Sinkformers: Transformers with Doubly Stochastic Attention (AISTATS2022)

Michael E. Sander, Pierre Ablin.etc.

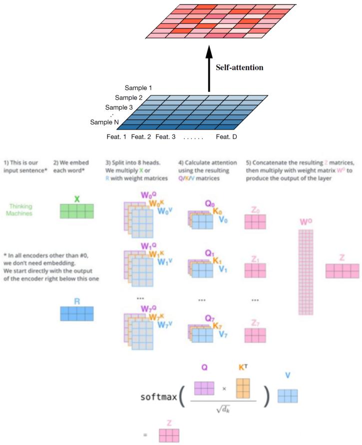
LieTransformer: Equivariant Self-Attention for Lie Groups (ICML2021)

Michael Hutchinson, Charline Le Lan.etc.

Presenter: Minjie Cheng

Outline

Overview of Self-Attention



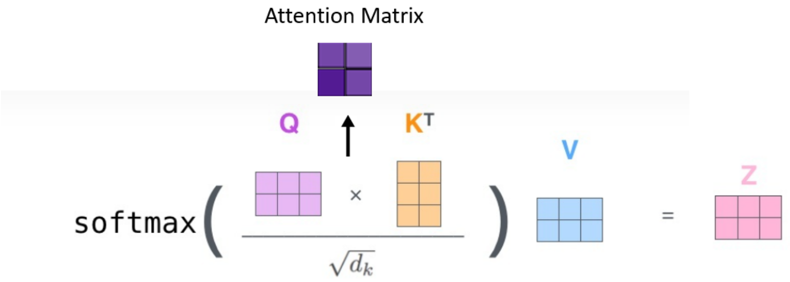
- A **mapping** $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n] \in \mathbb{R}^{N \times D} \mapsto \boldsymbol{Z} \in \mathbb{R}^{N \times D}$
- Using m to index the head $(m = 1, ..., M)$, the output of the mth head can be writen as:

$$f^m(\boldsymbol{X}) \triangleq WXW^{V,m} \in \mathbb{R}^{N \times D/M}$$

$$W \triangleq softmax(\boldsymbol{XW}^{Q,m}(\boldsymbol{XW}^{K,m})^T) \in \mathbb{R}^{N \times N}$$

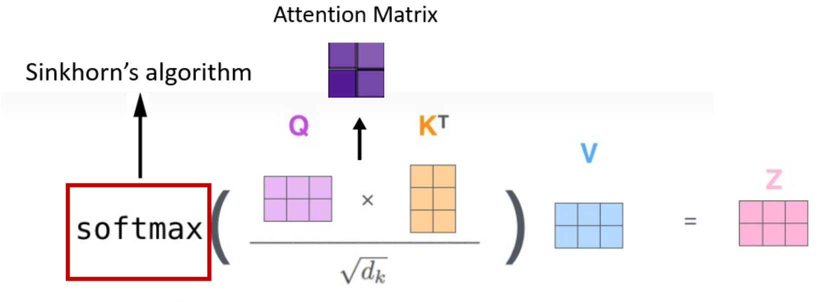
$$MSA(\boldsymbol{X}) \triangleq [f^1(\boldsymbol{X}), ..., f^M(\boldsymbol{X})]W^O \in \mathbb{R}^{N \times D}$$

Motivation-Sinkformers



- This attention matrix is normalized with the SoftMax operator, which makes it row-wise stochastic

Motivation-Sinkformers

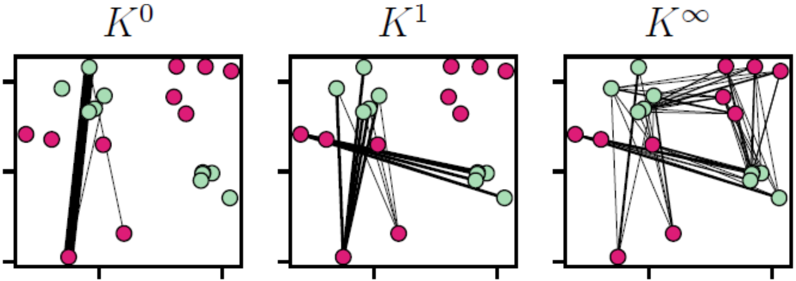


- This attention matrix is normalized with the SoftMax operator, which makes it row-wise stochastic
- Use Sinkhorn's algorithm to make attention matrices **doubly** stochastic.(i.e., rows and columns both sum to 1)

