Geometric Algebra Representations for Deep Learning

AGACSE 2024

David Ruhe - August 27, 2024

About Me

- PhD-student at AMLab (University of Amsterdam)
  - AI4Science
  - Generative Models
  - Time-Series
  - Geometric Deep Learning

AMLAB
Amsterdam
Machine Learning Lab

Ph.D. Student at the University of Amsterdam

Overview

- (Quick) Overview of the field.
- Machine learning and neural networks.
- Clifford Group Equivariant Neural Networks
- Subsequent Works.

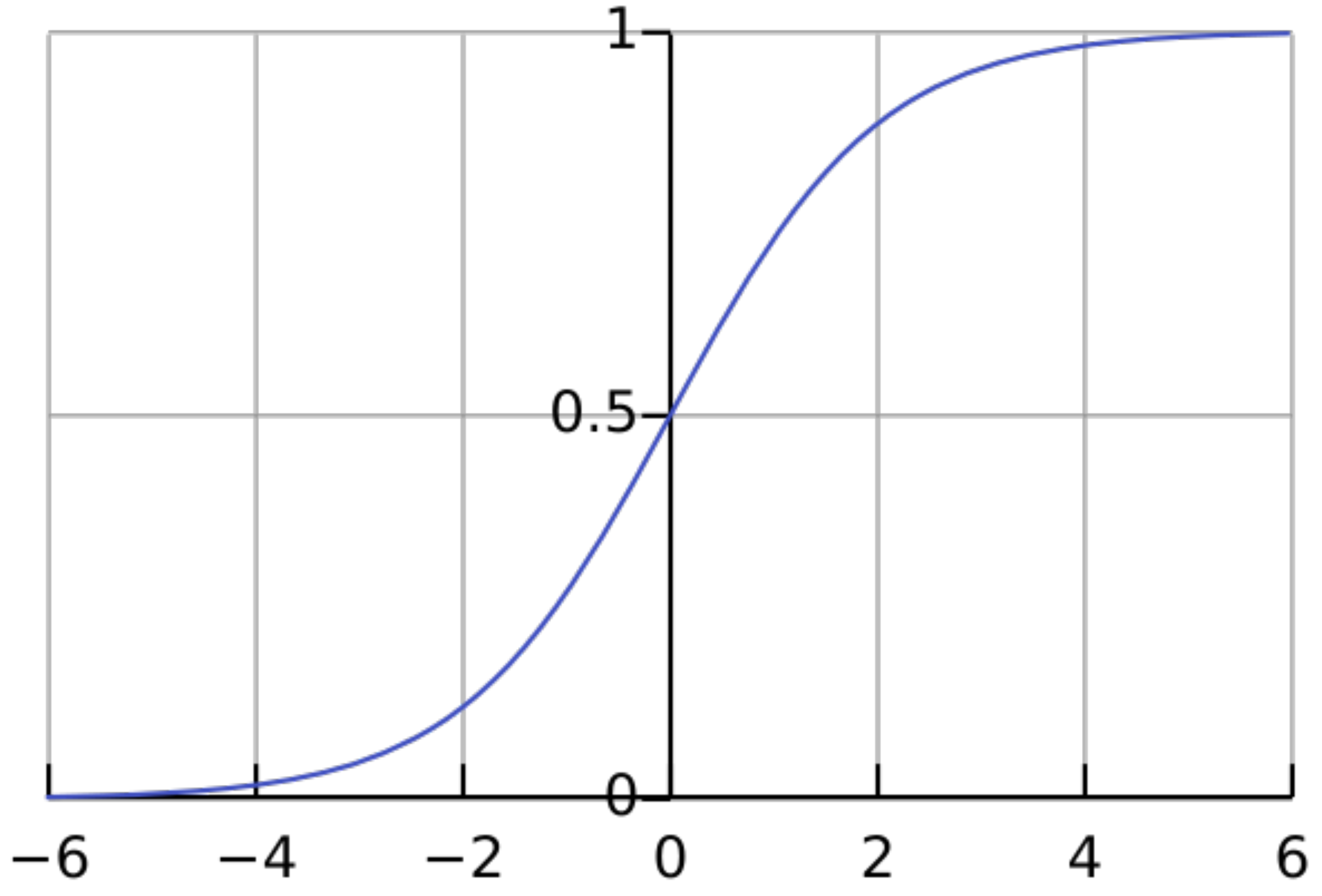
GA + NN Overview

- Early works from 90s, 00s.
- Brandstetter et al., 2022: Clifford Neural Layers for PDE Modeling (ICML)
  - Use the CA to encode and transform geometric quantities (vectors, bivectors).
  - Multivector weights.
  - Applications in fluid mechanics.
- Ruhe et al., 2023: Geometric Clifford algebra Networks
  - Based on *rotational layer.*
  - Use PGA to represent points, planes, etc.
  - Parameterized motors for dynamical systems.
- Clifford Group Equivariant Neural Networks…

