SE(3)-Hyena Operator for Scalable Equivariant Learning

Artem Moskalev [1] **Mangal Prakash** [1] **Rui Liao** [1] **Tommaso Mansi** [1]

Abstract

Modeling global geometric context while maintaining equivariance is crucial for accurate predictions in many fields such as biology, chemistry, or vision. Yet, this is challenging due to the computational demands of processing high-dimensional data at scale. Existing approaches such as equivariant self-attention or distance-based message passing, suffer from quadratic complexity with respect to sequence length, while localized methods sacrifice global information. Inspired by the recent success of state-space and long-convolutional models, in this work, we introduce SE(3)-Hyena operator, an equivariant long-convolutional model based on the Hyena operator. The SE(3)-Hyena captures global geometric context at sub-quadratic complexity while maintaining equivariance to rotations and translations. Evaluated on equivariant associative recall and n-body modeling, SE(3)-Hyena matches or outperforms equivariant self-attention while requiring significantly less memory and computational resources for long sequences. Our model processes the geometric context of $20k$ tokens $\times 3.5$ faster than the equivariant transformer and allows $\times 175$ longer a context within the same memory budget.

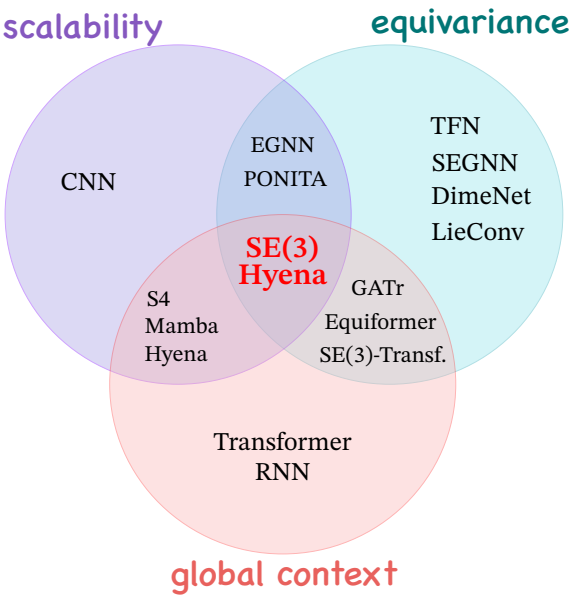

Figure 1. **SE(3)-Hyena operator marries global context, equivariance and scalability towards long sequences.** SE(3)-Hyena operator can process global geometric context in sub-quadratic time while preserving equivariance to rotations and translations.

1. Introduction

Modeling global geometric context while preserving equivariance is crucial in many real-world tasks. The properties of a protein depend on the global interaction of its residues (Baker & Sali, 2001). Similarly, the global geometry of DNA and RNA dictates their functional properties (Leontis & Westhof, 2001; Sato et al., 2021). In computer vision, modeling global geometric context is crucial when working with point clouds or meshes (Thomas et al., 2018; De Haan et al., 2020). In all these tasks, maintaining equivariance while capturing global context is essential for robust modeling and prediction.

Processing global geometric context with equivariance is challenging due to the computational demands of processing high-dimensional data at scale. Existing methods either rely on global all-to-all operators such as self-attention (Liao & Smidt, 2023; de Haan et al., 2024; Brehmer et al., 2023), which do not scale well due to their quadratic complexity, or they restrict processing to local neighborhoods (Thomas et al., 2018; Köhler et al., 2020; Fuchs et al., 2020), losing valuable global information. This limitation is a significant practical bottleneck, necessitating more efficient solutions for scalable equivariant modeling with a global geometric context.

An efficient algorithm for modeling global context should support parallelization during training while maintaining bounded computational costs relative to sequence length during inference. One approach involves recurrent operators (Orvieto et al., 2023; De et al., 2024), which provide

[1]Johnson and Johnson Innovative Medicine. Correspondence to: Artem Moskalev <ammoskalevartem@gmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

bounded compute but lack easy parallelization. Another family of methods relies on self-attention (Vaswani et al., 2017) allowing parallel processing at the cost of quadratic computational complexity. The most recent advances leverage state-space (Gu et al., 2021b; Fu et al., 2022; Gu & Dao, 2023) and long-convolutional (Romero et al., 2021; Poli et al., 2023) frameworks, enabling global context reasoning in sub-quadratic time with easy parallelization. Extending these models to accommodate equivariance remains an unexplored direction.

Inspired by the success of state-space and long-convolutional methods, in this work we propose a SE(3)-equivariant long-convolutional model based on the recent Hyena operator (Poli et al., 2023). The SE(3)-Hyena efficiently models global geometric context in sub-quadratic time while preserving equivariance to rotations and translations (Figure 1). Central to our method is the equivariant vector long convolution that leverages cross products between equivariant queries and keys. This vector convolution can be implemented in the Fourier domain with a computational complexity of $O(Nlog_2N)$, enabling scaling to much longer sequences than self-attention. We evaluate the proposed SE(3)-Hyena against its self-attention counterpart on a novel equivariant associative recall benchmark and the n-body dynamical system modeling task. Our results suggest that SE(3)-Hyena matches or surpasses equivariant self-attention in performance while requiring less memory and compute for long sequences. In particular, for a sequence of $20k$ tokens, the equivariant Hyena runs $\times 3.5$ faster than the equivariant transformer approach. Notably, when the equivariant transformer runs out of memory on sequences over $20k$ tokens, *our model can handle up to* $3.5M$ *million tokens on a single GPU*, providing up to 175 times longer context length within the same computational budget.

To sum up, we make the following contributions:

- We propose SE(3)-equivariant Hyena operator which enables modeling global geometric context in sub-quadratic time.
- We propose an equivariant counterpart for the mechanistic interpretability associative recall task.
- We demonstrate that the equivariant Hyena matches or outperforms the equivariant transformer, while requiring significantly less memory and compute for long-context modeling.

2. Related work

Equivariance Equivariance to group transformations, particularly rotations and translations in 3D, is crucial for modeling physical systems (Zhang et al., 2023). Schütt et al. (2017) condition continuous convolutional filters on relative distances to build model invariant to rotations and translations. Thomas et al. (2018); Fuchs et al. (2020); Brandstetter et al. (2021); Liao & Smidt (2023); Bekkers et al. (2023) utilize spherical harmonics as a steerable basis which enables equivariance between higher-order representations. Since computing spherical harmonics can be expensive, Jing et al. (2021b;a); Satorras et al. (2021); Deng et al. (2021) focus on directly updating vector-valued features to maintain equivariance. Another recent line of work (Ruhe et al., 2023a;b; Brehmer et al., 2023; Zhdanov et al., 2024) employs geometric algebra representation which natively provides a flexible framework for processing symmetries in the data (Dorst et al., 2009).

While these works focus on how to build equivariance into a neural network, in this paper we focus on efficient equivariance to model global geometric contexts at scale.