Important Notations

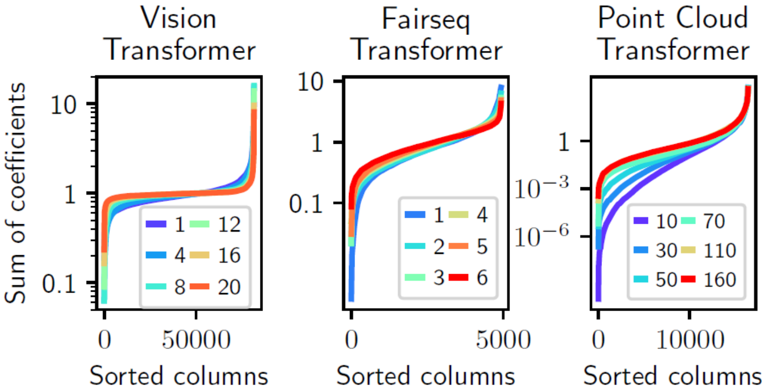
- $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n] \in \mathbb{R}^{N \times D}$
- $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \sum_{j=1}^{n} \boldsymbol{K}^1_{i,j} \boldsymbol{W}_V \boldsymbol{x}_j,$
  - $\boldsymbol{K}^1 := Softmax(\boldsymbol{C})$
  - $C_{i,j} = (\boldsymbol{W}_Q \boldsymbol{x}_i)^T \boldsymbol{W}_K \boldsymbol{x}_j$
  - Query, key, value matrices are $\boldsymbol{W}_Q \in \mathbb{R}^{m \times d}$, $\boldsymbol{W}_K \in \mathbb{R}^{m \times d}$, $\boldsymbol{W}_V \in \mathbb{R}^{d \times d}$
  - Denote $\boldsymbol{K}^0 := exp(\boldsymbol{C})$
  - $K^1_{ij} := K^0_{ij} / \sum_{l=1}^{n} K^0_{il}$
  - $\boldsymbol{K}^1$ is row-wise stochastic

Illustration of the different normalizations of attention matrices



- $C_{i,j} = (\boldsymbol{W}_Q \boldsymbol{x}_i)^T \boldsymbol{W}_K \boldsymbol{x}_j$(correlation coefficient)
- We form two point clouds $\boldsymbol{W}_Q \boldsymbol{x}_i$ (green) and $\boldsymbol{W}_K \boldsymbol{x}_j$ (red).
- We only display connections with $K_{ij}^k \geq 10^{-12}$
- For K^∞(Sinkhorn), all points are involved in an interaction.

Experiment for Sum over clumns



- We sum over columns of attention matrices at different training epochs (color) when training.
- The majority of columns naturally sum closely to 1.

Sinkhorn's algorithm

$$K^{l+1} = \begin{cases} N_R\left(K^l\right) & \text{if } l \text{ is even} \\ N_C\left(K^l\right) & \text{if } l \text{ is odd} \end{cases}$$

- N_R and N_C correspond to row-wise and column-wise normalizations:

$$(N_R(K))_{i,j} := \frac{K_{i,j}}{\sum_{l=1}^{n} K_{i,l}}$$

$$(N_C(K))_{i,j} := \frac{K_{i,j}}{\sum_{l=1}^{n} K_{l,j}}$$

- Note that it is doubly stochastic in the sense that $K^{\infty}\nVdash_n = \nVdash_n$ and $K^{\infty T}\nVdash_n = \nVdash_n$.

