Geometric And Physical Quantities Improve E(3) Equivariant Message Passing

LoGaG Reading Group

Johannes Brandstetter*[1,2]
Rob Hesselink*[1]
Elise van der Pol[1]
Erik Bekkers[1]
Max Welling[1]

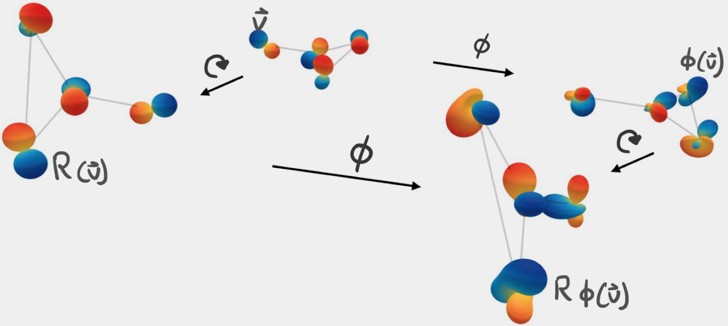
[1] University of Amsterdam
[2] Johannes Kepler University Linz

JOHANNES KEPLER UNIVERSITY LINZ

Vector-valued information

Vector valued quantities are abundant in natural sciences. How to exploit, embed, or learn geometric/physical cues?

- Extend E(3) equivariance towards vector-valued quantities, e.g. force or velocity.
- E(3) equivariance = equivariance with respect to rotations, translation, reflections, (and permutations).
- Augment message and node update networks with vector-valued quantities.



Steerable features, steerable vector spaces, steerable MLPs

Loosely speaking, steerability wrt certain group: objects transform with matrix-vector multiplication.

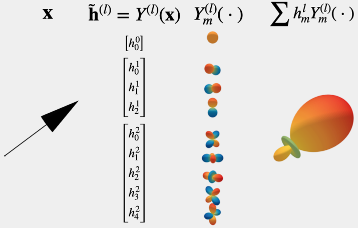
- We work in the basis spanned by spherical harmonics[1].
- Spherical harmonics embedding is equivariant w.r.t rotations (we work with subspaces on which rotations act).
- Clebsch-Gordan (CG) tensor product provides equivariant map between steerable vector spaces.

[1] Geiger et al. e3nn library `https://github.com/e3nn/e3nn`.

Steerable features, steerable vector spaces, steerable MLPs

Loosely speaking, steerability wrt certain group: objects transform with matrix-vector multiplication.

- We work in the basis spanned by spherical harmonics[1].
- Spherical harmonics embedding is equivariant w.r.t rotations (we work with subspaces on which rotations act).
- Clebsch-Gordan (CG) tensor product provides equivariant map between steerable vector spaces.

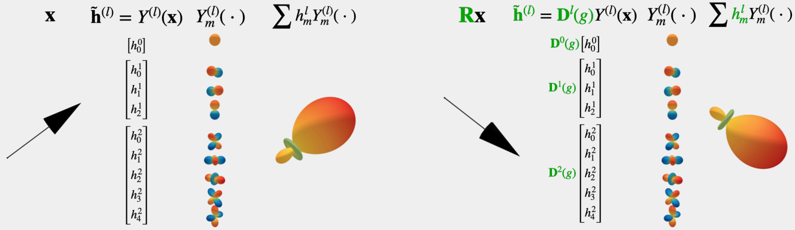


[1] Geiger et al. e3nn library `https://github.com/e3nn/e3nn`.

STEERABLE FEATURES, STEERABLE VECTOR SPACES, STEERABLE MLPS

Loosely speaking, steerability wrt certain group: objects transform with matrix-vector multiplication.

- We work in the basis spanned by spherical harmonics[1].
- Spherical harmonics embedding is equivariant w.r.t rotations (we work with subspaces on which rotations act).
- Clebsch-Gordan (CG) tensor product provides equivariant map between steerable vector spaces.



[1] Geiger et al. e3nn library `https://github.com/e3nn/e3nn`.

