A. Overview

In Appendix B we prove that Equation 4 and Equation 5 are necessary and sufficient conditions to satisfy the symmetry constraint. In Appendix C we prove that the simple iterative MVM method for finding the nullspace converges to the true nullspace with an exponential rate, and we derive the complexity upper bounds for various symmetry groups and representations. In Appendix D we show that using gated nonlinearities or Norm-ReLUs alone are not sufficient for universality. In Appendix E we detail the various groups we implement and calculate the equivariant subspaces for different tensor representations. In Appendix F we detail the steps necessary to extend our implementation to new groups and representations. Appendix G details some additional rules by which the solutions for group products can be sped up. Finally, Appendix I and Appendix J describe training hyperparameters and how the datasets were constructed.

B. Necessary and Sufficient Conditions for Equivariance

Theorem 1 *Given a (real) Lie group G with a finite number of connected components, and a representation ρ acting on vector space V, the symmetry constraint*

$$\forall g \in G : \rho(g)v = v \tag{11}$$

for $v \in V$ is satisfied if and only if

$$\forall i = 1, .., D : \quad d\rho(A_i)v = 0, \tag{12}$$
$$\forall \ell = 1, ..., M : \quad (\rho(h_\ell) - I)v = 0, \tag{13}$$

where $\{A_i\}_{i=1}^D$ are D basis vectors for the D dimensional Lie Algebra $\mathfrak{g}$ with induced representation $d\rho$, and for some finite collection $\{h_\ell\}_{\ell=1}^M$ of discrete generators.

Proof: As shown in Appendix H, elements of a (real) Lie group can be written as $g = \exp\left(\sum_i \alpha_i A_i\right)\Pi_i h_{k_i}$ for some collection of real valued coefficients $\alpha_i \in \mathbb{R}$ and discrete coefficients $k_i \in [-M, ..., M]$ which index the M discrete generators (and their inverses). Note that M is upper bounded $M \leq (D+1) + \text{nc}(G)$, by the sum of the dimension and the number of connected components of G and is often much smaller as shown by the examples in Appendix E. For subgroups of S_n for example, $M \leq n$ (Guralnick, 1989). Discrete groups are included as a special case of Lie groups with $D = 0$. The forward and backward directions of the proof are shown below.

Necessary: Assume $\forall g \in G : \rho(g)v = v$. Writing $g = \exp\left(\sum_i \alpha_i A_i\right)\Pi_i h_{k_i}$, we can freely choose group elements with $k = \varnothing$ to find that

$$\forall \alpha : \rho\big(\exp\big(\sum_{i=1}^D \alpha_i A_i\big)\big)v = \exp\big(d\rho\big(\sum_{i=1}^D \alpha_i A_i\big)\big)v = v$$

using the correspondence between group and algebra representation (1). From the linearity of $d\rho(\cdot)$, this implies $\exp\big(\sum_i \alpha_i d\rho(A_i)\big)v = v$. Taking the derivative with respect to α_i at $\alpha = 0$, we have that

$$\forall i = 1, .., D : \quad d\rho(A_i)v = 0, \tag{14}$$

since the exponential map satisfies $\frac{d}{dt}\exp(tB)|_{t=0} = B$.

Similarly for the discrete constraints we can set $\alpha_i = 0$ and $k = [\ell]$ so that $\rho(g)v = \rho(\Pi_i h_{k_i})v = \rho(h_\ell)v = v$. By setting varying ℓ, we get the M constraints

$$\forall \ell = 1, ..., M : \quad (\rho(h_\ell) - I)v = 0. \tag{15}$$

Sufficient: Assume Equation 12 and Equation 13 both hold. Starting with the continuous constraints, exponential map (defined through the Taylor series) satisfies

$$\exp(B)v = v + Bv + \frac{1}{2}B^2 v + ...$$

but if $Bv = 0$ then $\exp(B)v = v$ since all terms except v are 0. Setting $B = \sum_i \alpha_i d\rho(A_i) = d\rho(\sum_i \alpha_i A_i)$ which satisfies $Bv = 0$, we have that

$$\begin{aligned} \forall \alpha_i : \quad & \exp\big(d\rho(\sum_i \alpha_i A_i)\big)v = v \\ \forall \alpha_i : \quad & \rho\big(\exp(\sum_i \alpha_i A_i)\big)v = v. \end{aligned} \tag{16}$$

Similarly for the discrete generators, if $\forall \ell : \rho(h_\ell)v = v$ then

$$\forall \ell : \quad v = \rho(h_\ell)^{-1}v = \rho(h_{-\ell})v,$$

and any product satisfies

$$\rho(\Pi_{i=1}^N h_{k_i})v = \Pi_{i=1}^N \rho(h_{k_i})v = \big(\Pi_{i=1}^{N-1}\rho(h_{k_i})\big)\rho(h_{k_N})v.$$

since $\rho(h_{k_N})v = v$ we can remove that factor and repeat the argument to get

$$\rho(\Pi_{i=1}^N h_{k_i})v = \big(\Pi_{i=1}^{N-1}\rho(h_{k_i})\big)v = ... = v. \tag{17}$$

Putting Equation 16 and Equation 17 together, we have that for all α, k: $\rho(\exp\left(\sum_i \alpha_i A_i\right)\Pi_i h_{k_i})v = v$. Since every group element can be expressed as $g = \exp\left(\sum_i \alpha_i A_i\right)\Pi_i h_{k_i}$ for some α, k, the equivariance constraint $\forall g \in G : \rho(g)v = v$ is satisfied.