Sinkformers

- Note that $K^1 := \text{SoftMax}(C)$ is precisely the output of Sinkhorn's algorithm after 1 iteration.

$$x_i \leftarrow x_i + \sum_{j=1}^{n} K_{i,j}^{\infty} W_V x_j$$

- Only a few iterations of Sinkhorn are sufficient (**typically 3 to 5**) to converge to a doubly stochastic matrix.
- The practical training time of Sinkformers is comparable to regular Transformers.

Invariance to the cost function

Proposition 1. Let $C \in \mathbb{R}^{n\times n}$. Consider, for $(f, g) \in \mathbb{R}^n \times \mathbb{R}^n$ the modified cost function $\tilde{C}_{i,j} := C_{i,j} + f_i + g_j$. Then Sinkhorn $(C) = \text{Sinkhorn}(\tilde{C})$.

$$\tilde{C}_{i,j} := -\frac{1}{2}\left\|W_Q x_i - W_K x_j\right\|^2$$

- We can consider the cost $\tilde{C}_{i,j}$ instead of $C_{i,j} = \left(W_Q x_i\right)^\top W_K x_j$, without affecting K^∞.

Residual maps for attention-continuous counterparts

- We denote $c(x, x') := (W_Q x)^\top W_K x'$ and $k^0 := \exp(c)$. For some measure $\mu \in \mathcal{M}(\mathbb{R}^d)$, we define the SoftMax operator on the cost c by $k^1(x, x') = \mathrm{SoftMax}(c)(x, x') := \frac{k^0(x,x')}{\int k^0(x,y)d\mu(y)}$. Similarly, we define Sinkhorn's algorithm as the following iterations, starting from $k^0 = \exp(c)$:

$$k^{l+1}(x, x') = \begin{cases} \frac{k^l(x,x')}{\int k^l(x,y)d\mu(y)} & \text{if } l \text{ is even} \\ \frac{k^l(x,x')}{\int k^l(y,x)d\mu(y)} & \text{if } l \text{ is odd.} \end{cases}$$

We denote $k^\infty := \mathrm{Sinkhorn}(c)$ the resulting limit. Note that if μ is a discrete measure supported on a n sequence of particles $(x_1, x_2, \ldots, x_n)$, $\mu = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$, then for all (i,j), $k^0(x_i, x_j) = K^0_{i,j}$, $k^1(x_i, x_j) = K^1_{i,j}$ and $k^\infty(x_i, x_j) = K^\infty_{i,j}$, so that k^0, k^1 and k^∞ are indeed the **continuous equivalent** of the matrices K^0, K^1 and K^∞ respectively.

depth limit

- Transformer equation

$$x_i \leftarrow x_i + \sum_{j=1}^{n} K^1_{i,j} W_V x_j$$

- ResNet equation

$$x_i \leftarrow x_i + T(x_i)$$

- Sinkformer equation

$$x_i \leftarrow x_i + \sum_{j=1}^{n} K^\infty_{i,j} W_V x_j$$

Infinitesimal step-size regime

- In this framework, iterating the Transformer equation, the ResNet equation and the Sinkformer equation corresponds to a Euler discretization with step-size 1 of the ODEs

$$\dot{x}_i = T_\mu(x_i) \text{ for all } i,$$

- where $x_i(t)$ is the position of x_i at time t. For an arbitrary measure $\mu \in \mathcal{M}(\mathbb{R}^d)$, these ODEs can be equivalently written as a continuity equation

$$\partial_t \mu + \operatorname{div}(\mu T_\mu) = 0$$

- When T_μ is defined by the ResNet equation 2 , $T_\mu = T$ does not depend on μ. It defines an advection equation where the particles do not interact and evolve independently.
- $T_\mu^1(x) = \int k^1(x, x') W_{V'} x' d\mu(x')$
- $T_\mu^\infty(x) = \int k^\infty(x, x') W_{V'} x' d\mu(x')$