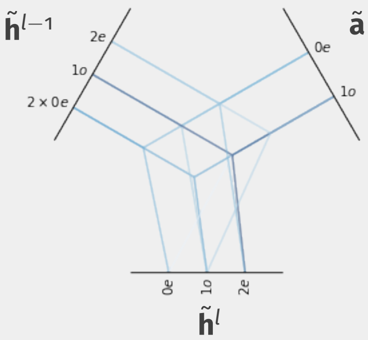
Steerable E(3) Equivariant Graph Neural Networks (SEGNNs)

Message (ϕ_m) and node update (ϕ_f) networks as CG tensor products interleaved with non-linearities:

- Steerable node vector $\tilde{\mathbf{f}}_i$ for node i, conditioned on geometric or physical cues $\tilde{\mathbf{a}}_i / \tilde{\mathbf{a}}_{ij}$.

$$\tilde{\mathbf{m}}_{ij} = \phi_m \left(\underbrace{\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_j, \|\mathbf{x}_j - \mathbf{x}_i\|^2}_{\tilde{\mathbf{h}}_{ij}}, \ \tilde{\mathbf{a}}_{ij} \right)$$

$$\tilde{\mathbf{f}}'_i = \phi_f \left(\underbrace{\tilde{\mathbf{f}}_i, \sum_{j \in \mathcal{N}(i)} \tilde{\mathbf{m}}_{ij}}_{\tilde{\mathbf{h}}_i}, \ \tilde{\mathbf{a}}_i \right)$$



Message passing of SEGNNs can be thought of as building neural networks via **non-linear (steerable) group convolutions**:

- Tensor field networks[2], Cormorant[3], or SE(3)-Transformer[4] can all be written in linear convolution form:

$$\tilde{\mathbf{f}}'_i = \sum_{j \in \mathcal{N}(i)} \underbrace{\mathbf{W}_{\tilde{\mathbf{a}}_{ij}}(\|\mathbf{x}_j - \mathbf{x}_i\|)}_{\text{CG weights}} \tilde{\mathbf{f}}_j \,, \quad \text{or} \quad \tilde{\mathbf{f}}'_i = \sum_{j \in \mathcal{N}(i)} \underbrace{\mathbf{W}_{\tilde{\mathbf{a}}_{ij}}(\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_j, \|\mathbf{x}_j - \mathbf{x}_i\|)}_{\text{CG weights}} \tilde{\mathbf{f}}_j \,.$$

- SEGNN messages are obtained highly non-linear:

$$\tilde{\mathbf{m}}_{ij} = \widetilde{\mathrm{MLP}}_{\tilde{\mathbf{a}}_{ij}}(\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_j, \|\mathbf{x}_j - \mathbf{x}_i\|^2) = \sigma(\mathbf{W}^{(n)}_{\tilde{\mathbf{a}}_{ij}}(\ldots(\sigma(\mathbf{W}^{(1)}_{\tilde{\mathbf{a}}_{ij}} \tilde{\mathbf{h}}_i)))) \,.$$

[2] Thomas et al. Rotation-and translation-equivariant neural networks for 3d point clouds.

[3] Anderson et al. Cormorant: Covariant molecular neuralnetworks.

[4] Fuchs et al. Se (3)-transformers: 3d roto-translation equivariant attention networks.

We work with gated non-linearities:

- Direct sum of two sets of irreps for $\mathbf{h}^l$ ($l > 0$): (i) scalar irreps passed through activation functions (gating), (ii) higher order irreps multiplied by gating

Framing message passing as non-linear convolution allows us to see the node update as **new equivariant activation function**:

$$\tilde{\mathbf{f}}'_i = \phi_f \left(\underbrace{\tilde{\mathbf{f}}_i, \sum_{j \in \mathcal{N}(i)} \tilde{\mathbf{m}}_{ij},}_{\tilde{\mathbf{h}}_i} \;\; \tilde{\mathbf{a}}_i \right) .$$

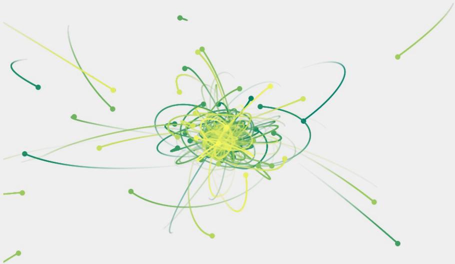
Activation function as non-linear MLPs, which are applied node-wise.

