LieTransformer: Equivariant Self-Attention for Lie Groups

Michael Hutchinson*†, Charline Le Lan*†, Sheheryar Zaidi*†, Emilien Dupont†, Yee Whye Teh†‡, Hyunjik Kim‡

† Department of Statistics, University of Oxford, ‡ DeepMind, London

*Equal contribution

Introduction

We propose a self-attention-based architecture that is equivariant to arbitrary Lie groups and their discrete sub-groups.

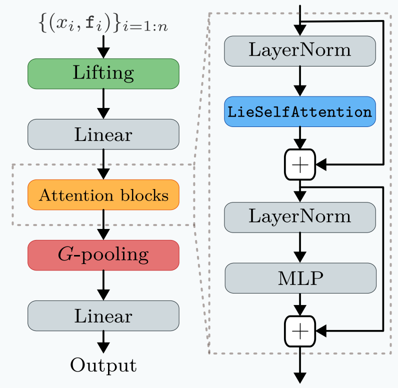
The setting we consider has the locations of points in a **homogenous space**, $\mathcal{X}$, of some **Lie group**, G, such that $\mathcal{X} = G/H$. The input to the model is a set of location-value pairs, $\{(x_i, \mathtt{f}_i)\}_{i=1}^n$, $x_i \in \mathcal{X}$, $\mathtt{f}_i \in \mathbb{R}$, and the output a vector, $y \in \mathbb{R}^n$.

Denoting the network function $y = \Phi(\{(x_i, \mathtt{f}_i)\}_{i=1}^n)$, the objective is to create a network that is invariant to the action of the Lie group, i.e.

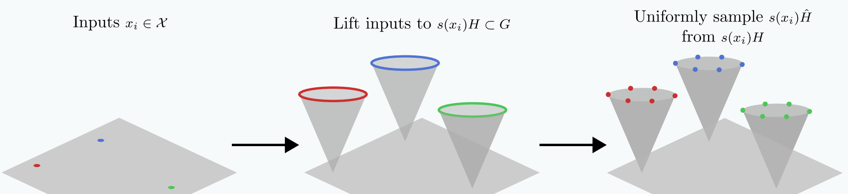
$$\Phi(\{(x_i, \mathtt{f}_i)\}_{i=1}^n)) = \Phi(\{(g \cdot x_i, \mathtt{f}_i)\}_{i=1}^n)).$$

Main Architecture

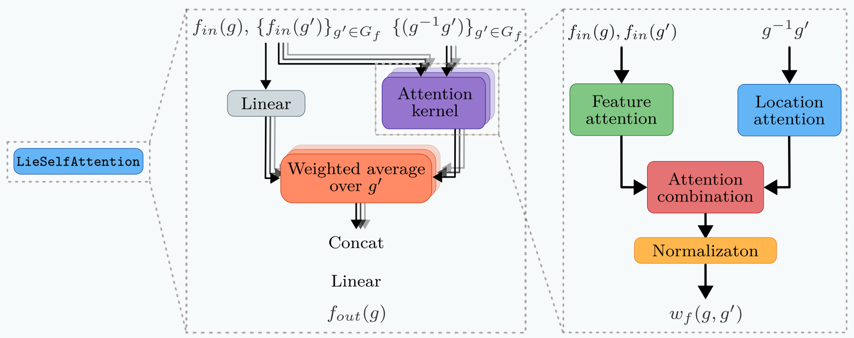
The architecture we propose closely follows that of the typical transformer (Vaswani et al., 2017), with a number of modifications to ensure equivariance of the model.



The first key component is the **lifting layer**. This maps each input pair $(x_i, \mathtt{f}_i)$ to the set of points $\{(g, \mathtt{f}_i) : g \in s(x_i)H\}$. The advantage of performing this lift is that it is now easier to define equivariant functions on these sets of points.



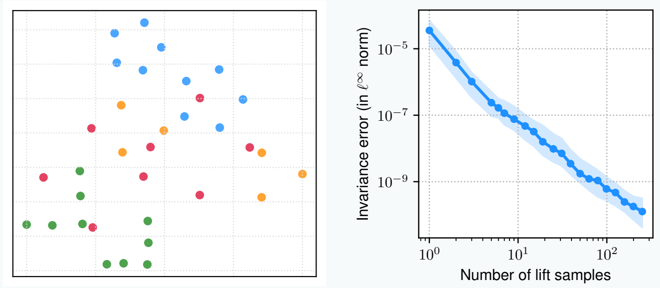
The modified **LieSelfAttention** layer incorperates attention based both on the features *and* the location in G of each point. The locations attention is parametrised by $g_i^{-1} g_j$, ensuring equivariance.



Shape Counting

On an SE(2)-invariant shape counting task for point clouds, LieTransformer outperforms the non-invariant SetTransformer. SetTransformer does not close the gap in performance **even when trained with SE(2) data augmentation**.

Since the invariance of LieTransformer only holds in expectation, we measured the invariance error as a function of the number of lift samples, finding the error to decrease monotonically with increasing lift samples.