Implementation details

- We implement Sinkhorn's algorithm in log domain for stability. Given a matrix $K^0 \in \mathbb{R}^{n \times n}$ such that $K^0_{i,j} = e^{C_{i,j}}$ for some $C \in \mathbb{R}^{n \times n}$, Sinkhorn's algorithm approaches $(f, g) \in \mathbb{R}^n \times \mathbb{R}^n$ such that $K^\infty = \operatorname{diag}\left(e^{f^\infty}\right) K^0 \operatorname{diag}\left(e^{g^\infty}\right)$ by iterating in log domain, starting from $g^0 = \mathbb{O}_n$,

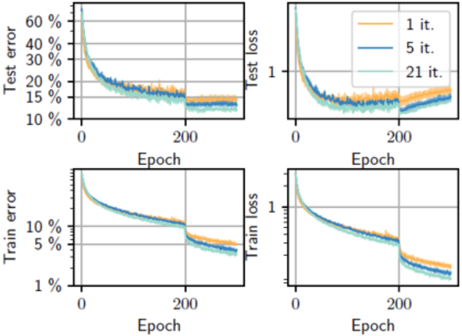
$$f^{l+1} = \log\left(\mathbb{1}_n / n\right) - \log\left(K e^{g^l}\right) \quad \text{if } l \text{ is even}$$

$$g^{l+1} = \log\left(\mathbb{1}_n / n\right) - \log\left(K^T e^{f^l}\right) \quad \text{if } l \text{ is odd.}$$

This allows for fast and accurate computations, where $\log\left(K e^{g^l}\right)$ and $\log\left(K^\top e^{f^l}\right)$ are computed using log-sum-exp.

Experiments

- ModelNet 40 classification



Model	Best	Median	Mean	Worst
Set Transformer	87.8%	86.3%	85.8%	84.7%
Set Sinkformer	**89.1%**	**88.4%**	**88.3%**	**88.1%**
Point Cloud Transformer	**93.2%**	92.5%	92.5%	92.3%
Point Cloud Sinkformer	93.1%	**92.8%**	**92.7%**	**92.5%**

Experiments

- Sentiment Analysis

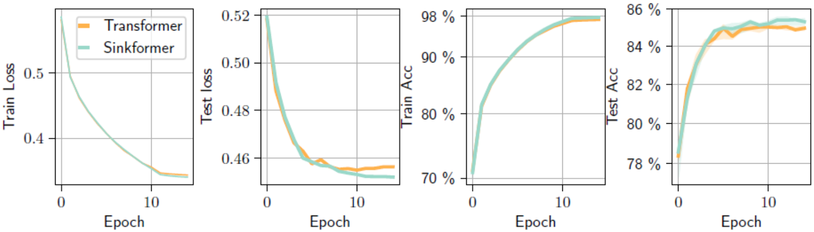
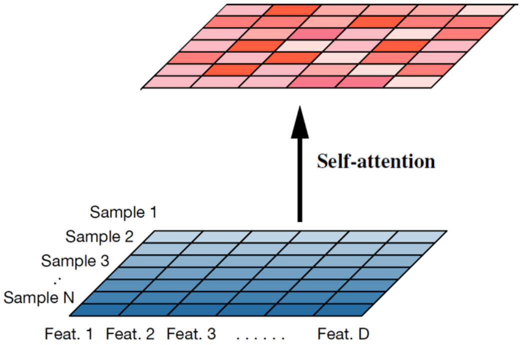


Figure 4: **Learning curves** when training a Transformer and a Sinkformer on the Sentiment Analysis task on the IMDb Dataset.