Performance and applicability

SEGNNs give you an advantage when (i) there is physical and geometrical information available, and (ii) full connectivity of the graphs is computationally not tractable.

- Enrich (steer) node updates via velocity, force, momentum, acceleration, spin, angular momentum ...
- Enrich (steer) messages via relative position, relative forces, dipole moments, ...

Method	MSE
SE(3)-Tr.	.0244
TFN	.0155
NMP	.0107
Radial Field	.0104
EGNN	.0070
SE_{linear}	.0116
$SE_{non\text{-}linear}$	.0060
$SEGNN_{G}$	.0056
$SEGNN_{G+P}$	.0043



ICLR Poster: 6225
Paper: Geometric and Physical Quantities Improve E(3) Equivariant Message Passing

arXiv:2110.02905

Code:

https://github.com/RobDHess/Steerable-E3-GNN

Code: https://github.com/RobDHess/Steerable-E3-GNN

$$\tilde{\mathbf{m}}_{ij} = \phi_m\left(\underbrace{\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_j, \|\mathbf{x}_j - \mathbf{x}_i\|^2}_{\tilde{\mathbf{h}}_{ij}},\ \tilde{\mathbf{a}}_{ij}\right)$$

$$\tilde{\mathbf{f}}'_i = \phi_f\left(\underbrace{\tilde{\mathbf{f}}_i, \sum_{j\in\mathcal{N}(i)} \tilde{\mathbf{m}}_{ij}}_{\tilde{\mathbf{h}}_i},\ \tilde{\mathbf{a}}_i\right)$$

Require: $\tilde{\mathbf{f}}_i, \mathbf{x}_{ij}, \mathbf{v}_i^1, \mathbf{v}_i^2$ ▷ Steerable nodes $\tilde{\mathbf{f}}_i$, relative position vector $\mathbf{x}_{ij}$ between node $\tilde{\mathbf{f}}_i$ and node $\tilde{\mathbf{f}}_j$, geometric or physical quantities $\mathbf{v}_i^1, \mathbf{v}_i^2$ such as velocity, acceleration, spin, or force.

function O3_Tensor_Product(input1, input2)
  output ← CGTensorProduct(input1, input2) ▷ Apply CG tensor product following Eq. (6)
  output ← output + bias ▷ Add bias to zero order irreps
  return output
end function
function O3_Tensor_Product_Swish_Gate(input1, input2)
  output $\oplus\, \mathbf{g}_i$ ← O3_Tensor_Product(input1, input2) ▷ Output plus scalar irreps $\mathbf{g}_i$
  $\text{output}_{\text{gated}}$ ← Gate(output, Swish($\mathbf{g}_i$)) ▷ Transform output via gated non-linearities
  return output
end function
$\tilde{\mathbf{a}}_{ij} \leftarrow$ SphericalHarmonicEmbedding($\mathbf{x}_{ij}$) ▷ Spherical harmonic embedding of $\mathbf{x}_{ij}$ (Eq. (4))
$\tilde{\mathbf{v}}_i^1 \leftarrow$ SphericalHarmonicEmbedding($\mathbf{v}_i^1$) ▷ Spherical harmonic embedding of $\mathbf{v}_i^1$ (Eq. (4))
$\tilde{\mathbf{v}}_i^2 \leftarrow$ SphericalHarmonicEmbedding($\mathbf{v}_i^2$) ▷ Spherical harmonic embedding of $\mathbf{v}_i^2$ (Eq. (4))
$\tilde{\mathbf{a}}_i \leftarrow \sum_j \tilde{\mathbf{a}}_{ij} + \tilde{\mathbf{v}}_i^1 + \tilde{\mathbf{v}}_i^2$ ▷ Node attributes