Modeling geometric context Various strategies are employed to process context information in geometric data. Convolutional methods aggregate context linearly within a local neighborhood, guided by either a graph topology (Kipf & Welling, 2016) or spatial relations in geometric graphs (Schütt et al., 2017; Wu et al., 2019; Thomas et al., 2018). Message-passing framework (Gilmer et al., 2017) generalizes convolutions, facilitating the exchange of nonlinear messages between nodes with learnable message functions. These approaches are favored for their simplicity, balanced computational demands, and expressiveness (Wu et al., 2020). However, they are limited to local interactions and are known to suffer from oversmoothing (Rusch et al., 2023). This hinders building deep message-passing networks capable of encompassing a global geometric context in a receptive field. To address these limitations, recent methods have turned to self-attention mechanisms for graph (Yang et al., 2021; Kreuzer et al., 2021; Kim et al., 2022; Rampášek et al., 2022) and geometric graph (Fuchs et al., 2020; Liao & Smidt, 2023; Brehmer et al., 2023) data, outperforming convolutional and message-passing approaches. Yet, the quadratic computational cost of self-attention poses significant challenges when modeling large-scale physical systems. In this work, we aim to develop a method for global geometric context processing with sub-quadratic computational complexity.

State-space and long-convolutional models The quadratic computational complexity of self-attention has driven the exploration of alternatives for modeling long context. Structured state-space models (Gu et al., 2021b) have emerged as a promising alternative, integrating recurrent and convolutional mechanisms within a single framework. These models enable parallelized training in a convolutional mode and maintain linear complexity with respect to sequence length in a recurrent mode.

Models like S4 (Gu et al., 2021a), H3 (Fu et al., 2022), and Mamba (Gu & Dao, 2023; Li et al., 2024) have consistently matched or exceeded transformer performance in diverse tasks such as genomics (Schiff et al., 2024), long-range language (Wang et al., 2024), and vision tasks (Zhu et al., 2024). Concurrently, another line of work integrates long-convolutional framework with implicit filters (Sitzmann et al., 2020; Romero et al., 2021) to capture global sequence context. The implicit filter formulation allows for data-controlled filtering similar to transformers, while FFT-based long convolution enables global context aggregation in sub-quadratic time (Poli et al., 2023). Such models have shown competitive performance comparable to state-space and transformer architectures in time-series modeling (Romero et al., 2021), genomics (Nguyen et al., 2024), and vision tasks (Poli et al., 2023).

Although state-space and long-convolutional methods dramatically reduced the computational costs associated with processing long sequences, their application to geometric data requiring equivariance remains unexplored. In this work, we adapt the recently proposed Hyena operator (Poli et al., 2023) to incorporate SE(3) equivariance. To the best of our knowledge, this is the first equivariant long-convolutional model that can process global geometric contexts with sub-quadratic memory and time requirements.

3. Method

We consider tasks that require modeling invariant and equivariant features in geometric graphs. A geometric graph of N nodes is represented by a set of features $\{\mathbf{x}_i, \mathbf{f}_i\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^3$ represents vector features (e.g. coordinates or velocities), and $\mathbf{f}_i \in \mathbb{R}^S$ represents scalar features (e.g. atom types, charges or fingerprints). We call $\mathbf{x}_i$ geometric or vector tokens, and $\mathbf{f}_i$ are scalar tokens. When working with geometric graphs, a neural network must respect symmetries of the input space such as rotation or translation. That means, a model must be equivariant with respect to geometric tokens and invariant with respect to scalar tokens.

3.1. SE(3)-Hyena operator

SE(3)-Hyena operator consists of invariant and equivariant streams which are responsible for processing scalar and vector features respectively, as illustrated in Figure 2. This way, the model takes scalar and vector inputs, and outputs processed scalar and vector features.

Formally, $\Psi : \mathbb{R}^3 \times \mathbb{R}^S \to \mathbb{R}^3 \times \mathbb{R}^D$ satisfies equivariance property:

$$\{L_g(\hat{\mathbf{x}}_i), \hat{\mathbf{f}}_i\}_{i=1}^N = \Psi\left(\{L_g(\mathbf{x}_i), \mathbf{f}_i\}_{i=1}^N\right)$$

where $L_g : \mathbb{R}^3 \to \mathbb{R}^3$ is a representation of a group action $g \in SE(3)$. Thus, geometric tokens $\mathbf{x}_i$ transform accordingly with the group action while scalar tokens $\mathbf{f}_i$ remain invariant.

For both invariant and equivariant streams, we make the overall information flow similar to the information flow of a transformer with the key difference in how the global context is aggregated. Akin to the transformer architecture, the input is firstly projected into keys, queries, and values. Next, the global context is aggregated via long convolution and gating. Finally, the global context is combined with values, and then projected via a feed-forward network.

3.2. SE(3)-Hyena modules

We design each layer to be equivariant with respect to transformations of geometric tokens, and invariant for scalar tokens. This way, a model consisting of a composition of equivariant layers is also equivariant by induction (Weiler & Cesa, 2019).

Input projection Similar to transformers, we firstly project input tokens to queries, keys and values. Because our SE(3)-Hyena operator consists of equivariant and invariant streams, we need to obtain queries, keys and values for both invariant and equivariant features. We define the projection layer $\phi : \mathbb{R}^3 \times \mathbb{R}^S \to (\mathbb{R}^3)^3 \times (\mathbb{R}^D)^3$ as $\mathbf{z}_i^{eqv}, \mathbf{z}_i^{inv} = \phi(\mathbf{x}_i, \mathbf{f}_i)$ where for i-th geometric token $\mathbf{z}_i^{eqv} = [\mathbf{q}_i^{eqv}, \mathbf{k}_i^{eqv}, \mathbf{v}_i^{eqv}]$ is equivariant query, key, and value, each represented by a vector in $\mathbb{R}^3$, and $\mathbf{z}_i^{inv} = [\mathbf{q}_i^{inv}, \mathbf{k}_i^{inv}, \mathbf{v}_i^{inv}]$ represents invariant query, key and value of dimensions $\mathbb{R}^D$ for i-th scalar token. This way, the projection layer emits both scalar and vector query, key, value triplets while also allowing interaction between equivariant and invariant subspaces.

To simultaneously preserve equivariance for geometric tokens and invariance for scalar tokens, while allowing interaction between them, we adopt E(n)-equivariant Clifford MLP (Ruhe et al., 2023a) as the input projection function. The inputs are firstly embedded into $CL(\mathbb{R}^3, q)$ Clifford algebra, and then processed via a geometric linear layer, geometric product, and normalization layers. Finally, the output is projected back to scalar and vector features by grade-one and grade-two projections respectively.

