiler (2020), Ravanbakhsh et al. (2017), and van der Pol et al. (2020) aim at generality, but each has limitations related to mathematical complexity, representation types, or scalability to large or continuous groups.

Background

- **Finite Groups and Discrete Generators**
    - A group G is finitely generated if we can write each element $g \in G$ as a sequence from a discrete set of generators $\{h_1, h_2, \ldots h_M\}$ and their inverses $h_{-k} = h_k^{-1}$. For example, we may have an element $g = h_1 h_2 h_2 h_1^{-1} h_3$ that can be written more compactly $g = \Pi_{i=1}^{N} h_{k_i}$for the integer sequence $k = [1,2,2,-1,3]$.
    - Even for large groups, the number of generators is *much* smaller than the size of the group: 1 for $\mathbb{Z}_n$ of size n, 2 for S_n of size n!, and 6 for the cube group of size 4×10^{19}.

Background

- **Continuous Groups and Infinitesimal Generators.**
  - Similarly, Lie theory provides a way of analyzing continuous groups in terms of their *infinitesimal* generators. The Lie Algebra $\mathfrak{g}$ of a Lie Group G (a continuous group that forms a smooth manifold) is the tangent space at the identity $\mathfrak{g} := T_{\mathrm{id}}G \subseteq \mathbb{R}^{n \times n}$, which is a vector space of infinitesimal generators of group transformations G. The exponential map $\exp : \mathfrak{g} \to G$ maps back to the Lie Group and can be understood through the series: $\exp(A) = \sum_{k=0}^{\infty} A^k/k!$.

Background

- **Continuous Groups and Infinitesimal Generators.**
  - A classic example is the rotation group $G = \mathrm{SO}(n)$ with matrices $\mathbb{R}^{n \times n}$ satisfying $R^\top R = I$ and $\det(R) = 1$. Parametrizing a curve $R(t)$ $R(0) = I, R'(0) = A$, one can find the tangent space by differentiating the constraint at the identity. The Lie Algebra consists of antisymmetric matrices: $\mathfrak{so}(n) = T_{\mathrm{id}}\mathrm{SO}(n) = \{A \in \mathbb{R}^{n \times n} : A^\top = -A\}$.

Background

- **Group Representations.**
  - In the machine learning context, a group element is most relevant in how it acts as a transformation on an input. A (linear finite dimensional) group *representation* $\rho : G \to \mathrm{GL}(m)$ associates each $g \in G$ to an invertible matrix $\rho(g) \in \mathbb{R}^{m \times m}$ that acts on $\mathbb{R}^m$. The representation satisfies $\forall g_1, g_2 \in G : \rho(g_1 g_2) = \rho(g_1)\rho(g_2)$, and therefore also $\rho(g^{-1}) = \rho(g)^{-1}$.

Background

- **Lie Algebra Representations.**
  - Mirroring the group representations, Lie Groups have an associated representation of their Lie algebra, prescribing how infinitesimal transformations act on an input. A Lie algebra representation $d\rho : \mathfrak{g} \to \mathfrak{gl}(N)$ is a linear map from the Lie algebra to $m \times m$ matrices.
  - An important result in Lie Theory relates the representation of a Lie Group to the representation of its Lie Algebra

$$\forall A \in \mathfrak{g} : \quad \rho(e^A) = e^{d\rho(A)}$$

Background

- **Tensor Representations.**
  - Given some base group representation ρ, Lie Algebra representation $d\rho$, acting on a vector space V, representations of increasing size and complexity can be built up through the tensor operations dual *, direct sum $\bigoplus$, and tensor product $\bigotimes$.

OP	ρ	$d\rho$	V
$*$	$\rho(g^{-1})^\top$	$-d\rho(A)^\top$	V^*
$\oplus$	$\rho_1(g) \oplus \rho_2(g)$	$d\rho_1(A) \oplus d\rho_2(A)$	$V_1 \oplus V_2$
$\otimes$	$\rho_1(g) \otimes \rho_2(g)$	$d\rho_1(A) \bar{\oplus} d\rho_2(A)$	$V_1 \otimes V_2$

  - Acting on matrices, $\bigoplus$ is the direct sum that concatenates the matrices on the diagonal $X \bigoplus Y = \begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix}$, and facilitates multiple representations which are acted upon separately. The $\bigotimes$ on matrices is the Kronecker product, and $\bar{\bigoplus}$ is the Kronecker sum: $X \bar{\bigoplus} Y = X \otimes I + I \otimes Y$. V^* is the dual space of V. The tensor product and dual are useful in describing linear maps from one vector space to another. Linear maps from $V_1 \to V_2$ form the vector space $V_2 \otimes V_1^*$ and have the corresponding representation $\rho_2 \otimes \rho_1^*$.

Methods

- **The Equivariance Constraint**
  - All equivariant linear layers $W \in \mathbb{R}^{N_2 \times N_1}$ that map from one vector space V_1 with representation ρ_1 to another vector space V_2 with representation ρ_2 for a matrix group G
  - Equivariance requires that transforming the input is the same as transforming the output:

$$\forall x \in V_1, \forall g \in G: \qquad \rho_2(g)Wx = W\rho_1(g)x\,.$$

  - Since true for all x, $\rho_2(g)W\rho_1(g)^{-1} = W$, or more abstractly:

$$\forall g \in G: \quad \rho_2(g) \otimes \rho_1(g^{-1})^\top \mathrm{vec}(W) = \mathrm{vec}(W)$$

    where vec flattens the matrix into a vector. $\rho_1(g^{-1})^\top$ is the dual representation $\rho_1^*(g)$, and so the whole object $\rho_2(g) \otimes \rho_1(g^{-1})^\top = (\rho_2 \otimes \rho_1^*)(g) = \rho_{21}(g)$ is a representation of how g acts on matrices mapping from $V_1 \to V_2$.

Methods

- **General Solution for Symmetric Objects**
    - The equivariant constraint with $\rho = (\rho_2 \otimes \rho_1^*)$ is a special case of a more general equation expressing the symmetry of an object v,

$$\forall g \in G : \quad \rho(g)v = v$$

    - Writing the elements of G in terms of their generators: $g = \exp(\sum_i^D \alpha_i A_i)\Pi_{i=1}^N h_{k_i}$. For group elements with $k = \emptyset$, we have

$$\forall \alpha_i : \quad \rho\big(\exp(\sum_i \alpha_i A_i)\big)v = v$$

    - Using the Lie Algebra - Lie Group representation correspondence and the linearity of $d\rho(\,\cdot\,)$ we have

$$\forall \alpha_i : \quad \exp\big(\sum_i \alpha_i d\rho(A_i)\big)v = v\,.$$

Methods

- **General Solution for Symmetric Objects**
  - Taking the derivative with respect to α_i at $\alpha = 0$, we get a constraint for each of the infinitesimal generators

$$\boxed{\forall i = 1,\ldots,D: \quad d\rho(A_i)v = 0}$$

  - For group elements with all $\alpha_i = 0$ and $N = 1$, we get an additional constraint for each of the discrete generators in the group:

$$\boxed{\forall k = 1,\ldots,M: \quad (\rho(h_k) - I)v = 0.}$$

  - A total of $O(M + D)$ constraints, one for each of the discrete and infinitesimal generators.
  - Proved that these reduced constraints are not just **necessary** but also **sufficient**, and therefore characterized all solutions to the symmetry equation.