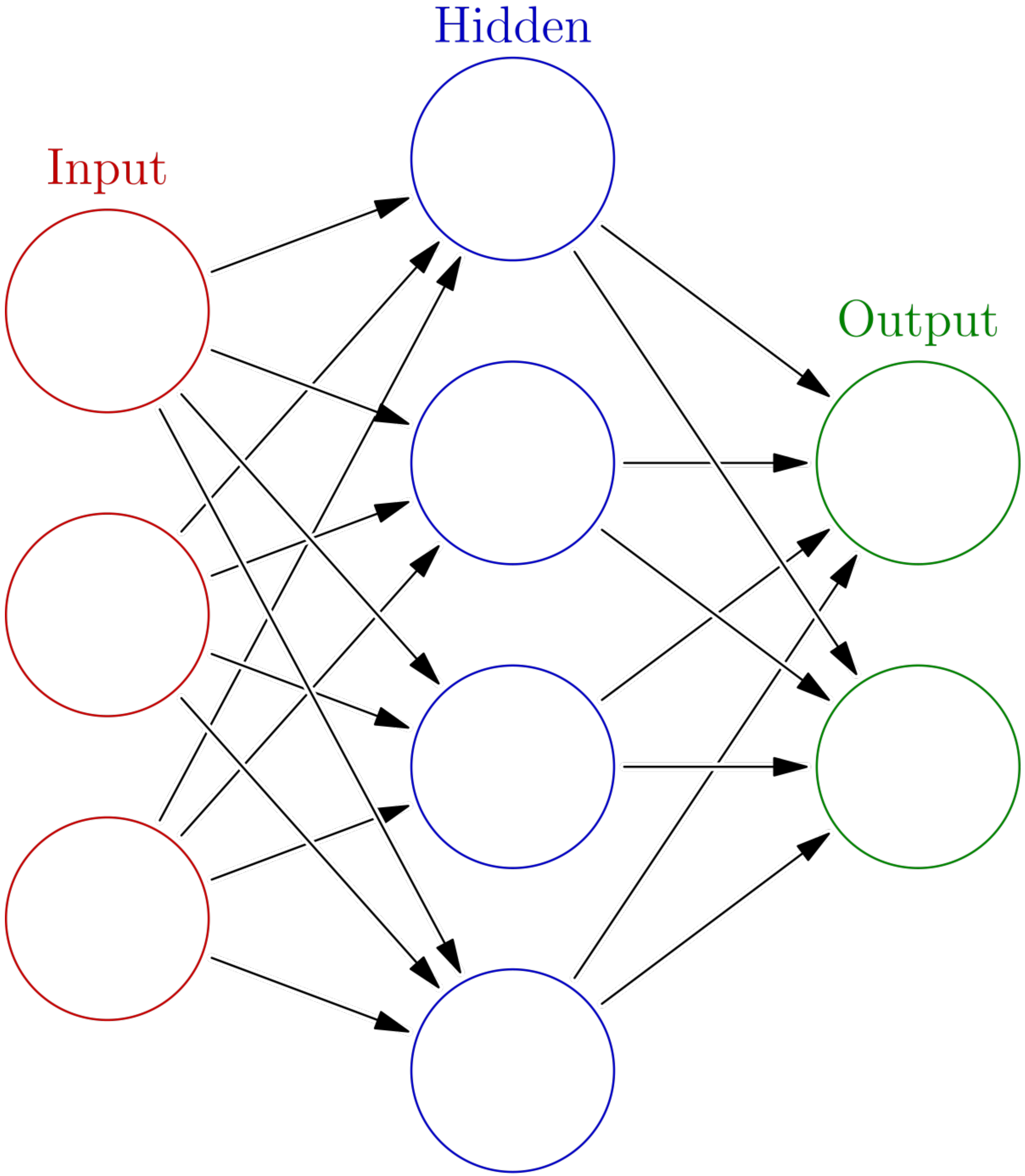
Machine Learning and Neural Networks

- Model $\phi : X \to Y$ that takes an input and outputs a *prediction*.
- Loss function $L : Y \times Y \to \mathbb{R}$ that *measures* how well the prediction was.
- Various optimization schemes to minimize L given a dataset
  - $\mathscr{D} := \{x_i, y_i\}_{i=1}^{N}$

Machine Learning and Neural Networks



- $H_l := \mathbb{R}^{d_l}, H_1 := X, H_L := Y$
- A neural network $\phi : X \to Y$ is a composition of layers with
- $\phi_l : H_l \to H_{l+1} \quad h_l \mapsto \phi_l(h_l) := \sigma(W_l h_l + b_l)$
  - $W_l \in \mathbb{R}^{d_{l+1} \times d_l}$ $b_l \in \mathbb{R}^{d_{l+1}}$ and σ is an element-wise nonlinearity.
- $\phi := \phi_{L-1} \circ \cdots \circ \phi_1$