Scalar long convolution To allow global context aggregation for invariant scalar features, we rely on long convolution (Romero et al., 2021; Poli et al., 2023) between query and key tokens. We treat the queries as input signal projection, and the keys constitute a data-controlled implicit filter. Similar to Romero et al. (2021); Poli et al. (2023), we employ circular FFT-convolution to reduce the computational complexity. Let $\mathbf{q}^{inv}$ and $\mathbf{k}^{inv}$ be two sequences of length N composed of sets of one-dimensional invariant

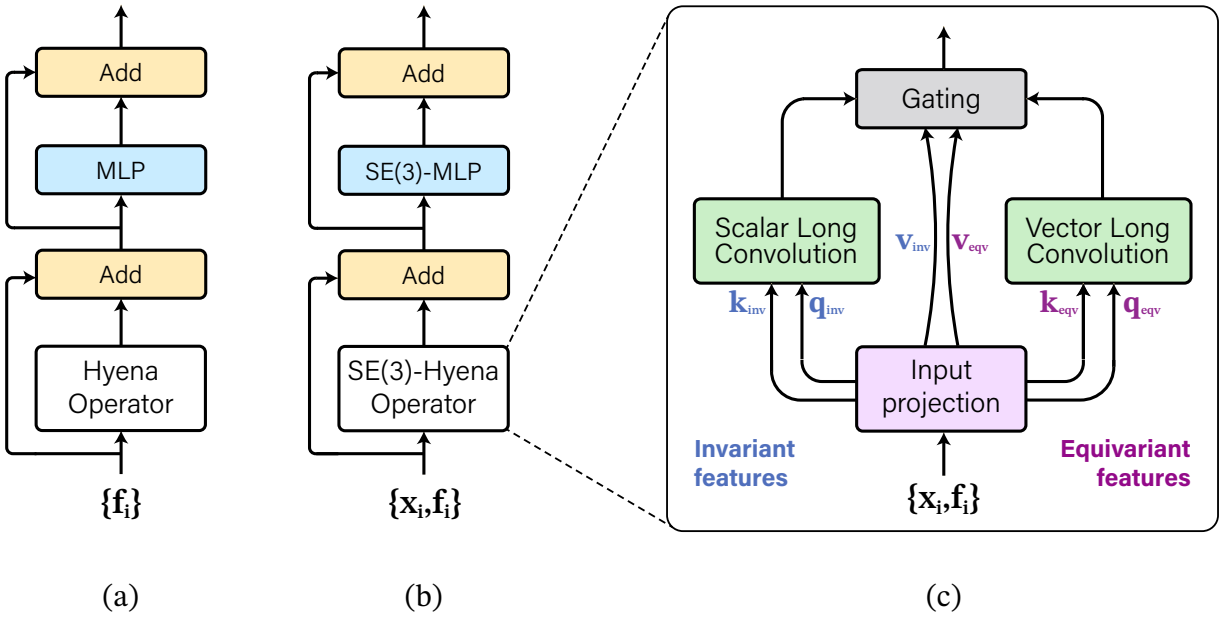

Figure 2. **SE(3)-Hyena building blocks. (a)** Schematic of existing Hyena architecture (Poli et al., 2023). **(b)** The proposed architecture consists of the SE(3)-Hyena operator, residual connections, and an equivariant MLP. **(c)** The block architecture of SE(3)-Hyena operator consists of two streams processing invariant and equivariant features. The key components are scalar and vector long convolution responsible for global context aggregation.

queries $\{\mathbf{q}_i^{inv}\}_{i=1}^N$ and keys $\{\mathbf{k}_i^{inv}\}_{i=1}^N$ respectively. Then, the global context can be aggregated by FFT-convolution as:

$$\begin{aligned}\mathbf{q}^{inv} \circledast \mathbf{k}^{inv} &= \mathbf{F}^H \mathbf{\Lambda}_{\mathbf{k}} \mathbf{F} \mathbf{q}^{inv} \\ &= \mathbf{F}^H \text{diag}(\mathbf{F}\mathbf{k}^{inv}) \mathbf{F} \mathbf{q}^{inv}\end{aligned} \tag{1}$$

where $\mathbf{F}$ is a discrete Fourier transform matrix, and $\text{diag}(\mathbf{F}\mathbf{k}^{inv})$ is a diagonal matrix containing Fourier transform of the kernel $\mathbf{k}^{inv}$.

In the case when query's and key's dimension $D > 1$, the scalar FFT-convolution runs separately for each dimension, rendering computational complexity of $O(DNlog_2N)$ that is sub-quadratic in sequence length.

SE(3) vector long convolution To allow global context aggregation for geometric tokens, we build equivariant vector long convolution. While a scalar convolution relies on dot-products between scalar signals, vector convolution operates with vector cross products $\times$ between vector signals. Formally, given a vector signal consisting of N vector tokens $\mathbf{q}^{eqv} \in \mathbb{R}^{N\times 3}$ and a vector kernel $\mathbf{k}^{eqv} \in \mathbb{R}^{N\times 3}$, we define the vector long-convolution as:

$$\left(\mathbf{q}^{eqv} \circledast_{\times} \mathbf{k}^{eqv}\right)_i = \sum_{j=1}^{N} \mathbf{q}_i^{eqv} \times \mathbf{k}_{j-i}^{eqv} \tag{2}$$

The computational complexity of a naive implementation for the vector convolution is quadratic since both the signal and a kernel are of the full length. To reduce the computational complexity, we show how the vector convolution can be formulated as a series of scalar convolutions that can be efficiently carried out by the FFT. This is due to the fact that a cross product can be written element-wise through the series of scalar products as $(\mathbf{a} \times \mathbf{b})\,[l] = \varepsilon_{lhp}\mathbf{a}[h]\mathbf{b}[p]$ where ε is *Levi-Civita* symbol, and $\mathbf{a}[h]$ denotes a projection onto h-th basis vector. Thus, the l-th component of the vector convolution in Equation 2 can be written element-wise as:

$$\begin{aligned}\left(\mathbf{q}^{eqv} \circledast_{\times} \mathbf{k}^{eqv}\right)_i [l] &= \varepsilon_{lhp} \sum_{j=1}^{N} \mathbf{q}_i^{eqv}[h]\, \mathbf{k}_{j-i}^{eqv}[p] \\ &= \varepsilon_{lhp} \left(\mathbf{q}^{eqv}[h] \circledast \mathbf{k}^{eqv}[p]\right)_i\end{aligned} \tag{3}$$

Thus, we can obtain l-th component of a resulting vector signal via a scalar convolution over the h-th and p-th components of the sequences $\mathbf{q}^{eqv}$ and $\mathbf{k}^{eqv}$ respectively. Since the scalar convolution can be implemented with the FFT, decomposing the vector convolution to the series of scalar convolutions allows reducing its quadratic complexity to $O(Nlog_2N)$.

Since a cross product is already equivariant to rotations, the whole vector convolution is also equivariant to rotations

provided the queries $\mathbf{q}^{eqv}$ and the keys $\mathbf{k}^{eqv}$ are rotated accordingly. The latter is guaranteed when the input projection function is equivariant. We can further obtain SE(3) equivariance by firstly centering (by subtracting the center of mass) and then uncentering with respect to translation.