Methods

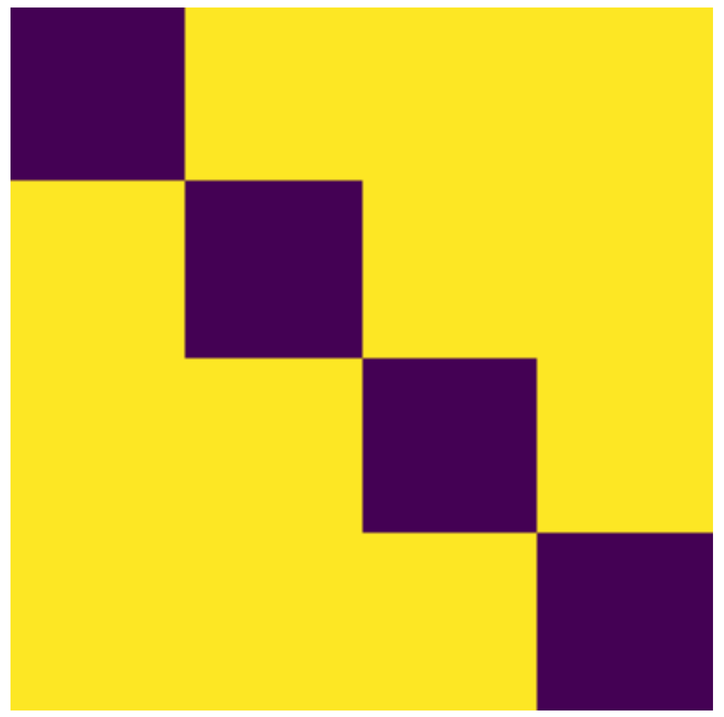
- **Solving the Constraint.**
  - Collect each of the symmetry constraints $C_1 = d\rho(A_1), C_2 = d\rho(A_2), \ldots, C_{D+1} = \rho(h_1) - I, \ldots$ into a single matrix C, which we can break into its nullspace spanned by the columns of $Q \in \mathbb{R}^{m \times r}$ and orthogonal complement $P \in \mathbb{R}^{m \times (m-r)}$ using the singular value decomposition:

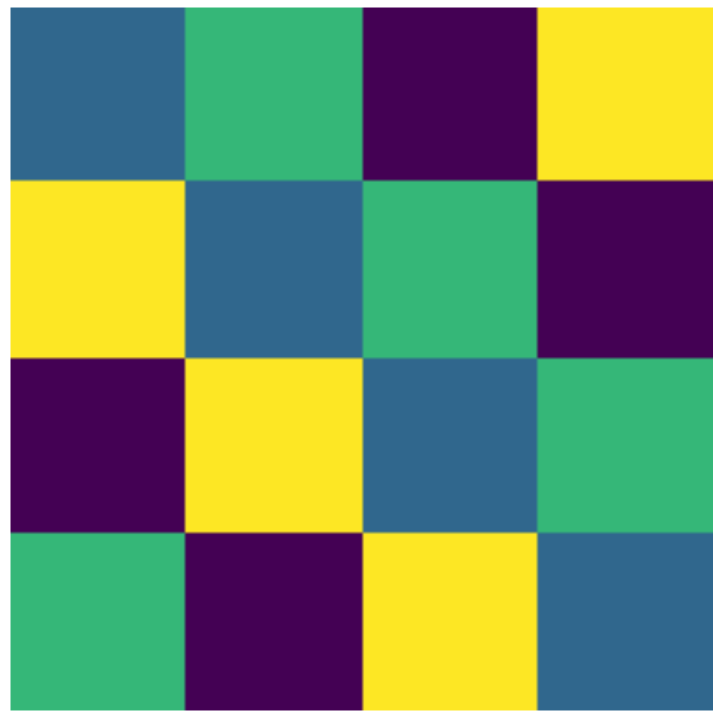
$$Cv = \begin{bmatrix} d\rho(A_1) \\ d\rho(A_2) \\ \cdots \\ \rho(h_1) - I \\ \cdots \end{bmatrix} v = U \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P^\top \\ Q^\top \end{bmatrix} v = 0.$$

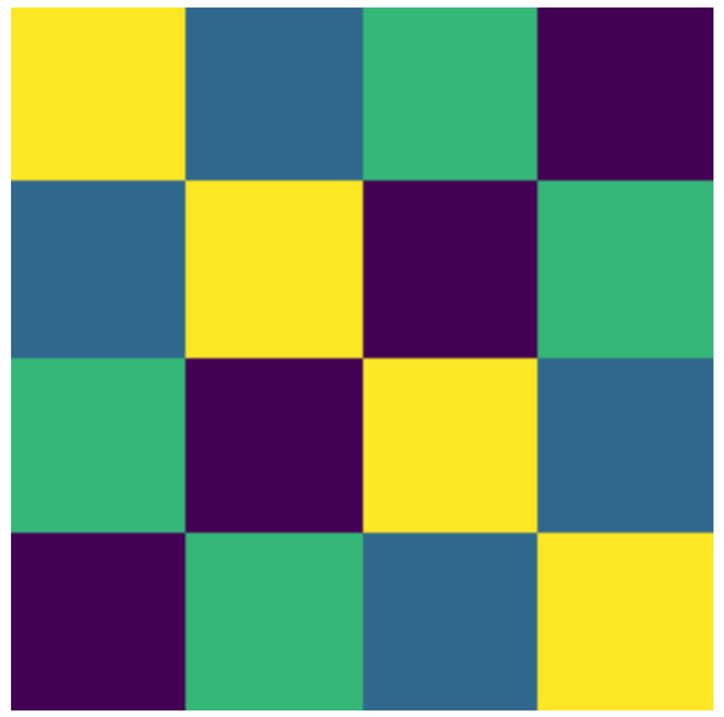
  - All symmetric solutions for v must lie in the nullspace of C: $v = Q\beta$ for some coefficients β, and we can then parametrize all symmetric solutions directly in this subspace. Alternatively, defining $\beta = Q^T v_0$ we can reuse any standard parametrization and initialization, but simply project onto the equivariant subspace: $v = QQ^T v_0$.
  - Thus given any finite dimensional linear representation, we can solve the constraints with a singular value decomposition. If $v \in \mathbb{R}^m$ the runtime of the approach is $O((M + D)m^3)$.