$\tilde{\mathbf{h}}_{ij} \leftarrow \tilde{\mathbf{f}}_i \oplus \tilde{\mathbf{f}}_j \oplus \|\mathbf{x}_{ij}\|^2$ ▷ Concatenate input for messages between $\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_j$
$\tilde{\mathbf{m}}_{ij} \leftarrow$ O3_Tensor_Product_Swish_Gate($\tilde{\mathbf{h}}_{ij}, \tilde{\mathbf{a}}_{ij}$) ▷ First non-linear message layer
$\tilde{\mathbf{m}}_{ij} \leftarrow$ O3_Tensor_Product_Swish_Gate($\tilde{\mathbf{m}}_{ij}, \tilde{\mathbf{a}}_{ij}$) ▷ Second non-linear message layer
$\tilde{\mathbf{m}}_i \leftarrow \sum_j \tilde{\mathbf{m}}_{ij}$ ▷ Aggregate messages $\tilde{\mathbf{m}}_{ij}$

$\tilde{\mathbf{f}}'_i \leftarrow$ O3_Tensor_Product_Swish_Gate($\tilde{\mathbf{f}}_i \oplus \tilde{\mathbf{m}}_i, \tilde{\mathbf{a}}_i$) ▷ First non-linear node update layer
$\tilde{\mathbf{f}}'_i \leftarrow \tilde{\mathbf{f}}_i +$ O3_Tensor_Product($\tilde{\mathbf{f}}'_i, \tilde{\mathbf{a}}_i$) ▷ Second linear node update layer

			Task Units	α bohr^3	$\Delta\varepsilon$ meV	ε_{HOMO} meV	ε_{LUMO} meV	μ D	C_ν cal/mol K
non-linear		no geometry	NMP	.092	69	43	38	.030	.040
	regular	$\mathbb{R}^3$	SchNet *	.235	63	41	34	.033	.033
pseudo-linear	steerable	$\mathbb{R}^3$	Cormorant	.085	61	34	38	.038	.026
	steerable	$SE(3)$	L1Net	.088	68	46	35	.043	.031
	regular	G	LieConv	.084	49	30	25	.032	.038
	steerable	$SE(3)$	TFN	.223	58	40	38	.064	.101
pseudo-linear	steerable	$SE(3)$	SE(3)-Tr.	.142	53	35	33	.051	.054
non-linear	regular	$\mathbb{R}^3 \times S^2 \times \mathbb{R}^+$	DimeNet++ *	**.043**	**32**	24	19	.029	.023
non-linear	regular	$\mathbb{R}^3 \times S^2 \times \mathbb{R}^+$	SphereNet *	.046	**32**	**23**	**18**	.026	**.021**
non-linear	reguleerable?	$SE(3)$	PaiNN *	.045	45	27	20	**.012**	.024
non-linear	regular	$\mathbb{R}^3$	EGNN	.071	48	29	25	.029	.031
non-linear	steerable	$SE(3)$	SEGNN (Ours)	.060	42	24	21	.023	.031

Table 2: Comparison on QM9.

- Group convolutions, one way or the other[5]:
  - "Any equivariant linear layer between feat maps on homogeneous spaces is a group conv
  - If $X \equiv G/H$: kernel has symmetry constraints (SchNet, EGNN, ...)
  - Idea of **non-linear convolution** discussed in Section 3.
- Recent work by Cesa, Lang & Weiler [6]: comprehensive theory and code framework for general steerable CNNs.

[5] See e.g. Thm. 1 in: Bekkers, E. J. (2019). B-Spline CNNs on Lie groups. In ICLR.

[6] Cesa, G, Lang, L., Weiler, M. (2022). A Program to Build E(N)-Equivariant Steerable CNNs. In ICLR.