- Parameters $\theta := \{W_l, b_l\}_{l=1}^{L-1}$ are typically refined using *gradient descent* or its variants.


Hidden
Input
Output

The Clifford Algebra

Why Deep Learning?

- Some indications CA data representations + CA weights yields more efficient learning + generalization properties.
  - Similar to complex neural networks.
- Can represent certain physics quantities through e.g. bivectors.
- Equivariance w.r.t. several groups in several dimensions (O(3), SO(3), O(2), O(1, 3), E(3), etc.
  - Translations (PGA), conformal group.
- Equivariant **multiplicative** operation (geometric product).
  - No need for spherical harmonics, CG coefficients, etc. Space is bounded.

Clifford Group Equivariant Neural Networks

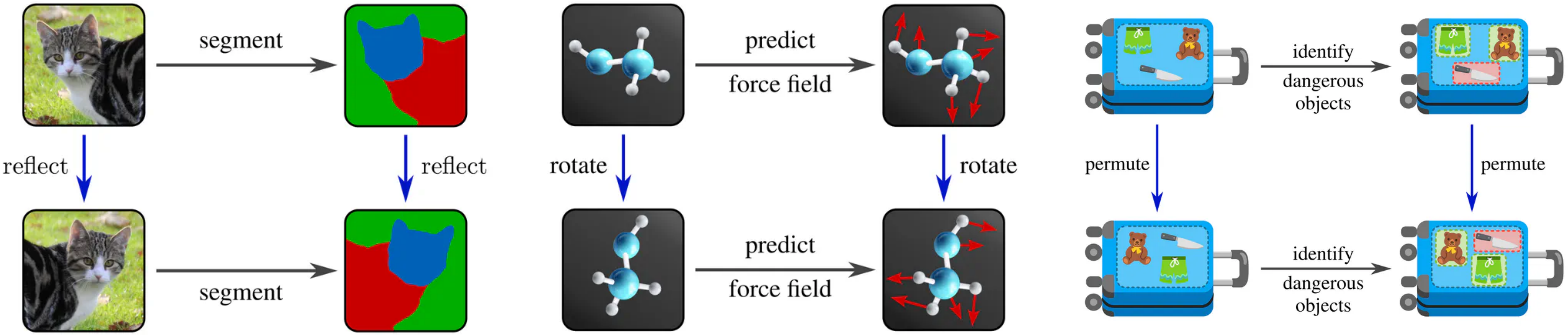
David Ruhe
University of Amsterdam
AI4Science Lab, AMLab, API
david.ruhe@gmail.com