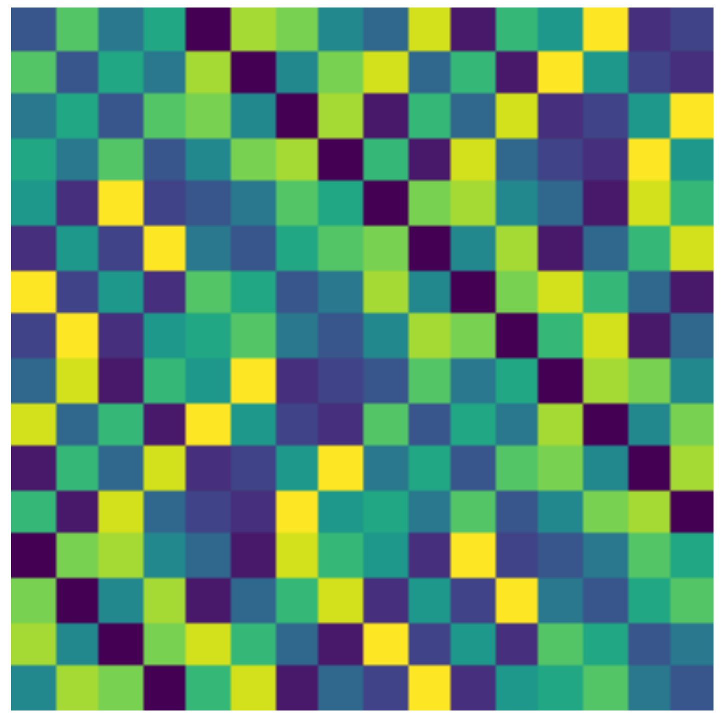
Methods

- **A Unifying Perspective on Equivariance**


(a) S_4


(b) $\mathbb{Z}_4$

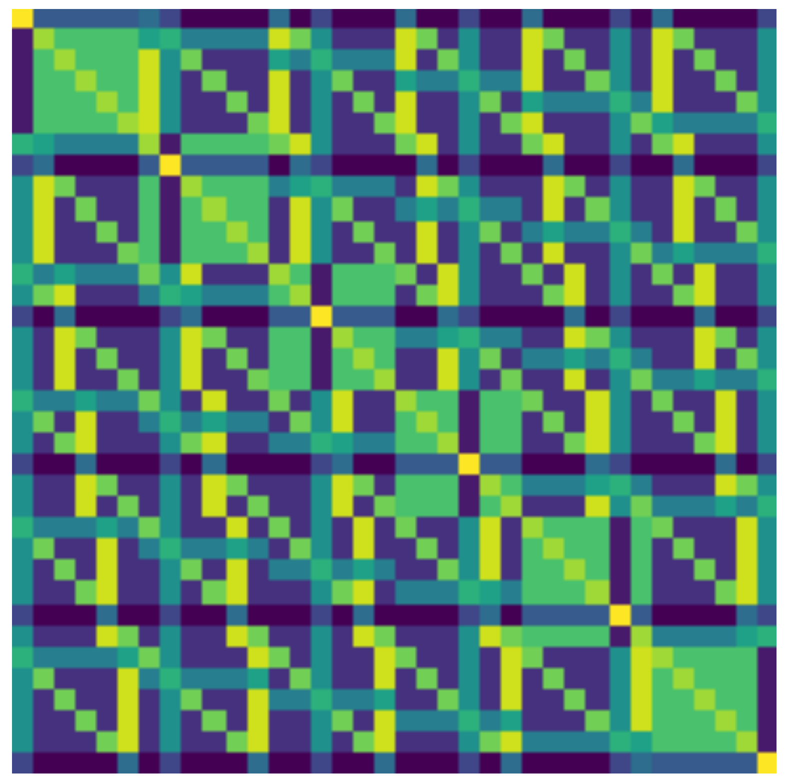

(c) $\mathbb{Z}_2^2$

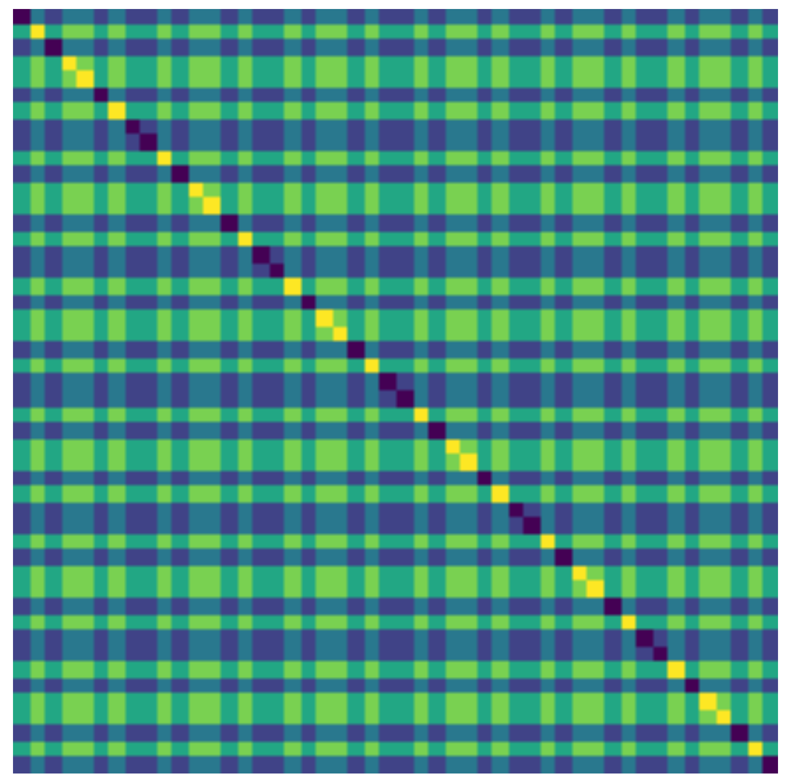

(d) $\mathbb{Z}_4 \ltimes \mathbb{Z}_2^2$

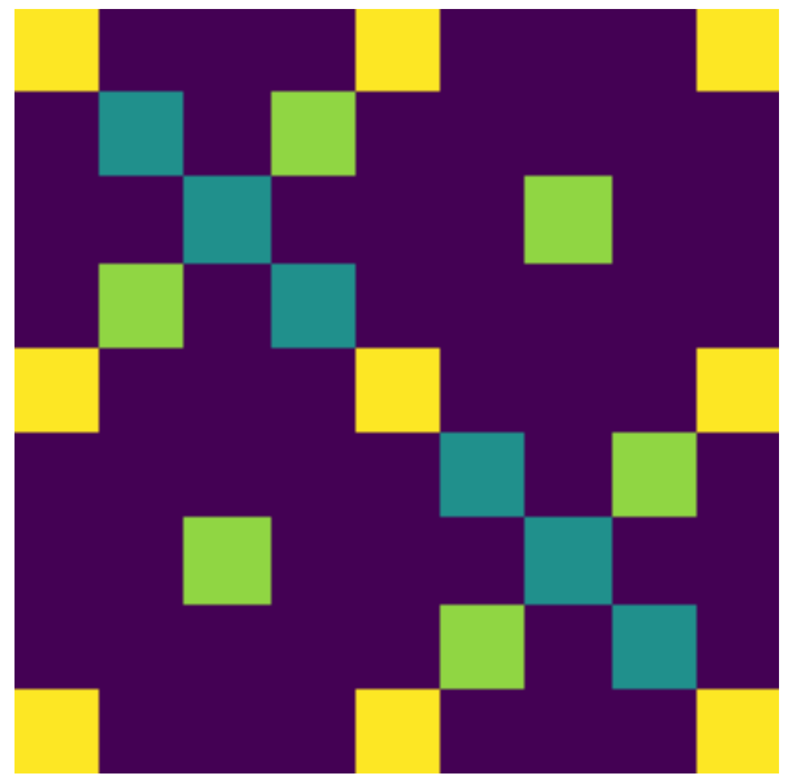
Equivariant basis for permutations, translation, 2d translation, and GCNN symmetries respectively, each of which are solutions to Q for different groups.