Related Work: SchNet[8] linear $\mathbb{R}^3$ convolution

Given the feature representations of n objects $X^l = (\mathbf{x}_1^l, \ldots, \mathbf{x}_n^l)$ with $\mathbf{x}_i^l \in \mathbb{R}^F$ at locations $R = (\mathbf{r}_1, \ldots, \mathbf{r}_n)$ with $\mathbf{r}_i \in \mathbb{R}^D$, the continuous-filter convolutional layer l requires a filter-generating function

$$W^l : \mathbb{R}^D \to \mathbb{R}^F,$$

that maps from a position to the corresponding filter values. This constitutes a generalization of a filter tensor in discrete convolutional layers. As in dynamic filter networks [34], this filter-generating function is modeled with a neural network. While dynamic filter networks generate weights restricted to a grid structure, our approach generalizes this to arbitrary position and number of objects. The output $\mathbf{x}_i^{l+1}$ for the convolutional layer at position $\mathbf{r}_i$ is then given by

Filter/weights conditioned on $\|\mathbf{r}_i - \mathbf{r}_j\|$

$$\mathbf{x}_i^{l+1} = (X^l * W^l)_i = \sum_j \mathbf{x}_j^l \circ W^l(\mathbf{r}_i - \mathbf{r}_j), \qquad (2)$$

Separable convolution (“gating”)

where "$\circ$" represents the element-wise multiplication. We apply these convolutions feature-wise for computational efficiency [35]. The interactions between feature maps are handled by separate object-wise or, specifically, atom-wise layers in SchNet.

- *Linear SE*(3) equivariant convolutions on $\mathbb{R}^3$.
- Depth/channel-wise seperable[7]

[7] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

[8] Schütt, K., Kindermans, P. J., Sauceda Felix, H. E., Chmiela, S., Tkatchenko, A., & Müller, K. R. (2017). Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. NeurIPS, 30.

Related Work: EGNN[9] non-linear $\mathbb{R}^3$ conv

$$\mathbf{m}_{ij} = \phi_e\left(\mathbf{h}_i^l, \mathbf{h}_j^l, \left\|\mathbf{x}_i^l - \mathbf{x}_j^l\right\|^2, a_{ij}\right) \tag{3}$$

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + C\sum_{j\neq i}\left(\mathbf{x}_i^l - \mathbf{x}_j^l\right)\phi_x\left(\mathbf{m}_{ij}\right) \tag{4}$$

$$\mathbf{m}_i = \sum_{j\neq i}\mathbf{m}_{ij} \tag{5}$$

$$\mathbf{h}_i^{l+1} = \phi_h\left(\mathbf{h}_i^l, \mathbf{m}_i\right) \tag{6}$$

- Non-linear $SE(3)$ equivariant "convolutions" on $\mathbb{R}^3$.
  - Messages "**conditioned**" on $\|\mathbf{x}_i - \mathbf{x}_j\|$
- We extend this to the steerable case to obtain:
  - *Non-linear* $SE(3)$ equivariant "convolutions" on $\mathbb{R}^3 \times SO(3)$.

[9] Satorras, V. G., Hoogeboom, E., & Welling, M. (2021, July). E (n) equivariant graph neural networks. In International Conference on Machine Learning (pp. 9323-9332). PMLR.

4.1.3 Layer definition

A given input inhabits one representation, a filter inhabits and at possibly many rotation orders. We can put everything together into our pointwise convolution layer definition:

we restrict them to the following form:

$$F_{cm}^{(l_f,l_i)}(\vec{r}) = R_c^{(l_f,l_i)}(r) Y_m^{(l_f)}(\hat{r}) \quad (2)$$

$$\mathcal{L}_{acm_o}^{(l_o)}\left(\vec{r}_a, V_{acm_i}^{(l_i)}\right) := \sum_{m_f, m_i} C_{(l_f,m_f)(l_i,m_i)}^{(l_o,m_o)} \sum_{b \in S} F_{cm_f}^{(l_f,l_i)}(\vec{r}_{ab}) V_{bcm_i}^{(l_i)} \quad (3)$$

Seperable convolution ("gating")

(where $\vec{r}_{ab} := \vec{r}_a - \vec{r}_b$ and the subscripts i, f, and o denote the representations of the input, filter, and output, respectively). A point convolution of an l_f filter on an l_i input yields outputs at $2\min(l_i, l_f)+1$ different rotation orders l_o (one for each integer between $|l_i - l_f|$ and $(l_i + l_f)$, inclusive), though in designing a particular network, we may choose not to calculate or use some of those outputs.