Johannes Brandstetter
Microsoft Research
AI4Science
johannesb@microsoft.com

Patrick Forré
University of Amsterdam
AI4Science Lab, AMLab
p.d.forre@uva.nl

Introduction

Equivariant Neural Networks



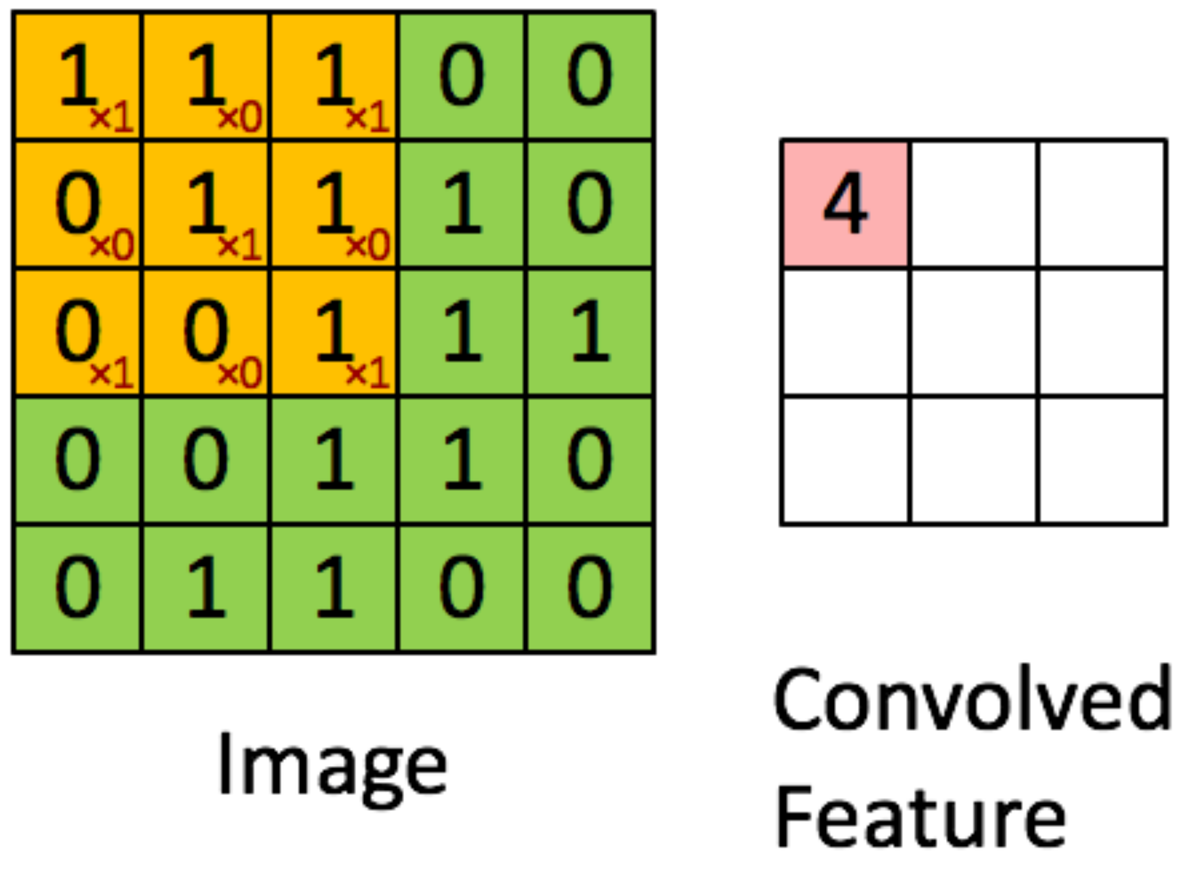
- $w \in G : \rho(w)\phi = \phi\rho(w)$
- Group equivariance stimulates *robust* and *reliable* results.