C. Krylov Method for Efficiently Finding the Equivariant Subspace

The iterative Krylov subspace algorithm that we use to find the nullspace of the constraint matrix C is a close variant of the iterative methods for finding the largest eigenvectors such as power iteration and Ojas method . We need to

be able to compute the nullspaces of the massively large constraint matrices C (such as the $(4 \times 10^5) \times (7 \times 10^4)$ sized matrix for computing the equivariant subspace of $T_7^{S_5}$ in Appendix E), making use of efficient structure that allows fast MVMs with C.

While the shift and invert strategy for finding small eigenvalues is commonly recommended (Ipsen, 1997), the costs of inversion via conjugate gradients for these massive matrices can make it exceedingly slow in practice. Other methods such as block Lanczos (Eberly, 2004) and Wiedemann (Turner, 2006) have been explored in the literature for the nullspace problem, but these methods tend to be exceedingly complicated.

Instead we provide a much simpler algorithm that is also extremely fast: simple gradient descent followed by a small SVD. We prove that gradient descent converges exponentially in this problem, and that with probability 1 our method converges to the correct nullspace with error ϵ in $O(\log(1/\epsilon))$ iterations. We then verify this fact empirically. The algorithm is closely related to power iteration (Francis, 1961), Oja's rule (Shamir, 2015), and the PCA approaches that are framed as optimization problems (Garber & Hazan, 2015).

As introduced in algorithm 1 we propose the following algorithm for finding the nullspace with rank $r \leq r_{\max}$ and then use iterative doubling of $r_{\max}$ until the conditions are met.

Fast Krylov Nullspace

def **CappedKrylovNullspace**$(C, r_{\max})$:
$X \sim \mathcal{N}(0,1)^{n \times r_{\max}}$
while $L(X) > \epsilon$ **do**
  $L(X) = \|CX\|_F^2$
  $X \leftarrow X - \eta \nabla L$
end
$\tilde{Q}, \tilde{\Sigma}, \tilde{V} = \text{SVD}(X)$
return $\tilde{Q}$

Assume then that $r \leq r_{\max}$ (abbreviated r_m) and that we will use the notation from Equation 6 that $C = U \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P^\top \\ Q^\top \end{bmatrix}$. The gradients of L are $\nabla_X L = C^\top C X$ and thus gradient descent can be written as the iteration

$$X_{t+1} = (I - \eta C^\top C) X_t. \tag{18}$$

We can write X in terms of the true singular vectors of C (the eigenvectors of $C^\top C$) which form a basis. Q and P are orthogonal ($Q^\top Q = I$, $Q^\top P = 0$, $P^\top P = I$) and we define the projections of X_t onto these subspaces, $W_t = Q^T X_t$ and $V_t = P^T X_t$. As orthogonal transformations of isotropic Gaussians are also isotropic Gaussians, the two

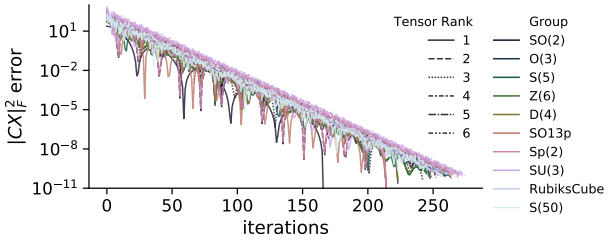

Figure 7. Exponential convergence of algorithm 1 shown empirically over a range of groups and tensor representations for $r_{\max} = 20$. In each of these cases, X converges to the limits of floating point precision in 300 iterations.

matrices are initially distributed $W_0 \sim \mathcal{N}(0,1)^{r \times r_m}$ and $V_0 \sim \mathcal{N}(0,1)^{(n-r) \times r_m}$.