Gating Similar to transformers, we aim to enable data-controlled gating. Similar to the input projection layer, we employ Clifford MLP to obtain a gating mask while allowing interaction between invariant and equivariant subspaces. The output of the Clifford MLP $\gamma : \mathbb{R}^3 \times \mathbb{R}^D \to (\mathbb{R})^2$ is two grade-one projected scalar features m_i^{eqv}, m_i^{inv} that are passed through a sigmoid function. The sigmoid outputs are multiplied element-wise with the scalar and vector tokens for the invariant and equivariant streams, respectively. In other words, given the input vector token $\mathbf{x}_i^{eqv}$, the gated token $\hat{\mathbf{x}}_i^{eqv}$ is computed as $\hat{\mathbf{x}}_i^{eqv} = \sigma(m_i^{eqv}) \cdot \mathbf{x}_i^{eqv}$, with scalar tokens processed in a similar manner.

To align the information flow with that of transformers, the gating is applied on top of the long convolution between queries and keys. Finally, the gated tokens are integrated with value tokens $\mathbf{v}_i^{eqv}$ and $\mathbf{v}_i^{inv}$ using cross and element-wise products for scalar and vector tokens respectively.

By employing grade-one output projection in the Clifford MLP to obtain masking values, the gating mechanism remains E(3)-invariant, thus maintaining invariance for scalar tokens and preserving the equivariance of vector tokens.

Output projection We add a residual connection between input tokens and the result from the gating layer, and pass it through the output equivariant Clifford MLP $\xi : \mathbb{R}^3 \times \mathbb{R}^D \to \mathbb{R}^3 \times \mathbb{R}^D$ with grade-one and grade-two projections to extract processed scalar and vector tokens.

3.3. Algorithm

We detail the SE(3)-Hyena algorithm in Algorithm 1. The input projection, gating, and output projection operate in parallel for each token and thus of $O(N)$ complexity. The scalar and vector long convolutions are implemented with FFT and thus come with a complexity of $O(Nlog_2N)$.

Algorithm 1 SE(3)-Hyena forward pass

Require: N input tokens $\mathbf{x} \in \mathbb{R}^{N\times 3}, \mathbf{f} \in \mathbb{R}^{N\times S}$

1. input projection:
$[\mathbf{q}^{eqv}, \mathbf{k}^{eqv}, \mathbf{v}^{eqv}], [\mathbf{q}^{inv}, \mathbf{k}^{inv}, \mathbf{v}^{inv}] = \phi\left(\mathbf{x}, \mathbf{f}\right)$

2. global context aggregation:
$\mathbf{u}^{eqv} = \mathbf{q}^{eqv} \circledast_{\times} \mathbf{k}^{eqv}$
$\mathbf{u}^{inv} = \mathbf{q}^{inv} \circledast \mathbf{k}^{inv}$

3. gating:
$m^{eqv}, m^{inv} = \gamma(\mathbf{u}^{eqv}, \mathbf{u}^{inv})$
$\hat{\mathbf{u}}^{eqv} = \sigma(m^{eqv}) \cdot \mathbf{u}^{eqv}$
$\hat{\mathbf{u}}^{inv} = \sigma(m^{inv}) \cdot \mathbf{u}^{inv}$

4. residual and output projection:
 # the cross product is between each $\hat{\mathbf{h}}_i^{eqv}$ *and* $\hat{\mathbf{v}}_i^{eqv}$
$\hat{\mathbf{x}} = \xi(\mathbf{x} + \hat{\mathbf{u}}^{eqv} \times \mathbf{v}^{eqv})$
$\hat{\mathbf{f}} = \xi(\mathbf{f} + \hat{\mathbf{u}}^{inv} \cdot \mathbf{v}^{inv})$

Return: $\hat{\mathbf{x}} \in \mathbb{R}^{N\times 3}, \hat{\mathbf{f}} \in \mathbb{R}^{N\times D}$

4. Experiments

Overview We conduct experiments to assess the performance of the SE(3)-equivariant Hyena in modeling global geometric context. We start with the associative recall (induction heads) task from mechanistic interpretability (Elhage et al., 2021) that has become a standard for comparing efficiency of models in processing global context. Due to the equivariance requirement of geometric context, we have developed a vector-valued extension of the standard associative recall with equivariance to rotations. Next, we evaluate our model on the n-body dynamical system task, where the focus is on accurately predicting the positions of particles based on their initial velocities and coordinates while maintaining equivariance to rotations. Lastly, we compare the runtime and memory profiles of the equivariant Hyena against the equivariant transformer to highlight the computational efficiency of our method for long sequences.

Baselines As a baseline, we evaluate the SE(3)-Hyena against the equivariant transformer, focusing specifically on global context modeling. To facilitate fair comparison, we aim to minimize architectural differences between the two models, altering only the global context mechanism. In the SE(3)-Hyena, we utilize scalar and vector long convolutions, whereas the transformer employs equivariant vector self-attention. The details on equivariant vector self-attention are in Appendix B. We also include comparisons with non-equivariant versions of both the Hyena and transformer models to underscore the benefits provided by equivariance.

4.1. Equivariant associative recall

Associative recall (Olsson et al., 2022) is one of the standard mechanistic interpretability tasks used for quantifying the contextual learning capabilities of sequence models. In this task, a model is required to perform associative recall and copying; for instance, if a model previously encounters the bigram "Harry Potter" in a sequence, it should accurately predict "Potter" the next time "Harry" appears, by referencing its past occurrence (Gu & Dao, 2023).

To adapt this for geometric contexts, we modify the standard associative recall to accommodate 3D vectors. In this version, the tokens within a bigram (key and value) relate

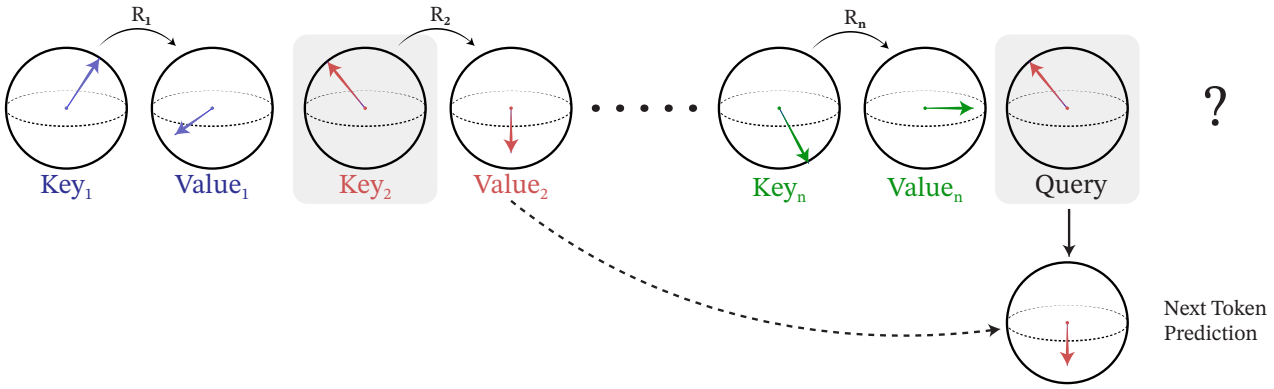

Figure 3. **Equivariant associative recall task.** An equivariant associative recall requires retrieving a vector token for a given vector query based on the context. The retrieval mechanism requires equivariance to rotation of tokens in a sequence. As standard associative recall serves to test the capability of models to learn global context, the equivariant associative recall task serves to test capability of models to learn global context with equivariance.