Methods

- Tensor Representations of Equivariant Basis


(a) $T_4^{S_6}$


(b) T_2^{Rubiks}


(c) $T_4^{\text{SO}(3)}$

Equivariant basis for various tensor representations T_k^G where G denotes the symmetry group. The r different solutions in the basis are shown in different colors. For $\text{SO}(3)$ the bases cannot be separated into disjoint sets of 0 or 1 valued vectors, so they choose overlapping colors randomly and add color for 0.

Methods

- **Efficiently Solving the Constraint**
- Two limitations:
  - The computational cost of computing the equivariant basis at initialization
  - The computational cost of applying the equivariant maps in the forward pass of a network
- The runtime for using SVD directly to compute the equivariant basis is too costly for all but very small representations $m = \dim(V) < 5000$. They improve upon the naive algorithm with two techniques:
  - Dividing the problem into a smaller set of independent subproblems: The feature space U in a neural network can be considered a combination of objects with different types and multiplicities. The features in standard CNN or deep set would be c copies of rank one tensors.
  - Exploiting structure in the constraint matrices to enable an efficient iterative Krylov subspace approach for computing the nullspace. Find the nullspace $Q \in \mathbb{R}^{n\times r}$ where r is the rank of the nullspace with the following optimization problem:

$$\min \quad \|CQ\|_F^2 \quad s.t \quad Q^\top Q = I.$$

Methods

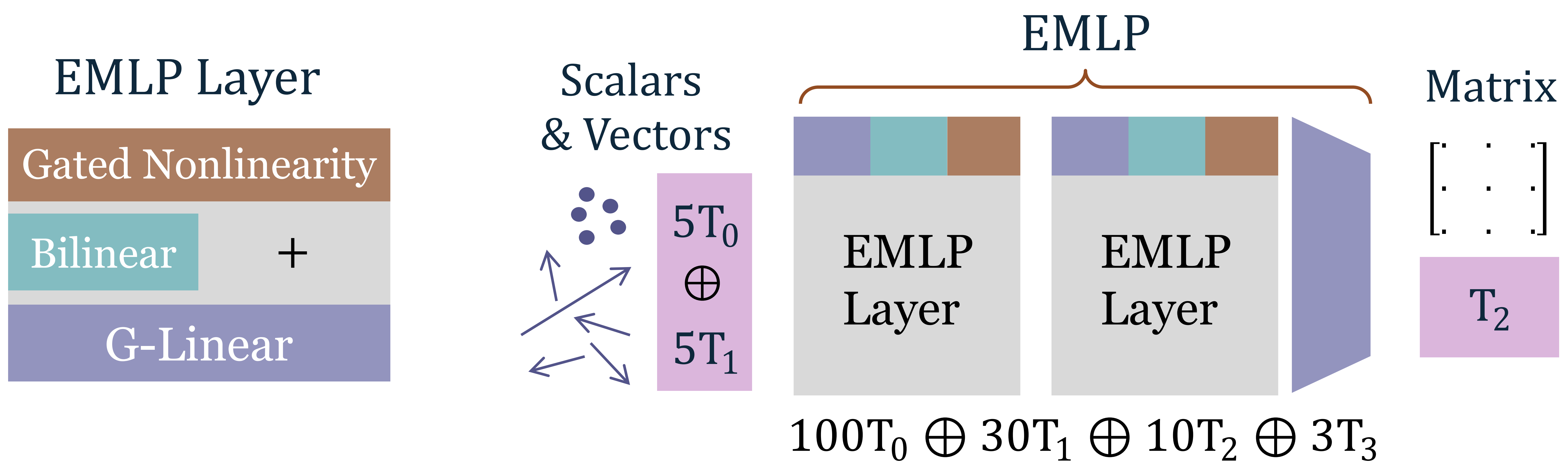
- **Network Architecture**
  - While the constraint-solving procedure can be applied to any linear representation, they use tensor representations to construct their network. The features in each layer are a collection of tensors of different ranks $v \in U = \bigoplus_{a \in \mathcal{A}} T_{(p_a, q_a)}$ with the individual objects $v_a \in T_{(p_a, q_a)}$. As a heuristic, they allocate the channels uniformly between tensor ranks for the intermediate layers. For example with 256 channels for an $\mathrm{SO}(3)$ equivariant layer with $\dim(T_{(p,q)}) = 3^{p+q}$, uniformly allocating channels produces $U = 70T_0 \oplus 23T_1 \oplus 7T_2 \oplus 2T_3$. The input and output layers are set by the types of the data. To build a full equivariant multilayer perceptron (EMLP) from the equivariant linear layer, we also need equivariant nonlinearities.
  - **Gated Nonlinearities**
    - For this purpose, they use gated nonlinearities introduced in Weiler et al. (2018). Gated nonlinearities act separately for each of the different objects in the features (that are concatenated through the direct sum $z = \mathrm{Concat}(\{v_a\}_{a \in \mathcal{A}})$). The nonlinearity takes values $\mathrm{Gated}(v_a) = v_a \sigma(s_a)$ where s_a is a scalar 'gate' for each of the objects.

Method

- **Network Architecture**
  - **Universality**
    - The theorem in Maron et al. (2019) shows that tensor networks with pointwise nonlinearities and G -equivariant linear layers for $G \leq S_n$ are universal. However, this result does not extend to the gated nonlinearities required for other groups and representations. The problem relates to not being able to express any kind of contractions between elements with the different objects within a feature layer (like a dot product).
  - **Cheap Bilinear Layers**
    - To address this limitation they introduce an inexpensive bilinear layer that performs tensor contractions on pairs of input objects that produce a given output type. Explicitly, two input objects $v_a \in T_{(a_1,a_2)}$ and $v_b \in T_{(b_1,b_2)}$ can be contracted to give a type $T_{(c_1,c_2)}$ if and only if $(a_1, a_2) = (c_1 + b_2, c_2 + b_1)$ or $(b_1, b_2) = (c_1 + a_2, c_2 + a_1)$. In other words, if v_a can be interpreted as a linear map from $T_b \to T_c$ then we can apply $y_c = \mathrm{Reshape}(v_a)v_b$ and vice versa.