Writing $X_t = QW_t + PV_t$, and noting $C^\top C = P\Sigma^2 P^\top$ we can now see the effect of the iteration on the subspaces:

$$\begin{aligned} QW_{t+1} + PV_{t+1} &= (I - \eta P\Sigma^2 P^\top)(QW_t + PV_t) \\ &= QW_t + P(I - \eta\Sigma^2)V_t. \end{aligned}$$

Unrolling the iteration, we have that $W_t = W_0$ and $V_t = P(I - \eta\Sigma^2)^t V_0$. So long as the learning rate is chosen $\eta < 2/\sigma_{\max}^2$ then the iteration will converge exponentially to $X = QW_0$. Given optimal learning rate, the convergence is $T = O(\kappa \log(1/\epsilon))$ where $\kappa = (\frac{\sigma_{\max}}{\sigma_{\min}})^2$. Since W_0 is a Gaussian random matrix $\mathcal{N}(0,1)^{r \times r_m}$, it will be full rank r with probability 1. Therefore, performing a final SVD on X will yield the nullspace Q. The runtime of this procedure is $O((M+D)\mathcal{T} r_m \log(\frac{1}{\epsilon}) + r_m^2 n)$ since each matrix multiply with C and $C^\top$ takes time $(M+D)\mathcal{T} r_m$ where $\mathcal{T}$ is the time for multiplies with the ρ and $d\rho$ matrices with a single vector, and there are $M + D$ such multiplies that go into a single multiply with C. Finally the $r_m^2 n$ factor is the cost of taking the SVD of X at the end. The exponential convergence is shown empirically across a range of groups in Figure 7.

If $r > r_{\max}$ then the SVD output $\tilde{Q}$ is a random projection of rank $r_{\max}$ of the true nullspace Q. Given an unknown r, we can simply rerun the algorithm doubling $r_{\max}$ each time until the rank of $\tilde{Q}$ is less than $r_{\max}$. Adding up the costs, the total runtime of the algorithm to reach an ϵ accurate solution for the nullspace is

$$O((M+D)\mathcal{T} r \log(\tfrac{1}{\epsilon}) + r^2 n). \tag{19}$$

To put this runtime into perspective, we can upper bound the runtime to compute the symmetric bases for rank p tensors T_k of any subgroup of the symmetric group $G \leq S_m$. Since all $G \leq S_m$ can be expressed with $D + M < m$ discrete generators (Guralnick, 1989), and axis-wise permutations of the $n = m^k$ sized tensors can be performed in time $\mathcal{T} =$

m^k, the runtime is upper bounded by $O(m^k r(m\log(\frac{1}{\epsilon}) + r))$.

Similarly for the orthogonal groups $\mathrm{SO}(m)$ and $\mathrm{O}(m)$ with $D = m(m-1)/2$ infinitesmal generators and $M \leq 1$ discrete generators, the MVM time can be done in $\mathcal{T} = km^k$ since the infinitesmal generators can be expressed with only 2 nonzero elements. Putting this together, the symmetric spaces for rank T_k tensors can be solved for in time $O(m^k r(km^2\log(\frac{1}{\epsilon}) + r))$, where r is also upper bounded by the Bell numbers $r \leq B_k$. Generalizations of the Lorentz group $\mathrm{SO}(p, n-p)$ and $\mathrm{O}(p, n-p)$ have identical runtimes, and similarly for the complex groups $\mathrm{SU}(n)$ and $\mathrm{U}(n)$, as well as the symplectic group $\mathrm{Sp}(n)$.

For the equivariant maps between the popular *irreducible* representations: $\rho = \psi_k \otimes \psi_\ell$ for the group $G = \mathrm{SO}(3)$, $\mathcal{T} = k^2\ell + k\ell^2$ giving the runtime of our method $O((k^2\ell + k\ell^2)r\log(\frac{1}{\epsilon}) + r^2k\ell)$. Meanwhile for irreducible representations of $G = \mathrm{SO}(2)$ the runtime is a mere $O(r\log(\frac{1}{\epsilon}) + r^2)$ regardless of k and ℓ.

D. Linear Layers and Gated Nonlinearities are Not Universal

Outside of the regular representations where each $\rho(g)$ is a permutation matrix, we cannot necessarily use pointwise nonlinearities. Existing equivariant nonlinearities for this setting such as Norm-ReLU and Gated-Nonlinearity can artificially limit the expressivity of the networks in cases such as when using tensor representations.

Theorem 2 *Consider equivariant networks built only from equivariant linear layers that map between (direct sums of) tensor representations with features $v \in \bigoplus_{a\in\mathcal{A}} T_a$ (as well as biases) and nonlinearities which act separately on each of the tensors $\sigma : T_a \to T_a$, or with an additional scalar as $\sigma : T_0 \times T_a \to T_a$. There exists groups (like $\mathrm{SO}(2)$ and $\mathrm{O}(3)$) for which these networks cannot approximate even simple equivariant functions.*

Suppose the base representation ρ of a group G includes elements that satisfy $\exists(g, g') : \rho(g') = -\rho(g)$, which we term the *parity property*. For simplicity assume the representation is orthogonal $\rho = \rho^*$ so that we can talk about rank k tensors rather than rank (p, q) tensors, but the same argument also applies for non-orthogonal representations setting $k = p + q$. Tensor representations $\rho_k(g) = \rho(g)^{\otimes k}$ that have an order k that is *odd* will also have the parity property since $\rho(g')^{\otimes k} = (-1)^k\rho(g)^{\otimes k}$. But because the equivariance constraint holds for all $g \in G$, the following two constraints must also hold for odd k:

$$\rho(g)^{\otimes k}v = v$$
$$-\rho(g)^{\otimes k}v = v.$$

Adding the two constraints together, we have that all equivariant tensors of odd rank (for groups with this property) are $v = 0$. This also means that all equivariant linear maps from a tensor with even rank to a tensor with odd rank will be 0.