2. LieTransformer: Equivariant Self-Attention for Lie Groups

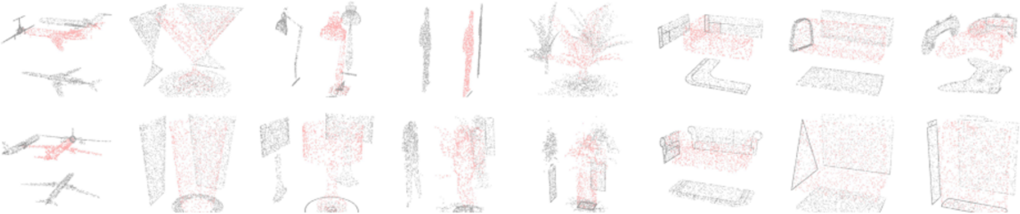
(ICML2021)

Motivation-LieTransformer



- Extend **group equivariance** to self-attention,**non-linear** map

Motivation-LieTransformer



- 3D point clouds
- We may want a classifier to output the same classification when the input is translated or rotated.
- Transformer-based model that is **permutation invariant**, but **not invariant to rotations or translations**.

Equivariant Self-Attention for Lie Groups

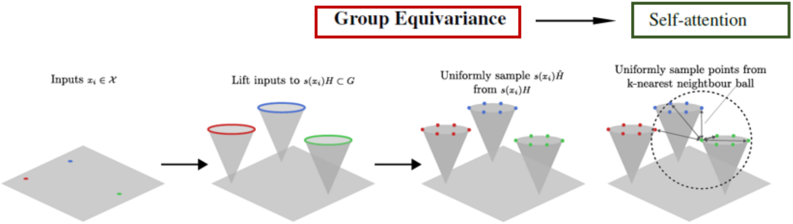


Figure 2. Visualisation of lifting, sampling $\hat{H}$, and subsampling in the local neighbourhood for $SE(2)$ acting on $\mathbb{R}^2$. Self-attention is performed on this subsampled neighbourhood.

Formal definitions for Groups and Representation Theory

- **Group** G is a set of symmetries, with each group element g corresponding to a symmetry transformation, group element $g \in G$
- **Graph representation** $\rho(g)$ takes the form of a matrix. For $SO(2)$,

$$\rho\left(g_\theta\right) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

- Definition. A group G is a set endowed with a single operator $: G \times G \mapsto G$ such that
  1. Associativity: $\forall g, g', g'' \in G, (g \cdot g') \cdot g'' = g \cdot (g' \cdot g'')$
  2. Identity: $\exists e \in G, \forall g \in G, g \cdot e = e \cdot g = g$
  3. Invertibility: $\forall g \in G, \exists g^{-1} \in G, g \cdot g^{-1} = g^{-1} \cdot g = e$

G-equivariance

- Definition 1. We say that a map $\Phi : V_1 \to V_2$ is G equivariant with respect to actions ρ_1, ρ_2 of G acting on V_1, V_2 respectively if: $\Phi\left[\rho_1(g)f\right] = \rho_2(g)\Phi[f]$ for any $g \in G$,$f \in V_1$.
- In the context of group equivariant neural networks, V is commonly defined to be the space of scalar-valued functions on some set S, so that $V = \{f \mid f : S \to \mathbb{R}\}$.
- i.e. a grey-scale image can be expressed as a feature map $f : \mathbb{R}^2 \to \mathbb{R}$ from pixel coordinate x_i to pixel intensity f_i, supported on the grid of pixel coordinates.(Linear operation)

Overview

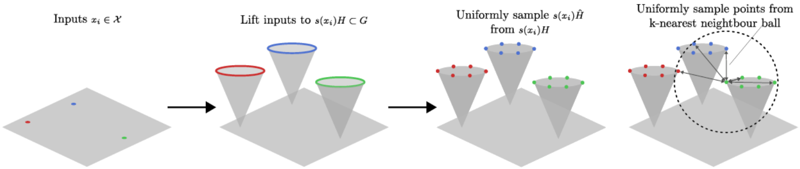


Figure 2. Visualisation of lifting, sampling $\hat{H}$, and subsampling in the local neighbourhood for $SE(2)$ acting on $\mathbb{R}^2$. Self-attention is performed on this subsampled neighbourhood.