to each other by a rotation matrix. The model processes a sequence of N vector tokens concluding with a query token[1] and must predict a vector associated with this query seen earlier in the sequence, as illustrated in Figure 3. The task ensures that rotating the entire sequence affects only the orientation of the predicted vector, not the key-value relationship within a bigram, thereby making the task equivariant to rotations. The task complexity depends on the number of tokens in a sequence and on vocabulary size where the vocabulary items are unique key-value vector bigrams.

We propose two versions of the equivariant associative recall task: random and fixed vocabulary versions. In the random vocabulary version, sequences are sampled from a randomly generated vocabulary at each training iteration while validation and test sets are fixed. Thus, an only way for a model to solve this task is to learn an equivariant retrieval mechanism that can associate a given query with a corresponding value vector. In the fixed vocabulary version, the vocabulary is fixed and shared among training, test and validation sets, and the sequences are randomly rotated during the training. This way, the fixed version tests the model's ability to learn underlying vocabulary and generalize it for various orientations.

4.1.1. Implementation Details

Dataset We set the training data size to 2600, and validation and test set size to 200 sequences each. The vocabulary size is set to 4 for both random and fixed versions. The vocabularies are generated as follows: one key-value bigram consists of two vectors with orientations sampled as random unit 3D vectors from an isotropic normal distribution, and with magnitudes randomly sampled from a uniform distribution in a range of $[1, \texttt{vocab_size}]$. We generate datasets with various sequence lengths from 2^5 to 2^8 tokens. When generating sequences from a vocabulary, the last token is a key that corresponds to a target value. We additionally constraint the generation so the target key-value pair is present in a sequence at least once.

Models The vector tokens are used as the input to the equivariant branch, while for the invariant branch, we use positional encoding features (Vaswani et al., 2017) of dimension 16 as the input. The SE(3)-Hyena model consists of 3 SE(3)-Hyena operators followed by a mean pooling of equivariant features and one output equivariant MLP. In SE(3)-Hyena operator we set the hidden dimension to 16 for invariant and to 128 for equivariant streams. In the gating operator, we use a smaller dimension of 8 for both equivariant and invariant features. This ends up in approximately $800k$ trainable parameters. Similarly, the equivariant transformer consists of 3 SE(3)-equivariant vector self-attention blocks followed by a mean pooling and one equivariant MLP at the end. The hidden dimension is kept similar to SE(3)-Hyena which results in a nearly identical number of trainable parameters. Non-equivariant baselines consist of 3 Hyena or standard self-attention blocks but with 3 times larger hidden dimensions to balance the number of trainable parameters with equivariant models.

Training We train all models for 300 epochs with a batch size of 32. We employ Adam optimizer (Kingma & Ba, 2014) with an initial learning rate of 0.001 and cosine learning rate annealing (Loshchilov & Hutter, 2016). The weight decay is set to 0.00001. Mean squared error is used as a loss function. For the fixed vocabulary experiment, we apply

[1]key,value and query tokens here refer to associative recall task and different from keys,values and queries outputted by the input projection

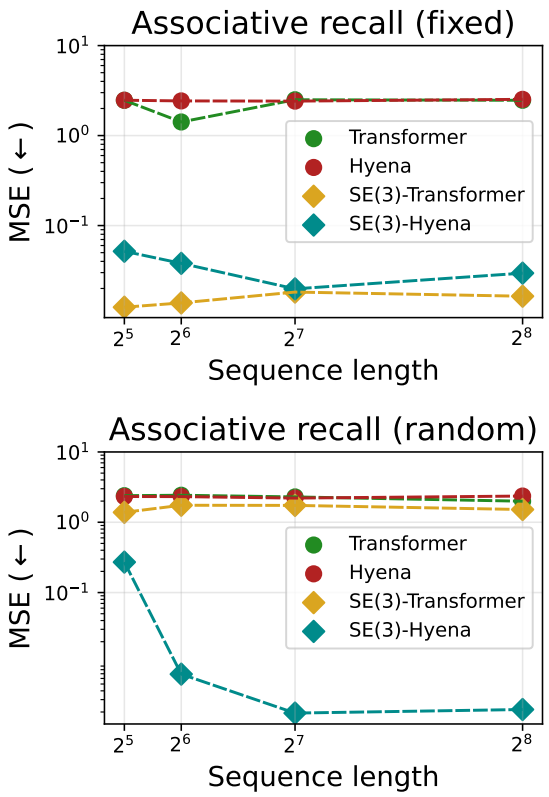

Figure 4. **Top row:** The MSE between retrieved and target vectors for the fixed vocabulary associative recall task is plotted across various sequence lengths. Equivariant models effectively learn and generalize the underlying vocabulary across different orientations. **Bottom row:** The MSE for the random vocabulary associative recall task. The SE(3)-Hyena excels in learning the equivariant retrieval function, successfully associating target queries with their corresponding value vectors.

random rotation augmentation on sequences in the training batch, for the random vocabulary variant this is not necessary as sequences already appear in arbitrary orientations. Final models are selected based on the best validation loss.

4.1.2. Results

The results for various sequence lengths are presented in Figure 4. We record the mean squared error between predicted and ground truth vectors as a performance measure.

For a fixed vocabulary variant, SE(3)-Hyena performs on par with the equivariant transformer across the whole range of sequence lengths. This demonstrates that both equivariant Hyena and transformer models can learn an underlying vocabulary and can generalize it for various orientations. We also observed that non-equivariant models were only able to learn the expectation across the training dataset.

For the random vocabulary variant, we observed that SE(3)-Hyena successfully learns the retrieval function to associate a target query with a corresponding target value vector. We observed better generalization for longer sequences which can be attributed to a higher frequency of target key-value bigram occurrence. Other models struggled to converge, with SE(3)-transformer only slightly outperforming non-equivariant models that only learn the expectation.

4.2. N-body problem

In the field of dynamical systems, comprehending the time evolution of point sets within geometric spaces is crucial for a variety of applications, such as control systems (Chua et al., 2018), model-based dynamics in reinforcement learning (Nagabandi et al., 2018), and simulations of physical systems (Watters et al., 2017). This task inherently requires equivariance, as any rotations and translations applied to the initial set of particles must be consistently represented throughout their entire trajectory.