This is a property of the equivariance for linear layers for certain groups and is not by itself a problem; however, if the equivariant nonlinearities σ act separately on each tensor and preserve its rank $\sigma : T_a \to T_a$ then all quantities in the network (inputs, outputs, and features) of even order are computationally disconnected from those of odd order. Since the nonlinearities act only on a given tensor and keep its order the same, and linear layers between even and odd are 0, there can be no nontrivial path between the two. Similarly for nonlinearities like Gated-Nonlinearities which take in an additional scalar gate as input, so that the nonlinearities are maps $\sigma : T_0 \times T_a \to T_a$ we can extend the result. For these kinds of nonlinearities, features of odd order can depend on inputs and features in previous layers of odd and even rank; however, there is still no path from a feature or input of odd order to a later feature or output of even order.

A simple example is the group $\mathrm{O}(3)$ and the standard vector representation for $R^\top R = I$, $\rho(R) = R$. Suppose the input to the network is the vector $v \in T_1$ and the target to be learned is the scalar $f(v) = \|v\| \in T_0$. Since the input is an odd order tensor and the output is an even order tensor, there can be no nonzero computational path in the network connecting them. Using Norm-ReLU or Gated-Nonlinearities the only valid output of such a network is a constant c which is independent of the input, and the network cannot fit a simple function such as $\|\cdot\|$.

Of course this limitation extends much beyond this simple example, preventing inner products, matrix vector multiplies, and many other kinds of valid equivariant functions from being expressed, regardless of the size of the network. In fact, using the standard vector representation for all groups $\mathrm{O}(n)$ as well as $\mathrm{SO}(2n)$ satisfies the parity property, and will provably have this limitation. Other groups like the Lorentz group $\mathrm{SO}(1, 3)$ and $\mathrm{O}(1, 3)$, and the symplectic group $\mathrm{Sp}(n)$ satisfy this property.

E. Equivariant Bases for Various Groups

In this section we list the dimension of the symmetric bases for various groups and tensor representations that we calculate using our algorithm, and visualize the bases.

E.1. Discrete Translation Group $\mathbb{Z}_n$

The discrete translation group (or cyclic group) $\mathbb{Z}_n$ is generated by a single shift permutation $\mathrm{P}[n, 1, 2, ..., n-1]$. Translation equivariant neural networks make use of convolutions which are maps $(T_1 \to T_1) \cong T_2$ and average

pooling ($T_1 \to T_0$) $\cong T_1$. We recover the n dimensional convolution bases and the 1 dimensional average pooling element as shown in the table and Figure 3. We also show the ranks for higher order tensors, which appear to satisfy $r = n^{k-1}$ for $\mathbb{Z}_n$ and T_k. While we are not aware of the higher order equivariant tensor having been derived in the literature, it's not unlikely due to the prominence of the translation group in signal processing.

	$\mathbb{Z}_2$	$\mathbb{Z}_3$	$\mathbb{Z}_4$	$\mathbb{Z}_5$	$\mathbb{Z}_6$	$\mathbb{Z}_7$	$\mathbb{Z}_8$
T_1	1	1	1	1	1	1	1
T_2	2	3	4	5	6	7	8
T_3	4	9	16	25	36	49	64
T_4	8	27	64	125	216	343	512
T_5	16	81	256	625	1296	2401	
T_6	32	243	1024	3125			
T_7	64	729	4096				

Table 2. Symmetric subspace rank r for tensors T_k of $G = \mathbb{Z}_n$

E.2. Permutation Group $\mathbb{S}_n$

We review the solutions to the permutation group $\mathbb{S}_n$ which were solved for analytically in Maron et al. (2018), which we solve for numerically using our algorithm. As expected, the solutions bases match the limiting size of the kth Bell number B_k as $n \to \infty$.

However, Maron et al. (2018) claim that the size of the basis is always B_k regardless of n and that is not quite correct. The basis derived in Maron et al. (2018) is always equivariant, but sometimes it contains linearly dependent solutions, leading to an overcounting when n is small. The fact that the basis cannot always be of size B_k can be seen from the fact that the total dimension of T_k is n^k and the equivariant subspace thus has rank $r \leq n^k$. The Bell numbers grow super exponentially in k (about $(k/\log(k))^k$) and therefore given any n they must exceed the maximum n^k for some value of k. The place where the original argument of Maron et al. (2018) breaks down is when the equivalence classes γ may be empty. For example the equivalence class $\gamma_1 = \{\{1\}, \{2\}, \{3\}, \{4\}\}$ corresponds to indices i_1,i_2,i_3,i_4 which are all distinct. But for $n < 4$ one cannot form a set of four indices that are all distinct, hence γ_1 is empty for $n < 4$. To the best of our knowledge, the dimension of the equivariant subspaces for small n that we report below are the precise values and have not been presented anywhere else.