Steerable group convolution of the form $\sum_{b \in S} \mathbf{W}_{\tilde{a}_{ij}}(r)\tilde{V}(\mathbf{x}_j)$, using spherical harmonics (SH) and the CG tensor product, here with with $\tilde{a}_{ij} = Y(\hat{r})$ the SH embedding of relative positions. See section 3.

- *Linear $SE(3)$ equivariant* "convolutions" on $SE(3)$.

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

[11] Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., ... & Kozinsky, B. (2021). Se (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. arXiv preprint arXiv:2101.03164.

Related Work: Cormorant[13]

Steerable group convolution of the form $\sum_{b \in S} \mathbf{W}_{\tilde{a}_{ij}}(r, F_i, F_j)\tilde{F}(\mathbf{x}_j)$

The actual form of the vertex activations captures "one-body interactions" propagating information from the previous layer related to the *same* atom and (indirectly, via the edge activations) "two-body interactions" capturing interactions between *pairs* of atoms:

$$F_i^{s-1} = \Big[\underbrace{F_i^s \oplus \big(F_i^{s-1} \otimes_{\mathrm{cg}} F_i^{s-1}\big)}_{\text{one-body part}} \oplus \underbrace{\Big(\sum_j G_{i,j}^s \otimes_{\mathrm{cg}} F_j^{s-1}\Big)}_{\text{two-body part}} \Big] \cdot W_{s,\ell}^{\text{vertex}}. \tag{8}$$

Here $G_{i,j}^s$ are SO(3)–vectors arising from the edge network. Specifically, $G_{i,j}^{s,\ell} = g_{i,j}^{s,\ell}\, Y^\ell(\widehat{\boldsymbol{r}}_{i,j})$, where $Y^\ell(\widehat{\boldsymbol{r}}_{i,j})$ are the spherical harmonic vectors capturing the relative position of atoms i and j. The edge activations, in turn, are defined

$$g_{i,j}^{s,\ell} = \mu^s(r_{i,j}) \left[\left(g_{i,j}^{s-1,\ell} \oplus \left(F_i^{s-1} \cdot F_j^{s-1}\right) \oplus \eta^{s,\ell}(r_{i,j}) \right) W_{s,\ell}^{\text{edge}} \right] \tag{9}$$

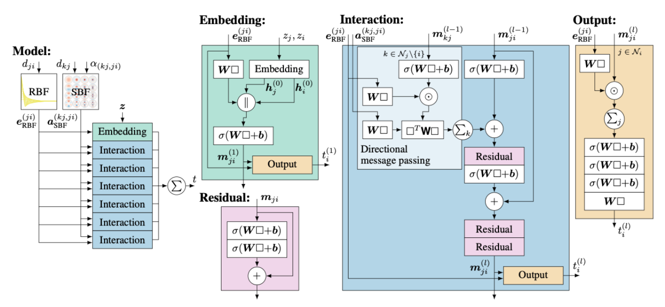
where we made the $\ell = 0, 1, \ldots, L$ irrep index explicit. As before, in these formulae, $\oplus$ denotes concatenation over the channel index c, $\eta_c^{s,\ell}(r_{i,j})$ are learnable radial functions, and $\mu_c^s(r_{i,j})$ are learnable cutoff functions limiting the influence of atoms that are farther away from atom i. The learnable parameters of the network are the $\{W^{\text{vertex}}\}$ and $\{W^{\text{edge}}\}$ weight matrices.

Fully separable (learnable channel mixing outside aggregation)

- *Pseudo-Linear* $SE(3)$ equivariant "convolutions" on $SE(3)$.
- See also $SE(3)$ transformers[12] with learnable "attention"

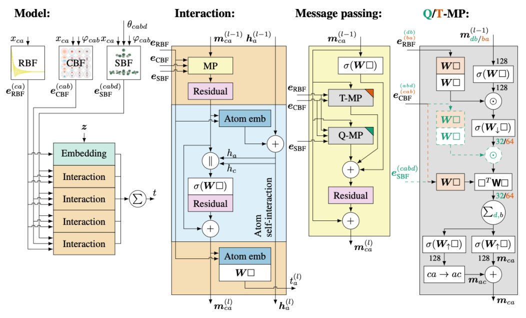
[12] Fuchs, F., Worrall, D., Fischer, V., & Welling, M. (2020). Se (3)-transformers: 3d roto-translation equivariant attention networks. NeurIPS, 33, 1970-1981.