We benchmark SE(3)-Hyena on forecasting trajectories in the dynamical system where the dynamics is governed by physical interaction between 5 charged particles that carry a positive or negative charge. Particles have a position and a velocity in 3-dimensional space. The objective is to predict the future positions of particles given initial positions and velocities.

4.2.1. Implementation details

Dataset We utilize the standard n-body benchmark from Fuchs et al. (2020); Satorras et al. (2021). We sample 1000 particles for training, 2000 for validation, and 2000 for testing. A training sample consists of equivariant features, i.e. initial positions $\mathbf{p}_0 \in \mathbb{R}^{5\times 3}$ and velocities positions $\mathbf{v}_0 \in \mathbb{R}^{5\times 3}$. Positions of the particles $\mathbf{p}_{1000} \in \mathbb{R}^{5\times 3}$ after 1.000 timesteps are used as labels.

Models and training The initial positions and velocities serve as the input to the equivariant branch, and one-hot encoded charges are used as the input to the invariant branch. The SE(3)-Hyena model consists of 2 SE(3)-Hyena operators followed by one output equivariant MLP. The equivariant transformer baseline consists of 2 vector self-attention blocks and one output equivariant MLP. For equivariant models, the invariant hidden dimension is set to 8 and the equivariant hidden dimension to 16. The gating in SE(3)-Hyena operator uses a hidden dimension of 8 for both equivariant and invariant features. Non-equivariant baselines also consist of 2 Hyena or standard self-attention blocks but with 3 times larger invariant and equivariant hidden dimensions to match the number of trainable parameters with equivariant models. We also use the linear baseline model which is simply a linear motion equitation $\mathbf{p}_t = \mathbf{p}_0 + t \cdot \mathbf{v}_0$. The models are trained for 300 epochs with a batch size of 100. We use Adam optimizer with a learning rate set to 0.0001

Method	MSE
Linear	0.0322
Transformer	0.0163
Hyena	0.0150
SE(3)-Transformer	**0.0019**
SE(3)-Hyena	**0.0018**

Table 1. Mean squared error for the future position estimation in the N-body system experiment.

and weight decay of 0.00001. The mean squared error is used as the loss function.

4.2.2. Results

The results are presented in Table 1. The SE(3)-Hyena model outperforms non-equivariant baselines and performs on par with the equivariant transformer. Interestingly, the non-equivariant Hyena also slightly outperformed the non-equivariant transformer. This may stem from Hyena's structural similarities to SSMs (Poli et al., 2023) which are derived from the state-space representation of differential equation and hence are well-suited for modeling dynamical systems (Hinrichsen & Pritchard, 2005).

4.3. Runtime and memory benchmarks

We benchmark the runtime and memory consumption of a single layer SE(3)-Hyena against a single layer equivariant transformer when processing various sequence lengths. Similar to (Dao et al., 2022; Poli et al., 2023), we use random sequences for the runtime evaluation and we increase the sequence length until the SE(3)-Transformer runs out of memory. We record all runtimes on NVIDIA A10G GPU with CUDA version 12.2.

The comparison is reported in Figure 5, showing forward pass time in milliseconds and total GPU memory utilization in gigabytes. SE(3)-Hyena easily scales to longer sequences whereas SE(3)-Transformer is $3.5\times$ slower than our model for a sequence length of $20k$ tokens. Similarly to Nguyen et al. (2024), we observed Hyena to be slightly slower for shorter sequences. This can be attributed to expensive I/O between layers of the GPU memory hierarchy and can be addressed with the recent GPU-optimized versions of the FFT long convolution (Fu et al., 2023). Regarding memory usage, for the sequences length of $20k$ tokens the equivariant Hyena requires 18 times less GPU memory than transformer for a forward pass. Moreover, we observed that when the equivariant transformer runs out of memory on $> 20k$ tokens, *our model supports up to* $3.5M$ *tokens on a single GPU allowing for* 175 *times longer geometric context*. This memory efficiency is attributed to the FFT long convolution that avoids materializing a quadratic self-attention matrix.

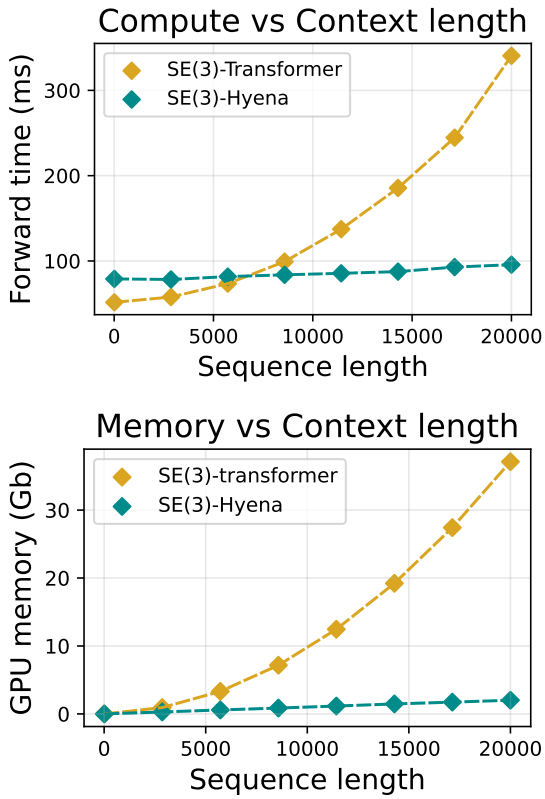

Figure 5. **Top row:** Forward runtime comparison. SE(3)-Hyena scales sub-quadratically and achieves a considerable speedup compared to SE(3)-Transformer when processing long sequences. **Bottom row:** Total GPU memory utilization for equivariant Hyena and transformer models.

5. Conclusions

We introduced the SE(3)-Hyena operator being, to the best of our knowledge, the first equivariant long-convolutional model with sub-quadratic complexity for global geometric context. Through experiments on the dynamical system and novel equivariant associative recall task, we demonstrated that the equivariant long-convolutional model can perform competitively to the equivariant self-attention while requiring a fraction of the computational and memory cost of transformers for long context modeling. Our scalable equivariant model efficiently captures the global context, highlighting its potential for a multitude of future applications in various domains.

Limitations and Future work This work introduces a novel approach for equivariant modeling of global geometric context at scale, with initial experiments designed to validate the fundamental principles of our method. While these experiments confirm the key advantages of our approach, they represent the first step of a comprehensive experimental analysis that is necessary to uncover the model's capabilities

across a wider range of real-world tasks. On a technical side, an interesting direction for future improvement is to adapt the vector convolution to function across arbitrary dimensions as it currently relies on a cross product, which is only feasible in 3 and 7 dimensions (Massey, 1983). Also, our method relies on the FFT convolution that is not permutation equivariant. While this is not a significant limitation in domains such as molecular systems, where canonical ordering is typically available (Jochum & Gasteiger, 1977), it becomes critical in fields like point cloud processing, where establishing a canonical order is challenging. Enhancing the long-convolutional framework to incorporate permutation equivariance could offer substantial advantages in these areas.