	$\mathbb{S}_2$	$\mathbb{S}_3$	$\mathbb{S}_4$	$\mathbb{S}_5$	$\mathbb{S}_6$	$\mathbb{S}_7$	$\mathbb{S}_8$	B_k
T_1	1	1	1	1	1	1	1	1
T_2	2	2	2	2	2	2	2	2
T_3	4	5	5	5	5	5	5	5
T_4	8	14	15	15	15	15	15	15
T_5	16	41	51	52	52	52	52	52
T_6	32	122	187	202	203	203		203
T_7	64	365	715	855				877

Table 3. Symmetric Subspace rank r for tensors T_k of $G = \mathbb{S}_n$

E.3. Rubik's Cube Group

Showing the capabilities to apply to unexplored groups and representations, we compute the equivariant bases for linear layers that are equivariant to the action of the Rubik's Cube group. The Rubik's cube group is a subgroup of the permutation group $G < S_{48}$ containing all valid Rubik's cube transformations. The group is extremely large $|G| > 4 \times 10^{19}$, but is generated by only 6 generators: F, B, U, D, L, R a quarter turn about the front, back, up, down, left, and right faces.

We use the standard 48 dimensional regular representation where each of the $6 * (9 - 1) = 48$ facets (the center facets are excluded) is a component. The state of the Rubik's cube is represented by a 48 dimensional vector where each component is an integer $0 - 5$ representing the 6 possible colors each facet can take. Using this representation, the 6 generators can be expressed as permutations and we refer the readers to the code for the (lengthy) values of the permutations. Below we show the dimension of the equivariant basis and the size of the tensors in which the basis is embedded. Note that as the Rubik's cube is a subgroup of S_{48}, and has fewer group elements as symmetries, the size of the equivariant basis is larger $2, 6, 22, ...$ vs $1, 2, 5, ...$ For T_4 of size $48^4 = 5308416$ we were able to run the solver with $r_{\max} = 20$ before running out of GPU memory.

	T_1	T_2	T_3	T_4
r	2	6	22	>20
48^k	48	2304	110592	5308416

Table 4. Symmetric Subspace rank r for tensors T_k of Rubik's Cube Group

E.4. Continuous Rotation Groups $\mathrm{SO}(n)$ and $\mathrm{O}(n)$

The special orthogonal group $\mathrm{SO}(n)$ and the orthogonal group $\mathrm{O}(n)$ are continuous Lie groups have the Lie algebra

$$\mathfrak{o}(n) = \mathfrak{so}(n) = T_{\mathrm{id}}\mathrm{SO}(n) = \{A \in \mathbb{R}^{n \times n} : A^\top = -A\}$$

of dimension $D = n(n-1)/2$. The orthogonal group can be constructed with the additional discrete generator

$h = \begin{bmatrix} -1 & 0 \\ 0 & I_{n-1} \end{bmatrix}$ that has $\det(h) = -1$.

	SO(2)	SO(3)	SO(4)	SO(5)	SO(6)	SO(7)
T_2	2	1	1	1	1	1
T_3	0	1	0	0	0	0
T_4	6	3	4	3	3	3
T_5	0	6	0	1	0	0
T_6	20	15	25	15	16	15
T_7	0	36	0	15		
T_8	70	91	196			

Table 5. Symmetric subspace rank r for tensors T_k of $G = \mathrm{SO}(n)$

We omit the first row T_1 since all the values are 0. The additional basis element along the diagonal $n = k$ can be recognized as the well known anti-symmetric Levi-Civita symbol $\varepsilon_{ijk\ell\ldots}$. However this basis element does not respect orientation reversing isometries, and is thus absent in the equivariant basis for $\mathrm{O}(n)$:

	O(2)	O(3)	O(4)	O(5)	O(6)	O(7)
T_2	1	1	1	1	1	1
T_3	0	0	0	0	0	0
T_4	3	3	3	3	3	3
T_5	0	0	0	0	0	0
T_6	10	15	15	15	15	15
T_7	0	0	0	0		
T_8	35	91	105			

Table 6. Symmetric subspace rank r for tensors T_k of $G = \mathrm{O}(n)$

E.5. Lorentz Groups $\mathrm{SO}^+(1,3)$, $\mathrm{SO}(1,3)$, $\mathrm{O}(1,3)$

The Lorentz group is defined as the set of matrices that preserve the Lorentz metric η: $\mathrm{O}(1,3) = \{L \in \mathbb{R}^{4\times4} : L^\top \eta L = \eta\}$. Differentiating, one gets the $D = 6$ dimensional Lie algebra $\mathfrak{so}(1,3) = \{A \in \mathbb{R}^{4\times4} : A^\top\eta + \eta A = 0\}$. The full Lorentz group $\mathrm{O}(1,3)$ has four connected components.