Method

- Network Architecture



EMLP layers. G-equivariant linear layers, followed by the bilinear layer and a shortcut connection, and finally a gated nonlinearity. Stacking these layers together and choosing some internal representation, the EMLP maps some collection of geometric quantities to some other collection. Here they show the equivariant mappings from scalars and vectors to matrices.

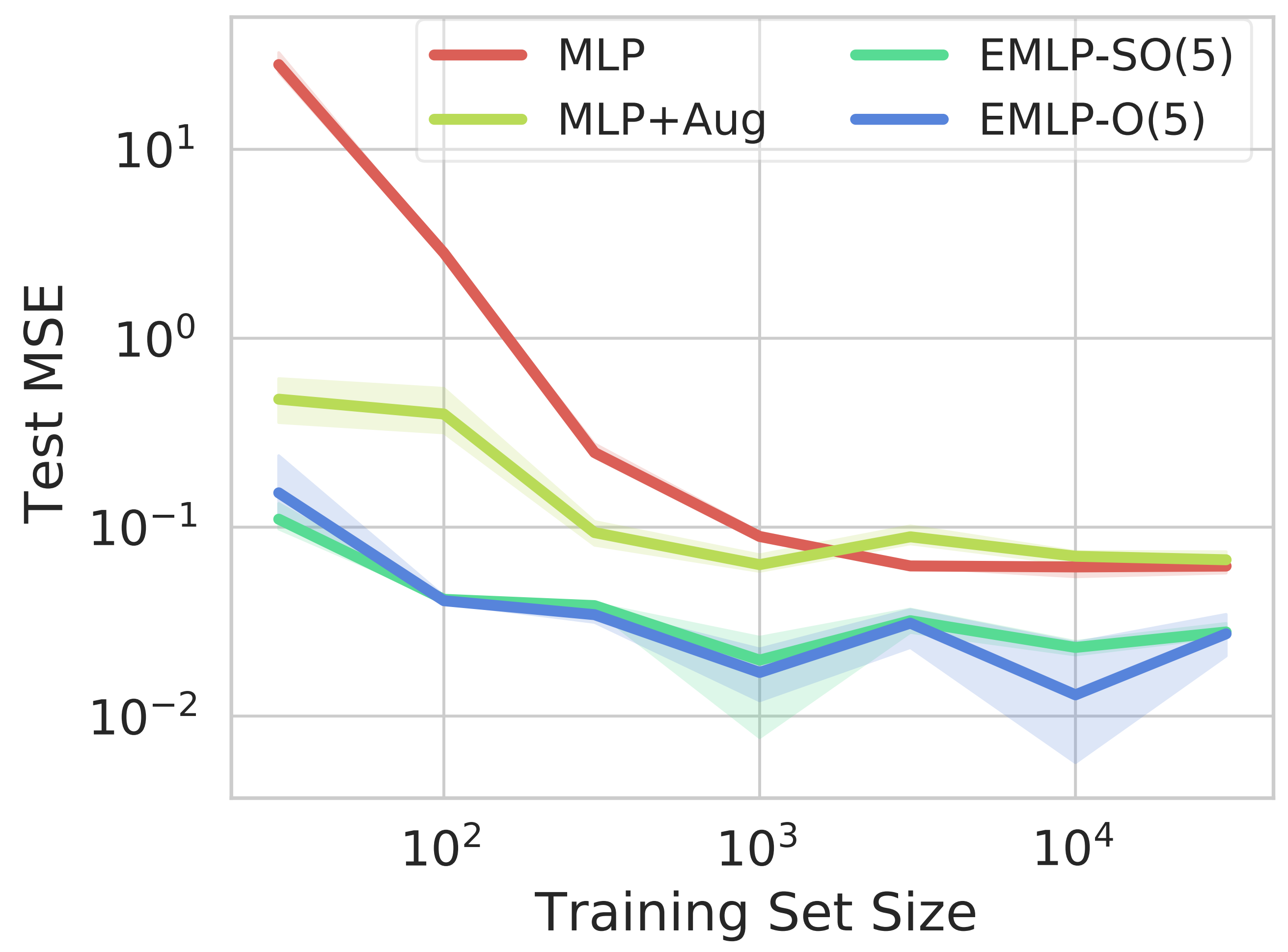
Results

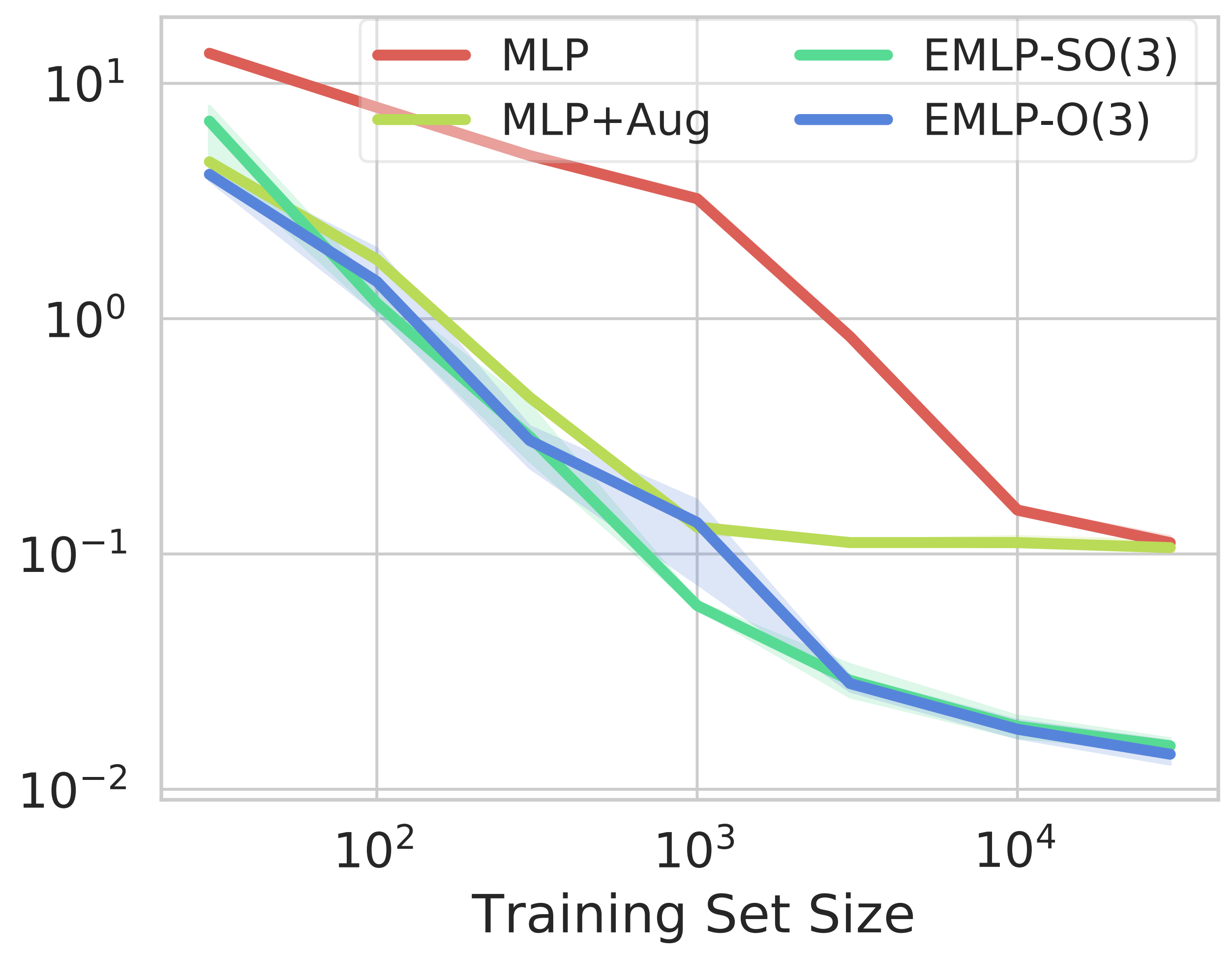