References

Baker, D. and Sali, A. Protein structure prediction and structural genomics. *Science*, 294:93 – 96, 2001. URL https://api.semanticscholar.org/CorpusID:7193705.

Bekkers, E. J., Vadgama, S., Hesselink, R. D., van der Linden, P. A., and Romero, D. W. Fast, expressive se (n) equivariant networks through weight-sharing in position-orientation space. *arXiv preprint arXiv:2310.02970*, 2023.

Brandstetter, J., Hesselink, R., van der Pol, E., Bekkers, E. J., and Welling, M. Geometric and physical quantities improve e (3) equivariant message passing. In *International Conference on Learning Representations*, 2021.

Brehmer, J., Haan, P. D., Behrends, S., and Cohen, T. Geometric algebra transformer. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=M7r2CO4tJC.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

De, S., Smith, S. L., Fernando, A., Botev, A., Cristian-Muraru, G., Gu, A., Haroun, R., Berrada, L., Chen, Y., Srinivasan, S., et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024.

De Haan, P., Weiler, M., Cohen, T., and Welling, M. Gauge equivariant mesh cnns: Anisotropic convolutions on geometric graphs. *arXiv preprint arXiv:2003.05425*, 2020.

de Haan, P., Cohen, T., and Brehmer, J. Euclidean, projective, conformal: Choosing a geometric algebra for equivariant transformers. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 27, 2024. URL https://arxiv.org/abs/2311.04744.

Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., and Guibas, L. Vector neurons: a general framework for so(3)-equivariant networks. *arXiv preprint arXiv:2104.12229*, 2021.

Dorst, L., Fontijne, D., and Mann, S. *Geometric algebra for computer science (revised edition): An object-oriented approach to geometry*. Morgan Kaufmann, 2009.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Fu, D. Y., Dao, T., Saab, K. K., Thomas, A. W., Rudra, A., and Ré, C. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

Fu, D. Y., Kumbong, H., Nguyen, E., and Ré, C. Flashfftconv: Efficient convolutions for long sequences with tensor cores. *arXiv preprint arXiv:2311.05908*, 2023.

Fuchs, F., Worrall, D., Fischer, V., and Welling, M. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021a.

Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., and Ré, C. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34: 572–585, 2021b.

Hinrichsen, D. and Pritchard, A. J. *Mathematical systems theory I: modelling, state space analysis, stability and robustness*, volume 48. Springer, 2005.

Jing, B., Eismann, S., Soni, P. N., and Dror, R. O. Equivariant graph neural networks for 3d macromolecular structure. *arXiv preprint arXiv:2106.03843*, 2021a.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., and Dror, R. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=1YLJDvSx6J4.

Jochum, C. and Gasteiger, J. Canonical numbering and constitutional symmetry. *Journal of Chemical Information and Computer Sciences*, 17(2):113–117, 1977.

Kim, J., Nguyen, D., Min, S., Cho, S., Lee, M., Lee, H., and Hong, S. Pure transformers are powerful graph learners. *Advances in Neural Information Processing Systems*, 35: 14582–14595, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Köhler, J., Klein, L., and Noé, F. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pp. 5361–5370. PMLR, 2020.

Kreuzer, D., Beaini, D., Hamilton, W., Létourneau, V., and Tossou, P. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.

Leontis, N. B. and Westhof, E. Geometric nomenclature and classification of rna base pairs. *RNA*, 7(4):499–512, 2001. doi: 10.1017/S1355838201002515.

Li, S., Singh, H., and Grover, A. Mamba-nd: Selective state space modeling for multi-dimensional data. *arXiv preprint arXiv:2402.05892*, 2024.

Liao, Y.-L. and Smidt, T. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KwmPfARgOTD.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Massey, W. S. Cross products of vectors in higher dimensional euclidean spaces. *The American Mathematical Monthly*, 90(10):697–701, 1983.

Nagabandi, A., Kahn, G., Fearing, R. S., and Levine, S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 7559–7566. IEEE, 2018.

Nguyen, E., Poli, M., Faizi, M., Thomas, A., Wornow, M., Birch-Sykes, C., Massaroli, S., Patel, A., Rabideau, C., Bengio, Y., et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36, 2024.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Orvieto, A., Smith, S. L., Gu, A., Fernando, A., Gulcehre, C., Pascanu, R., and De, S. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pp. 26670–26698. PMLR, 2023.

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pp. 28043–28078. PMLR, 2023.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

Romero, D. W., Kuzina, A., Bekkers, E. J., Tomczak, J. M., and Hoogendoorn, M. Ckconv: Continuous kernel convolution for sequential data. *arXiv preprint arXiv:2102.02611*, 2021.

Ruhe, D., Brandstetter, J., and Forré, P. Clifford group equivariant neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=n84bzMrGUD.

Ruhe, D., Gupta, J. K., De Keninck, S., Welling, M., and Brandstetter, J. Geometric clifford algebra networks. In *International Conference on Machine Learning*, pp. 29306–29337. PMLR, 2023b.

Rusch, T. K., Bronstein, M. M., and Mishra, S. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023.

Sato, K., Akiyama, M., and Sakakibara, Y. Rna secondary structure prediction using deep learning with thermodynamic integration. *Nature communications*, 12(1):941, 2021.

Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Schiff, Y., Kao, C.-H., Gokaslan, A., Dao, T., Gu, A., and Kuleshov, V. Caduceus: Bi-directional equivariant long-range dna sequence modeling. *arXiv preprint arXiv:2403.03234*, 2024.

Schütt, K., Kindermans, P.-J., Sauceda Felix, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, X., Wang, S., Ding, Y., Li, Y., Wu, W., Rong, Y., Kong, W., Huang, J., Li, S., Yang, H., et al. State space model for new-generation network alternative to transformers: A survey. *arXiv preprint arXiv:2404.09516*, 2024.

Watters, N., Zoran, D., Weber, T., Battaglia, P., Pascanu, R., and Tacchetti, A. Visual interaction networks: Learning a physics simulator from video. *Advances in neural information processing systems*, 30, 2017.

Weiler, M. and Cesa, G. General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32, 2019.

Wu, W., Qi, Z., and Fuxin, L. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 9621–9630, 2019.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Yang, J., Liu, Z., Xiao, S., Li, C., Lian, D., Agrawal, S., Singh, A., Sun, G., and Xie, X. Graphformers: Gnn-nested transformers for representation learning on textual graph. *Advances in Neural Information Processing Systems*, 34:28798–28810, 2021.

Zhang, X., Wang, L., Helwig, J., Luo, Y., Fu, C., Xie, Y., Liu, M., Lin, Y., Xu, Z., Yan, K., et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. *arXiv preprint arXiv:2307.08423*, 2023.