The identity component of the Lorentz group $\mathrm{SO}^+(1,3)$ is just the exponential of the Lie algebra $\mathrm{SO}^+(1,3) = \exp(\mathfrak{o}(1,3))$. The subgroup $\mathrm{SO}(1,3)$ of $\mathrm{O}(1,3)$ with determinant 1 can be constructed with the additional generator $h_1 = -I$ (which combines time reversal with a parity transformation), and the full Lorentz group $\mathrm{O}(1,3)$ includes h_1 as well as the generator $h_2 = \begin{bmatrix} -1 & 0 \\ 0 & I_3 \end{bmatrix}$ that reverses time only. As these groups are not orthogonal, we must distinguish $T_{(a,b)}$ from $T_{(a+b,0)}$. Below we show the number of basis vectors for $T_{(k,0)}$ which for these 3 groups is the same number as for $T_{(k-i,i)}$ although the bases elements are distinct.

	$T_{(2,0)}$	$T_{(3,0)}$	$T_{(4,0)}$	$T_{(5,0)}$	$T_{(6,0)}$	$T_{(7,0)}$	$T_{(8,0)}$
$\mathrm{SO}^+(1,3)$	1	0	4	0	25	0	196
$\mathrm{SO}(1,3)$	1	0	4	0	25	0	196
$\mathrm{O}(1,3)$	1	0	3	0	15	0	105

Table 7. Symmetric subspace rank r for tensors $T_{(k,0)}$ for the Lorentz groups.

E.6. Symplectic Group $\mathrm{Sp}(n)$

Similar to the orthogonal group and the Lorentz group, the symplectic group is defined through the perservation of a quadratic form.

$$\mathrm{Sp}(n) = \{M \in \mathbb{R}^{2n\times2n} : M^\top \Omega M = \Omega\},$$

where $\Omega = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$ and is often relevant in the context of Hamiltonian mechanics and classical physics. The quadratic form Ω can be interpreted as a measurement of oriented area (in phase space) and is preserved by the evolution of many systems. The $D = n(2n+1)$ dimensional Lie algebra satisfies

$$\mathfrak{sp}(n) = \{A \in \mathbb{R}^{2n\times2n} : A^\top\Omega + \Omega A = 0\},$$

and any element in $\mathrm{Sp}(n)$ can be written $M = \exp(A_1)\exp(A_2)$ for some $A_1, A_2 \in \mathfrak{sp}(n)$.

	Sp(1)	Sp(2)	Sp(3)	Sp(4)	Sp(5)	Sp(6)
$T_{(2,0)}$	1	1	1	1	1	1
$T_{(3,0)}$	0	0	0	0	0	0
$T_{(4,0)}$	2	3	3	3	3	3
$T_{(5,0)}$	0	0	0	0	0	0
$T_{(6,0)}$	5	14	15	15		
$T_{(7,0)}$	0	0	0			
$T_{(8,0)}$	14	84				

Table 8. Symmetric subspace rank r for tensors $T_{(k,0)}$ of $G = \mathrm{Sp}(n)$

Although for large values of n, the dimension of the basis for $\mathrm{Sp}(n)$ becomes similar to that of its subgroup $\mathrm{O}(n)$, the basis elements themselves are quite different. Like the Lorentz groups, different ways of distributing the rank between the base vector space and its dual as $T_{(k-i,i)}$ for different i yields different solutions for the equivariant basis.

E.7. Special Unitary Group $\mathrm{SU}(n)$

Using a complex valued SVD and replacing the objective in the iterative algorithm $L(Q) = \|CQ\|_F^2 = \mathrm{Tr}(Q^\top C^\top CQ)$

with $L(Q) = \mathrm{Tr}(Q^\dagger C^\dagger C Q)$ where $\dagger$ is the complex conjugate transpose, we can apply our method to solve for the equivariant bases for complex groups such as the special unitary group $\mathrm{SU}(n)$ relevant for the symmetries of the standard model of particle physics.

The group can be defined

$$\mathrm{SU}(n) = \{U \in \mathbb{C}^{n\times n} : U^\dagger U = 1, \;\; \det(U) = 1\}.$$

The Lie algebra of dimension $D = n^2 - 1$ satisfies

$$\mathfrak{su}(n) = \{A \in \mathbb{C}^{n\times n} : A^\dagger = -A, \;\; \mathrm{Tr}(A) = 0\}.$$

The group is contained in the image of the exponential map, $\exp(\mathfrak{su}(n)) = \mathrm{SU}(n)$. Since the size of the basis differs between $T_{(4,1)}$ and $T_{(3,2)}$ for example, we show the solutions for only a selection of tensor ranks. It may also be useful to consider *anti*-linear maps, but we leave this to future work.