Lifting

- Suppose we have data in the form of a set of input pairs $(x_i, \mathbf{f}_i)_{i=1}^{n}$ where $x_i \in \mathcal{X}$ are **spatial coordinates** and $\mathrm{f}_i \in \mathcal{F}$ are **feature values**.
- All elements of $\mathcal{X}$ are connected by the action: $\forall x, x' \in \mathcal{X}, \exists g \in G : \rho(g)x = x'$

LieSelfAttention

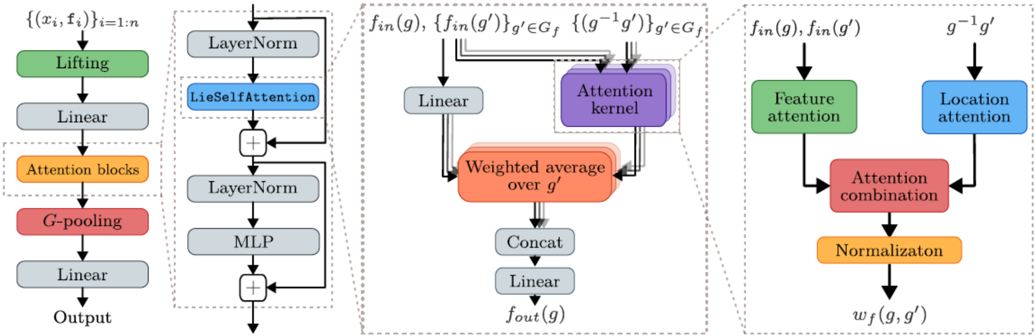


Figure 1. Architecture of the LieTransformer.

Content-based attention

- Content-based attention $k_c\left(f(g), f\left(g'\right)\right)$:
  1. Dot-product: $\frac{1}{\sqrt{d_v}}\left(W^Q f(g)\right)^{\top} W^K f\left(g'\right) \in \mathbb{R}$ for $W^Q, W^K \in \mathbb{R}^{d_v \times d_v}$
  2. Concat: $\operatorname{Concat}\left[W^Q f(g), W^K f\left(g'\right)\right] \in \mathbb{R}^{2 d_v}$
  3. Linear-Concat-linear: $W \operatorname{Concat}\left[W^Q f(g), W^K f\left(g'\right)\right] \in \mathbb{R}^{d_s}$ for $W \in \mathbb{R}^{d_s \times 2 d_v}$.

Location-based attention

- Location-based attention $k_l\left(g^{-1}g'\right)$ for Lie groups G :
  1. Plain: $\nu\left[\log\left(g^{-1}g'\right)\right]$
  2. MLP: MLP $\left(\nu\left[\log\left(g^{-1}g'\right)\right]\right)$

Experiments-Counting Shapes in 2D Point Clouds

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 ± 0.07	0.44 ± 0.02	0.44 ± 0.02	0.61 ± 0.02	0.51 ± 0.01	0.55 ± 0.01
LieTransformer-T2	0.75 ± 0.03	0.75 ± 0.03	0.63 ± 0.06	0.75 ± 0.03	0.63 ± 0.06	0.70 ± 0.03
LieTransformer-SE2	0.71 ± 0.01	0.71 ± 0.01	0.69 ± 0.02	0.71 ± 0.01	0.69 ± 0.02	0.72 ± 0.04

Table 1. Mean and standard deviation of test accuracies on the shape counting task at convergence (over 8 random initialisations).

- Transformer-based model that is **permutation invariant**, but **not invariant to rotations or translations**.
- G-equivariance: rotations, translation

Experiments-QM9: Molecule Property Regression

Task	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_ν	G	H	R^2	U	U_0	ZPVE
Units	bohr^3	meV	meV	meV	D	cal/mol K	meV	meV	bohr^2	meV	meV	meV
WaveScatt (Hirn et al., 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**	<u>.073</u>	**19**	**14**	1.70
Cormorant (Anderson et al., 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	7.7	7.1	.387	6.7	6.9	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

- **non-invariant** models specifically designed for the QM9 task.
- **invariant models** specifically designed for the QM9 task.

Summary and Future Work

- Mapping
- UOT-based module replace the self-attention modules in Transformer.