Zhdanov, M., Ruhe, D., Weiler, M., Lucic, A., Brandstetter, J., and Forré, P. Clifford-steerable convolutional neural networks, 2024.

Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., and Wang, X. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

A. Equivariant vector long convolution

We provide Pytorch implementation for the rotation-equivariant (without centering) vector long convolution in Code 1.

```python

class VectorLongConv(nn.Module):
    def __init__(self):
        super(VectorLongConv, self).__init__()

        # L cross-prod tensor factorized:
        l_reduced = torch.FloatTensor([[1, 0, 0],
                                       [1, 0, 0],
                                       [0, 1, 0],
                                       [0, 1, 0],
                                       [0, 0, 1],
                                       [0, 0, 1]])
        self.register_buffer("l_reduced", l_reduced, persistent=False)

        # H cross-prod tensor factorized:
        h_reduced = torch.FloatTensor([[0 , 1, 0],
                                       [0 , 0,-1],
                                       [-1, 0, 0],
                                       [0 , 0, 1],
                                       [1 , 0, 0],
                                       [0 ,-1, 0]])
        self.register_buffer("h_reduced", h_reduced, persistent=False)

        # P cross-prod tensor factorized:
        p_reduced = torch.FloatTensor([[0,0,0,1,0,1],
                                       [0,1,0,0,1,0],
                                       [1,0,1,0,0,0]])
        self.register_buffer("p_reduced", p_reduced, persistent=False)

    def forward(self, q, k):

        # q,k: (batch, sequence, dim=3)
        B, N, D = q.shape

        # batchify L,H,P reduced matrices
        l_reduced = self.l_reduced[None,None].repeat(B,N,1,1)
        h_reduced = self.h_reduced[None,None].repeat(B,N,1,1)
        p_reduced = self.p_reduced[None,None].repeat(B,N,1,1)

        # expand inputs with reduced L,H,P matricies
        q_expd = torch.matmul(l_reduced, q.unsqueeze(-1)).squeeze(-1)
        k_expd = torch.matmul(h_reduced, k.unsqueeze(-1)).squeeze(-1)

        # fft conv
        fft_q = torch.fft.rfft(q_expd, n=N, dim=1)
        fft_k = torch.fft.rfft(k_expd, n=N, dim=1)
        fft_conv_qk = torch.fft.irfft(fft_q*fft_k, n=N, dim=1)

        # reduce to vector product and normalize
        u = torch.matmul(p_reduced, fft_conv_qk.unsqueeze(-1)).squeeze(-1) / N

        return u
```

Code 1. Pytorch implementation of the equivariant vector long convolution.

B. Equivariant vector self-attention

Since SE(3)-Hyena utilizes long convolutions based on cross products, we design cross product equivariant self-attention to align the global geometric context aggregation for Hyena and transformer models. Firstly, the self-attention tensor is

built from all pairs of cross products between queries and keys. Then, the L_2 norm matrix is extracted and softmax is run row-wise on top of this matrix. The resulting softmax matrix serves as a selection mechanism to select vectors from the cross product tensor. Finally, the values are cross-multiplied with the self-attention tensor. This yields a set of processed values modulated with global self-attention information.

Cross product vector self-attention Consider sequences of N vector queries, keys, and values denoted as $\mathbf{q}, \mathbf{k}, \mathbf{v} \in \mathbb{R}^{N\times 3}$. We construct a query-key cross product tensor $\mathbf{C} \in \mathbb{R}^{(N\times N)\times 3}$ where each element $\mathbf{C}_{ij} = \mathbf{q}_i \times \mathbf{k}_j$, or using Levi-Civita notation as in Eq. 3, $\mathbf{C}_{ij}[l] = \varepsilon_{lhp}\mathbf{q}_i[h]\mathbf{k}_j[p]$. To integrate a softmax selection mechanism, as in standard self-attention, we first compute a matrix $\eta(\mathbf{C}) \in \mathbb{R}^{N\times N}$ containing the L_2 norms of all cross products, specifically $\eta(\mathbf{C})_{ij} = \|\mathbf{q}_i \times \mathbf{k}_j\|_2$. Applying softmax to $\eta(\mathbf{C})$ then determines the vector pairs to select from the cross product tensor. Lastly, the values $\mathbf{v}$ are cross-multiplied with the softmax-filtered cross product tensor. Overall, the equivariant vector self-attention reads as:

$$\mathbf{S} = \texttt{softmax}(\frac{1}{\sqrt{N}}\eta(\mathbf{C})) \odot \mathbf{C} \tag{4}$$

$$\mathbf{u}_i = \frac{1}{N}\sum_{j=1}^{N} \mathbf{S}_{ij} \times \mathbf{v}_j \tag{5}$$

where the softmax is applied row-wise, and $\odot$ stands for element-wise product. Consequently, $\mathbf{S} \in \mathbb{R}^{(N\times N)\times 3}$ represents a tensor that encapsulates a soft selection of cross products between $\mathbf{q}_i$ and $\mathbf{k}_j$. Initially, we considered using just $\texttt{softmax}(\frac{1}{\sqrt{N}}\eta(\mathbf{C}))$ as a self-attention matrix, but early experiments indicated that the method outlined in Eq. 4 yields better results. Additionally, we found that normalizing the sum by $1/N$ in Eq. 5 further improves convergence.

Since the tensor $\mathbf{C}$ is constructed using cross products, it naturally maintains equivariance to rotations of queries and keys. Furthermore, the softmax is applied to the L_2 norms of these cross products, which makes it invariant to rotations. Consequently, the self-attention tensor $\mathbf{S}$ is a product of rotation-invariant scalar and rotation-equivariant vector quantities, rendering it rotation-equivariant. The Eq. 5 further preserves rotation-equivariance due to the inherent equivariance of the cross product. Equivariance to translations can be achieved by initially centering the data (subtracting the center of mass) and then re-centering the resulting tokens.

We provide Pytorch implementation for the rotation-equivariant (without centering) vector self-attention in Code 2.

```python

class VectorSelfAttention(nn.Module):
    def __init__(self):
        super(VectorSelfAttention, self).__init__()

    def forward(self, q, k, v):

        # q,k,v: (batch, sequence, dim=3)
        B, N, D = q.shape

        # cross product matrix C: (B, N, N, 3)
        q_expd, k_expd = q.unsqueeze(2), k.unsqueeze(1)
        C = torch.cross(q_expd, k_expd, dim=-1)

        # \eta(C) matrix of norms and softmax: (B, N, N, 1)
        eta_C = C.norm(dim=-1, keepdim=True)
        sm_eta_C = nn.functional.softmax(eta_C/(N**0.5), dim=2)

        # S matrix: (B, N, N, 3)
        S = sm_eta_C*C

        # compute u = S x v: (B, N, 3)
        pre_u = torch.cross(S, v.unsqueeze(2), dim=-1)
        u = pre_u.mean(dim=2)

        return u
```

Code 2. Pytorch implementation of equivariant vector self-attention.