	$T_{(3,0)}$	$T_{(3,1)}$	$T_{(3,2)}$	$T_{(3,3)}$	$T_{(4,0)}$	$T_{(4,1)}$	$T_{(4,2)}$
SU(2)	0	2	0	5	2	0	5
SU(3)	1	0	0	6	0	3	0
SU(4)	0	0	0	6	1	0	0

Table 9. Symmetric subspace rank r for tensors $T_{(q,p)}$ of $G = \mathrm{SU}(n)$

F. Recipe for Use

Below we outline the minimum required steps for adding new groups and representations to our existing implementation written in Jax (Bradbury et al., 2018).

Adding new groups:

1. Specify a sufficient set of M discrete generators and their base representation as a matrix $\rho(h_i)$ or as a matrix vector multiply $v \to \rho(h_i)v$ for each $i = 1, 2, ..., M$
2. Specify a basis for the Lie algebra (if any) and its base representation as a matrix $d\rho(A_i)$ or as a matrix vector multiply $v \to d\rho(A_i)v$ for each $i = 1, 2, ..., D$

We walk through these steps and provide examples in **Adding new representations $\tilde{\rho}$ to existing groups**:

1. Specify $\tilde{\rho}(h_i)$ as a function of $\rho(h_i)$
2. Define $\dim(V)$, $==$, and hash functions for the representation

We provide more detailed instructions along with examples for implementing new groups at https://emlp.readthedocs.io/en/latest/notebooks/3new_groups.html and new representations at https://emlp.readthedocs.io/en/latest/notebooks/4new_representations.html.

Given that a base representation ρ is *faithful*, all representations can be constructed as functions of this ρ whatever it may be.

A specification of $\tilde{\rho}(h) = f(\rho(h))$ induces the Lie algebra representation $d\tilde{\rho}(A)$ which may be computed automatically using autograd Jacobian vector products (JVP), without needing to specify it manually. $d\tilde{\rho}(A) = \mathrm{JVP}(f, I, d\rho(A))$ which corresponds in math terms to the operation $d\tilde{\rho}(A) = Df|_{\rho(h)=I}(d\rho(A))$.

G. Group Products

Many of the relevant groups for larger problems like 2D arrays, GCNNs, point clouds (Fuchs et al., 2020), sets of images (Maron et al., 2020), and hierarchical structures (Wang et al., 2020) have multiple distinct group substructures. 2D translation symmetry is the group $G = \mathbb{Z}_n \times \mathbb{Z}_n = \mathbb{Z}_n^2$, GCNNs have $G = H \ltimes \mathbb{Z}_n^2$ point clouds typically have $G = S_n \times \mathrm{E}(3)$, sets of images have the symmetry $S_m \times \mathbb{Z}_n^2$, and a voxelized point cloud network could be $(S_n \times \mathrm{E}(3)) \wr \mathbb{Z}_m^3$. Here these symbols for combining groups are the direct product ($\times$), semi-direct product ($\ltimes$) and wreath product ($\wr$). An additional asymptotic speedup can be achieved for groups that are constructed using these structures by exploiting knowledge about how solutions for the larger group depend on the solutions for the constituent groups.

Suppose representation ρ_a of the group G_a acts on the space V_a which has the symmetric basis Q_a, and ρ_b of G_b acts on V_b with the symmetric basis Q_b.

($\times$): As shown in (Maron et al., 2020), the equivariant basis for $G_a \times G_b$ with rep $\rho_a \otimes \rho_b$ can be written as the Kronecker product $Q_{ab} = Q_a \otimes Q_b$.

($\wr$): In (Wang et al., 2020) it was worked out that the equivariant basis for $G_a \wr G_b$ with rep $(\rho_a \wr \rho_b) \otimes (\rho_a \wr \rho_b)^*$ satisfies $\mathrm{unvec}(Q_{ab}[\alpha, \beta]) = \mathrm{unvec}(Q_a \beta) \otimes I + \mathbb{1}\mathbb{1}^\top \otimes \mathrm{unvec}(Q_b \alpha)$ where α, β are coefficient vectors of size r_a,r_b and the $\mathrm{unvec}(\cdot)$ operation reshapes a vector into a matrix.

H. Parametrizing Lie Groups with Continuous and Discrete Generators

According to Winkelmann (2003), every real connected Lie group G of dimension D contains a dense subgroup $H \leq G$ generated by $D + 1$ elements (which can be constructed explicitly by sampling elements in a neighborhood of the identity). Since H is dense in G, for every element

$g \in G$ has a neighborhood $\mathcal{N}(g)$ which contains an element $h \in H$. Since $h \in \mathcal{N}(g)$ it must also be true that $gh^{-1} \in \mathcal{N}(\mathrm{id})$ is in a neighborhood of the identity. Since the exponential map exp is a bijection for a neighborhood around the identity, it must be possible to express $gh^{-1} = \exp(A) = \exp(\sum_i \alpha_i A_i)$ for some $A \in \mathfrak{g}$ that we can then write in terms of the basis elements A_i. Rearranging, and expressing h in terms of the finite generators for H, $h = \Pi_i h_{k_i}$, we have that any $g \in G$ can be written

$$g = \exp(\sum_i \alpha_i A_i)\Pi_i h_{k_i}. \tag{20}$$

or more succinctly $G = \exp(\mathfrak{g})H$. It is likely that the result can also be extended to other fields like $\mathbb{C}$ such as by embedding the group in a higher dimensional real group, but we focus on the real case here. For groups with multiple connected components, we can apply the same result with at most one additional discrete generator for each of the connected components.