- **Synthetic Experiments**

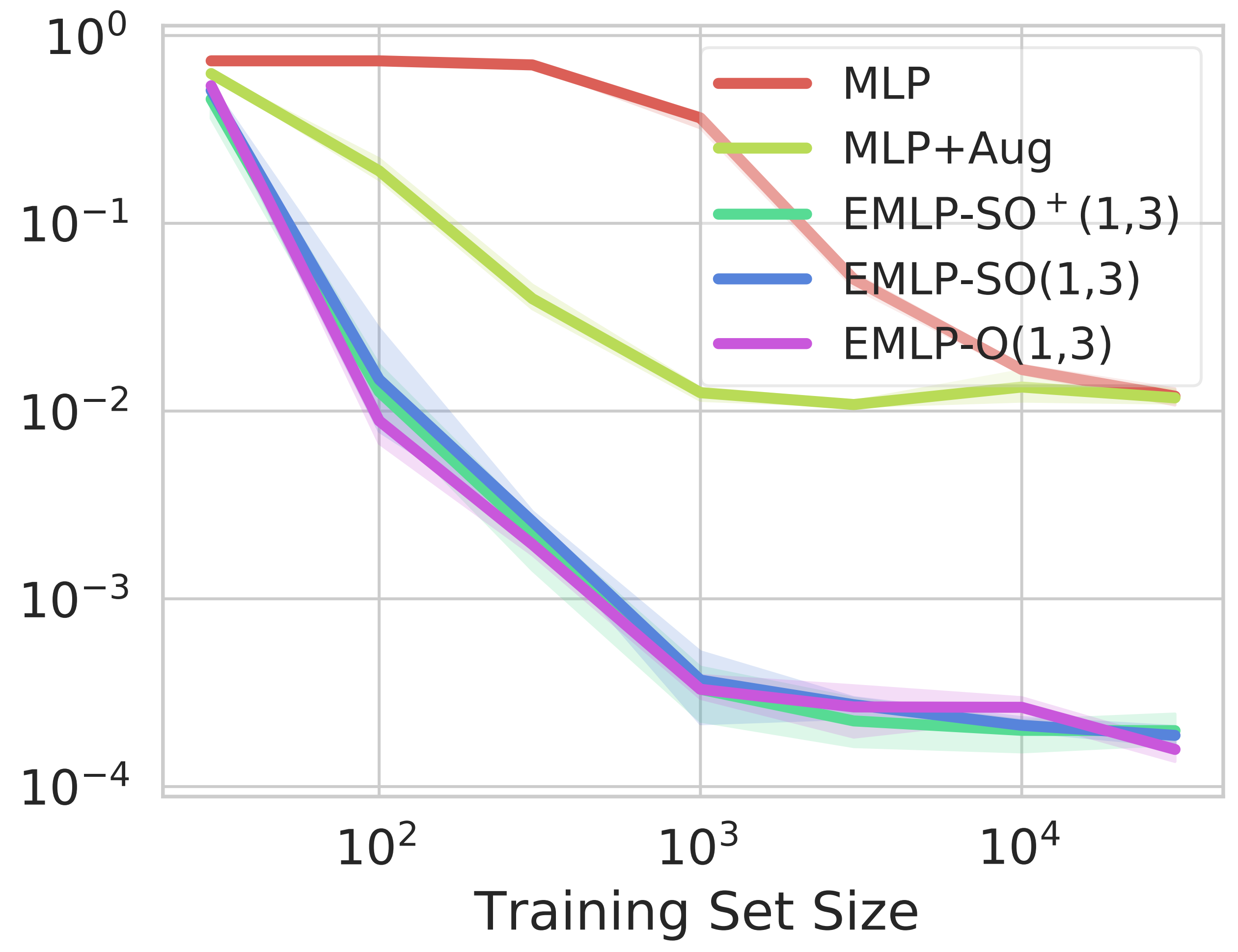
- Data: $2T_1 \to T_0$ | $5T_0 + 5T_1 \to T_2$ | $4T_1 \to T_0$

- Target: $f(x_1, x_2) = \sin(\|x_1\|) - \|x_2\|^3/2 + \frac{x_1^\top x_2}{\|x_1\|\|x_2\|}$ | $\mathcal{I} = \sum_i m_i(x_i^\top x_i I - x_i x_i^\top)$ | $[p^{(\mu}\tilde{p}^{\nu)} - (p^\alpha \tilde{p}_\alpha - p^\alpha p_\alpha)\eta^{\mu\nu}]$ $[q_{(\mu}\tilde{q}_{\nu)} - (q^\alpha \tilde{q}_\alpha - q^\alpha q_\alpha)\eta_{\mu\nu}]$