Training data Test data	D_{train} D_{test}	D_{train} D_{test}^{T2}	D_{train} D_{test}^{SE2}	D_{train}^{T2} D_{test}^{T2}	D_{train}^{T2} D_{test}^{SE2}	D_{train}^{SE2} D_{test}^{SE2}
SetTransformer	0.58$_{\pm 0.07}$	0.44$_{\pm 0.02}$	0.44$_{\pm 0.02}$	0.61$_{\pm 0.02}$	0.51$_{\pm 0.01}$	0.55$_{\pm 0.01}$
LieTransformer-T2	**0.75**$_{\pm 0.03}$	**0.75**$_{\pm 0.03}$	0.63$_{\pm 0.06}$	**0.75**$_{\pm 0.03}$	0.63$_{\pm 0.06}$	0.70$_{\pm 0.03}$
LieTransformer-SE2	0.71$_{\pm 0.01}$	0.71$_{\pm 0.01}$	**0.69**$_{\pm 0.02}$	0.71$_{\pm 0.01}$	**0.69**$_{\pm 0.02}$	**0.72**$_{\pm 0.04}$

Molecular Regression on QM9

The QM9 task (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014) aims to model 12 molecular properties of 133,885 small organic molecules comprised of Carbon, Hydrogen, Oxygen, Nitrogen and Fluorine.

LieTransformer performs best amongst general-purpose invariant models for molecular regression on 8 of 12 tasks in QM9, with small improvements over the most comparable work, LieConv (Finzi et al., 2020).

Task Units	α bohr^3	$\Delta\epsilon$ meV	ϵ_{HOMO} meV	ϵ_{LUMO} meV	μ D	C_ν cal/mol K	G meV	H meV	R^2 bohr^2	U meV	U_0 meV	ZPVE meV
WaveScatt (Hirn, Mallat, and Poilvert, 2017)	.160	118	85	76	.340	.049	–	–	–	–	–	–
NMP (Gilmer et al., 2017)	**.092**	69	43	38	**.030**	.040	19	17	.180	20	20	**1.50**
SchNet (Schütt et al., 2017)	.235	**63**	**41**	**34**	.033	**.033**	**14**	**14**	**.073**	**19**	**14**	1.70
Cormorant (Anderson, Hy, and Kondor, 2019)	.085	61	34	38	.038	.026	20	21	.961	21	22	2.03
DimeNet++ (Klicpera et al., 2020) *	**.049**	**34**	**26**	**20**	**.033**	**.024**	**8**	**7**	.387	**7**	**7**	**1.23**
L1Net (Miller et al., 2020)	.088	68	45	35	.043	.031	14	14	**.354**	14	13	1.56
TFN (Thomas et al., 2018)	.223	58	40	38	.064	.101	–	–	–	–	–	–
SE3-Transformer (Fuchs et al., 2020)	.148	53	36	33	.053	.057	–	–	–	–	–	–
LieConv-T3 (Finzi et al., 2020) †	.125	60	36	32	.057	.046	35	37	1.54	36	35	3.62
LieConv-T3 + SO3 Aug (Finzi et al., 2020)	.084	49	30	**25**	**.032**	.038	22	24	.800	19	19	2.28
LieConv-SE3 (Finzi et al., 2020)†	.097	**45**	**27**	**25**	.039	.041	39	46	2.18	49	48	3.27
LieConv-SE3 + SO3 Aug (Finzi et al., 2020)†	.088	**45**	**27**	**25**	.038	.043	47	46	2.12	44	45	3.25
LieTransformer-T3 (Us)	.179	67	47	37	.063	.046	27	29	.717	27	28	2.75
LieTransformer-T3 + SO3 Aug (Us)	**.082**	51	33	27	.041	**.035**	**19**	**17**	**.448**	**16**	**17**	**2.10**
LieTransformer-SE3 (Us)	.104	52	33	29	.061	.041	23	27	2.29	26	26	3.55
LieTransformer-SE3 + SO3 Aug (Us)	.105	52	33	29	.062	.041	22	25	2.31	24	25	3.67

Hamilonian Dynamics

The **Hamiltonian**, $\mathcal{H}$, of a system specifies the total energy of the system for a given set of **positions**, $\mathbf{p}$, and **momenta**, $\mathbf{q}$. The dynamics of a conservative system are described by Hamilton's equations:

$$\frac{d\mathbf{q}}{dt} = \frac{d\mathcal{H}}{d\mathbf{p}}, \quad \frac{d\mathbf{p}}{dt} = \frac{d\mathcal{H}}{d\mathbf{q}}.$$

Conserved physical quantities such as total energy, momentum, and angular momentum correspond to **invariances of the Hamiltonian** (Noether's theorem).

We can learn the Hamiltonian of a system from observations of its states over time. By building in invariance of the network learning the Hamiltonian of the system, the resulting model better incorporates the inductive biases arising from physical laws.

We observe that LieTransformer is **more data efficient** than the invariant LieConv model and non-invariant baselines. LieTransformer is also **more parameter efficient** than LieConv when incorporating the precise invariances of the task.