I. Implementation Details

While there is substantial freedom in the chosen representation for a given feature layer of a neural network, for maximum expressiveness in the subsequent neural network architecture given a fixed channel budget, we suggest allocating the channels to the different representations in each layer with a uniform allocation heuristic. Progressing from lower dimensional representations to higher dimensional ones, the multiplicity of the representation should be chosen so that the number of channels for associated with each is approximately the same. We use this heuristic for the equivariant networks of each experiment in the paper.

For the three synthetic experiments, we use networks constructed with 3 EMLP layers each with $c = 384$ channels, followed by a single equivariant linear layer mapping to the output type. The baseline MLPs also have 3 hidden layers of size $c = 384$ each. We train all models with batchsize 500, and learning rate 3×10^{-3} with the Adam optimizer (Kingma & Ba, 2014). We train for a total of $\min(900000/N, 1000)$ epochs where N is the size of the dataset, which we found was ample for convergence of both models in all cases. The training time for the EMLP is a couple minutes, while the MLP model trains in < 1 minute.

For the modeling of the double spring pendulum dynamical system we use the same hyperparameters except with $c = 128$ for all models. For the numerical integrator, we use the adaptive RK integrator that is default to Jax with tolerance 2×10^{-6}. We measure relative error as $\mathrm{relative\_error}(a, b) = \|a - b\|/(\|a\| + \|b\|)$. For calculating state relative error, the state is the full vector z of both position and momentum. We train for a total of 2000 epochs, long enough for each of the models to converge.

J. Datasets

We generate the $\mathrm{O}(5)$ invariant dataset using the function $f(x_1, x_2) = \sin(\|x_1\|) - \|x_2\|^3/2 + \frac{x_1^\top x_2}{\|x_1\|\|x_2\|}$ and sampling $x_1, x_2 \sim \mathcal{N}(0, 1)^5$. We generate the $\mathrm{O}(3)$ equivariant inertia dataset by computing $\mathcal{I} = \sum_{i=1}^5 m_i(x_i^\top x_i I - x_i x_i^\top)$ for $x_i \sim \mathcal{N}(0, 1)^3$ and sampling positive masses by passing random entries through a softplus: $m_i \sim \mathrm{Softplus}(\mathcal{N}(0, 1))$.

For the Lorentz invariant particle interaction dataset we calculate the targets $y = 4[p^{(\mu}\tilde{p}^{\nu)} - (p^\alpha \tilde{p}_\alpha - p^\alpha p_\alpha)\eta^{\mu\nu}][q_{(\mu}\tilde{q}_{\nu)} - (q^\alpha \tilde{q}_\alpha - q^\alpha q_\alpha)\eta_{\mu\nu}]$ from the sampled momenta $p_\mu, \tilde{p}_\mu, q_\mu, \tilde{q}_\mu \sim \mathcal{N}(0, 1/4^2)$.

For each of these datasets we separate out a test set of size 5000 and a validation set of size 1000 which we use for early stopping.

For the double spring dynamical system, we generate the ground truth trajectories using the Hamiltonian dynamics of the Hamiltonian

$$H(x_1, x_2, p_1, p_2) = V(x_1, x_2) + T(p_1, p_2)$$

where $T(p_1, p_2) = \|p_1\|^2/2m_1 + \|p_2\|^2/2m_2$ and $V(x_1, x_2) =$

$$\tfrac{1}{2}k_1(\|x_1\| - \ell_1)^2 + \tfrac{1}{2}k_2(\|x_1 - x_2\| - \ell_2)^2 + m_1 g^\top x_1 + m_2 g^\top x_2.$$

The constants are chosen $m_1 = m_2 = k_1 = k_2 = \ell_1 = \ell_2 = 1$. The gravity direction is down $g = [0, 0, 1]$. We sample the initial conditions from the distribution $x_1 \sim [0, 0, -1.5] + \mathcal{N}(0, .2^2)^3$, $x_2 \sim [0, 0, -3] + \mathcal{N}(0, .2^2)^3$ and $p_1, p_2 \sim \mathcal{N}(0, .4^2)^3$. We integrate these systems for a time $T = 30s$ and for each initial condition we select a randomly chosen $1s$ chunk (evaluated at five $0.2s$ intervals) as the training data. We generate 1500 trajectory chunks which we split up into 500 for each of the train, validation, and test sets.