- Equivariance: $O(5)$ | $SO(3)$ | $SO(1,3)$



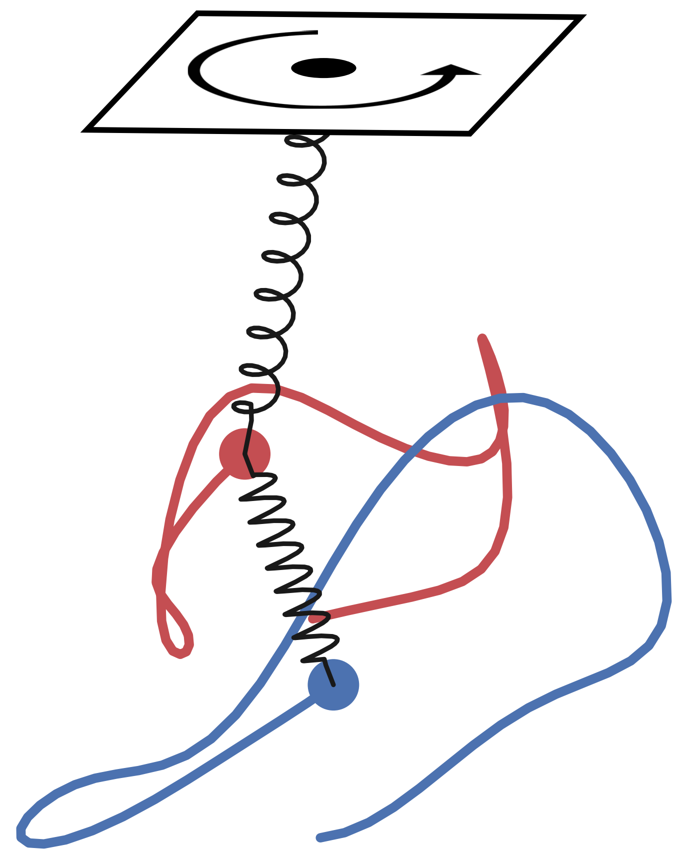


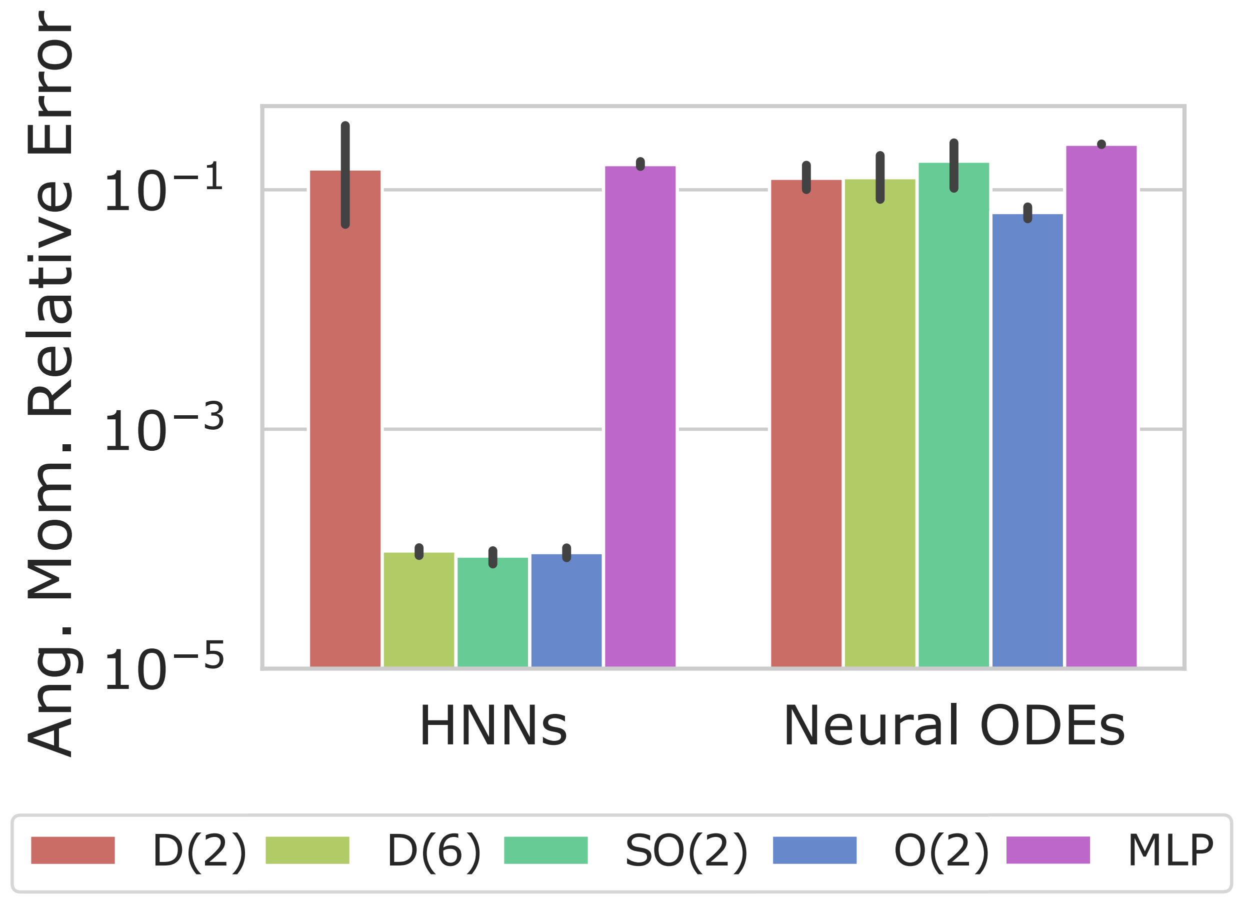


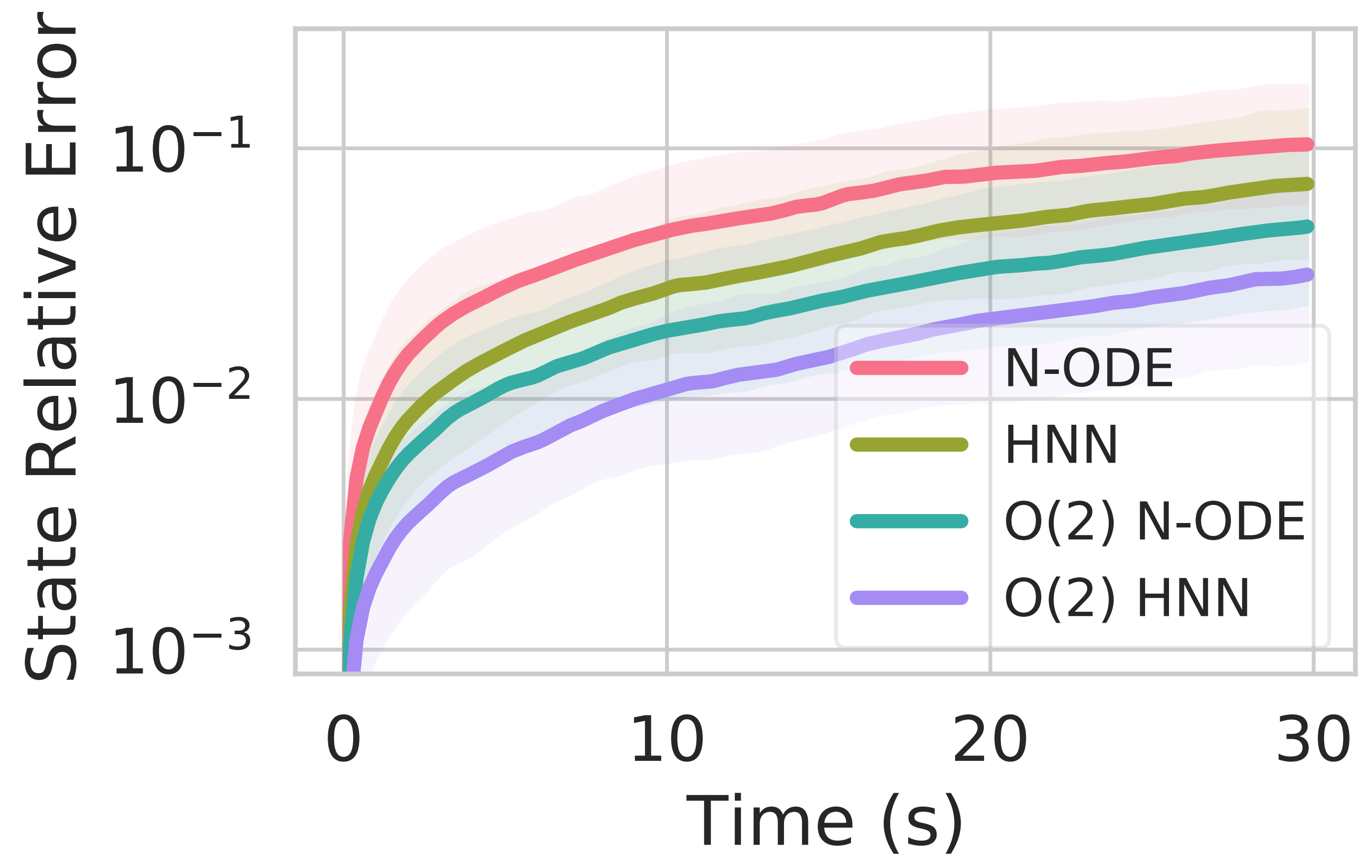
Results

- **Dynamical Systems**

- They apply their EMLP model to the task of learning the dynamics of a double pendulum connected by springs in 3D. The problem exhibits a $\mathrm{O}(2)$ rotational and reflectional symmetry about the z-axis as well as the Hamiltonian structure. As the state space cannot be traversed by the group elements alone, it is not a homogeneous space, a setting that has been explored very little in the equivariance literature.



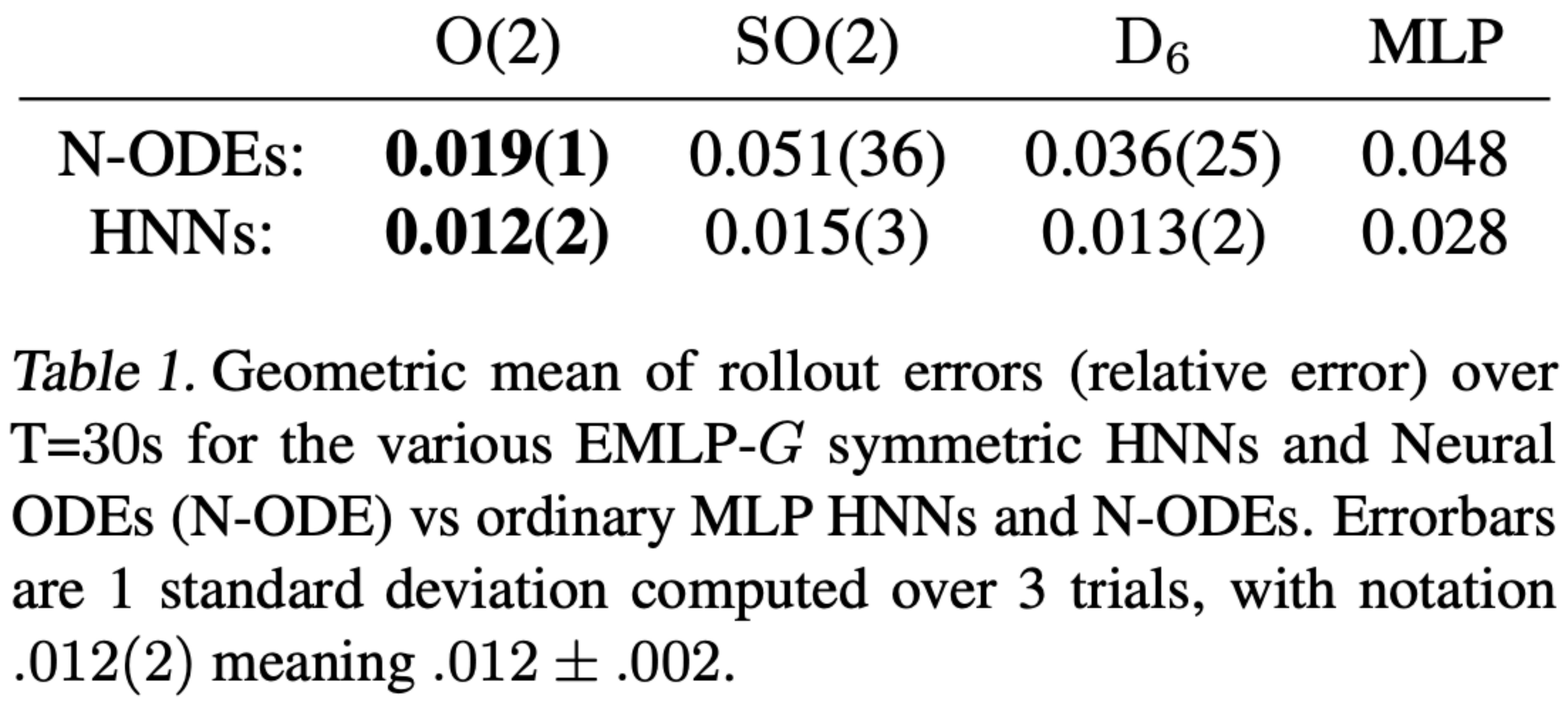




Left: A double spring pendulum (12s sample trajectory is shown). The system has an $\mathrm{O}(2)$ symmetry about the z-axis. **Middle:** Conservation of angular momentum about the z-axis (the geometric mean of the relative error is computed over 30s rollouts and averaged across initial conditions). **Right:** The relative error in the state as the trajectory is rolled out.

Results

- **Dynamical Systems**



	O(2)	SO(2)	D_6	MLP
N-ODEs:	**0.019(1)**	0.051(36)	0.036(25)	0.048
HNNs:	**0.012(2)**	0.015(3)	0.013(2)	0.028

Table 1. Geometric mean of rollout errors (relative error) over T=30s for the various EMLP-G symmetric HNNs and Neural ODEs (N-ODE) vs ordinary MLP HNNs and N-ODEs. Errorbars are 1 standard deviation computed over 3 trials, with notation .012(2) meaning $.012 \pm .002$.

Conclusion

- They present a construction for equivariant linear layers that is completely general to the choice of representation and matrix group.
- Convolutions, deep sets, equivariant graph networks, and GCNNs all fall out of the algorithm naturally as solutions for a given group and representation. Translating these capabilities into practice, they build EMLP and apply the model to problems with symmetry including Lorentz invariant particle scattering and dynamical systems, showing consistently improved generalization.
- Proposed an iterative approach to solve for the equivariant bases of very large representations.
- Though EMLP is not much slower than a standard MLP, dense matrix multiplies in an MLP and our EMLP make it slow to train models the size of convnets or large graph networks which have specialized implementations.

References

- Their ICML presentation can be found here.
- Their Github repo can be found here.