[13] Anderson, B., Hy, T. S., & Kondor, R. (2019). Cormorant: Covariant molecular neural networks. Advances in neural information processing systems, 32.



- *Non-Linear $SE(3)$* equivariant "convolutions" on ??.
  - Messages conditioned on *invariants* s.a. distances and angles.

[14]Klicpera, J., Gross, J., & Günnemann, S. (2019, September). Directional Message Passing for Molecular Graphs. In International Conference on Learning Representations.



- *Non-Linear* $SE(3)$ equivariant "convolutions" on $\mathbb{R}^3 \times S^2$.
  - Messages conditioned on *invariants* s.a. distances and angles.

[15] Klicpera, J., Becker, F., & Günnemann, S. (2021, May). GemNet: Universal Directional Graph Neural Networks for Molecules. In NeurIPS.

4 From spherical representations to directional message passing

Eq. (3) only defines the relationship for a fixed direction, while models commonly use different directional meshes for the input and output. To incorporate this we add a convolution with a learned filter F_2, which can only improve the model's expressiveness. Since the input and output are spherical functions, the used filter F_2 has to be *zonal*, i.e. it can only depend on one angle. This can be expressed as [17]

Group convolution!

$$\tilde{\boldsymbol{H}}_a^{\text{dir}}(\boldsymbol{X}, \boldsymbol{H})(\hat{\boldsymbol{r}}_o) = \theta \boldsymbol{H}_a(\hat{\boldsymbol{r}}_o) + \int_{\text{SO}(3)} \sum_{b \in \mathcal{N}_a} F_{\text{sphere}}(\boldsymbol{x}_{ba}, \boldsymbol{R}\hat{\boldsymbol{n}}) \sum_{i \in \mathcal{R}_b} \boldsymbol{H}_{bi} \delta(\boldsymbol{R}\hat{\boldsymbol{n}} - \hat{\boldsymbol{r}}_i) F_2(\boldsymbol{R}^{-1}\hat{\boldsymbol{r}}_o) \, \mathrm{d}\boldsymbol{R}$$

$$= \theta \boldsymbol{H}_a(\hat{\boldsymbol{r}}_o) + \sum_{b \in \mathcal{N}_a} \sum_{i \in \mathcal{R}_b} F_{\text{sphere}}(\boldsymbol{x}_{ba}, \hat{\boldsymbol{r}}_i) \boldsymbol{H}_{bi} F_2(\measuredangle \hat{\boldsymbol{r}}_o \hat{\boldsymbol{r}}_i), \quad (6)$$

+ Sparse (sum of dirac-δ) signals on $\mathbb{R}^3 \times S^2$

where $\mathcal{R}_b$ denotes the directional mesh of atom b with mesh directions denoted by $\hat{\boldsymbol{r}}_i$, and $\hat{\boldsymbol{r}}_o$ specifies the output direction. The integral vanishes due to the Dirac delta δ.

= **Message passing between the edges of an $\mathbb{R}^3$ graph**

- *Non-Linear SE*(3) equivariant "convolution" on $\mathbb{R}^3 \times S^2$.
- Eq. (6) is a regular *linear* group conv evaluated at a sparse grid of directions $\subset S^2$ at each node location $\in \mathbb{R}^3$.
- They adjust to non-linear message passing!

[16] Klicpera, J., Becker, F., & Günnemann, S. (2021, May). GemNet: Universal Directional Graph Neural Networks for Molecules. In NeurIPS. non-linear regular $E(3)$ conv

Related Work: Conclusion

- Related works can all be thought of as G-convs of some kind
- $SE(3)$ group convolutions beat $\mathbb{R}^3$ convolutions (no isotropy constraints)
- Non-lin. equivariant layers beat lin. equivariant layers (G-convs)
- *Our method* combines best of both worlds!
- *Our method* conveniently handles geometric/physical quantities and shows how it leads to improved performance!