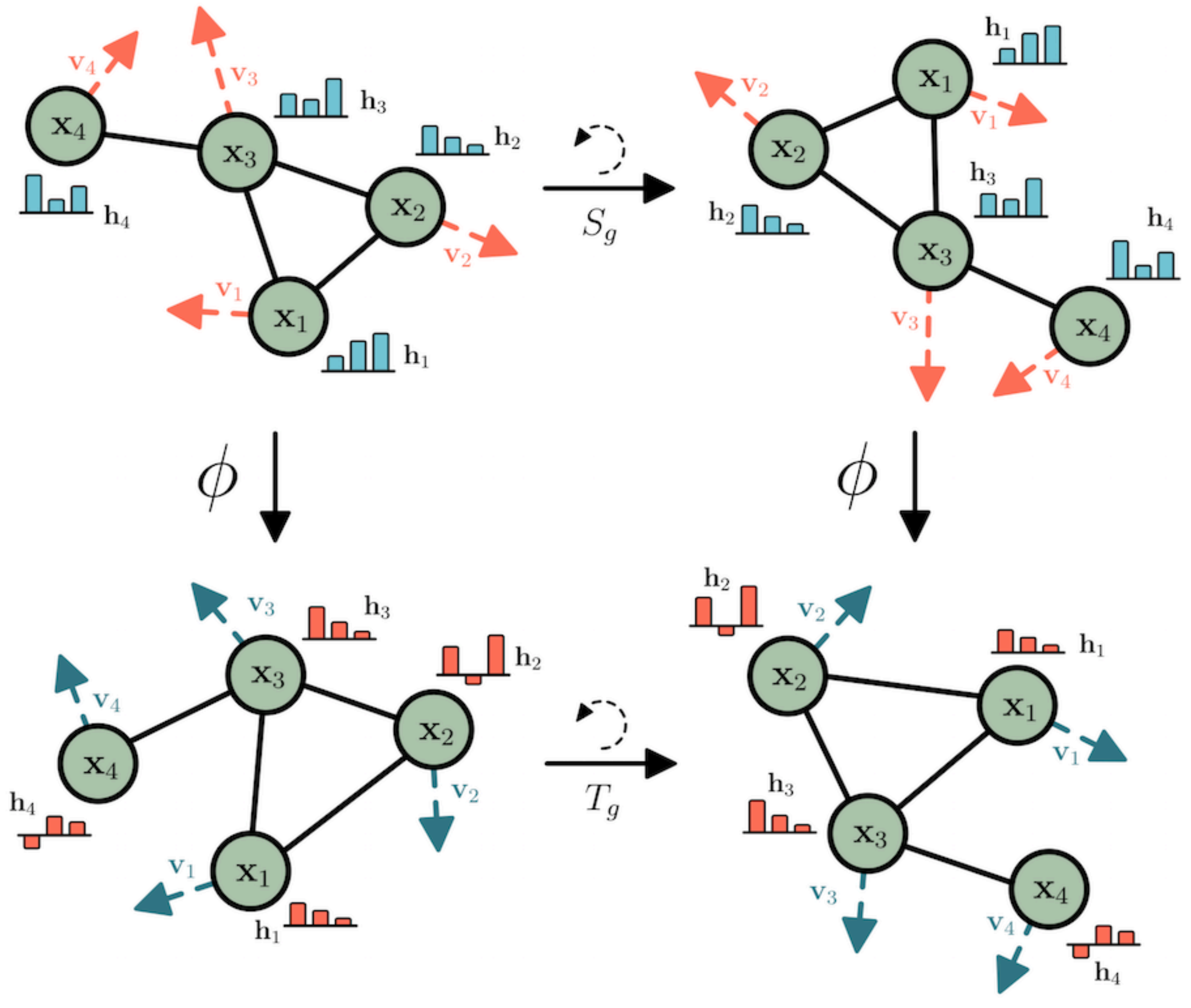
- Images by Maurice Weiler

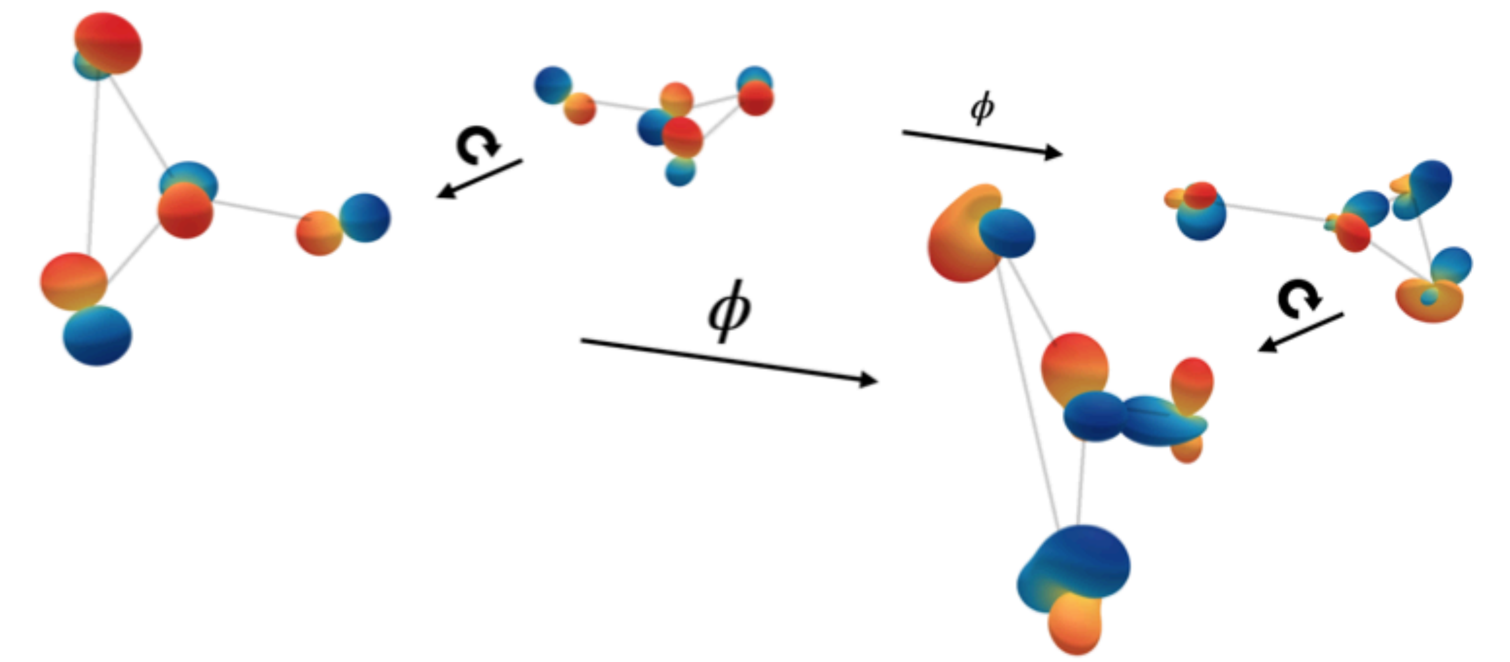
Introduction

Equivariant Neural Networks: Categorization

- Group convolutions (LieConv, B-spline CNNs).
  - Integral over a group - computationally intensive.
- Scalarization methods (EGNN, GVP, VN).
  - Operate almost exclusively with invariant (scalar) features.
  - Restricted expressivity.
- *E(3)-NN* based methods (TFN, SEGNN).
  - Tensor products of Wigner-D representations decomposed into irreps using Clebsch-Gordan coefficients.
  - Operate on spherical harmonics basis.
  - Not trivially extended to other dimensions or groups than O(3).

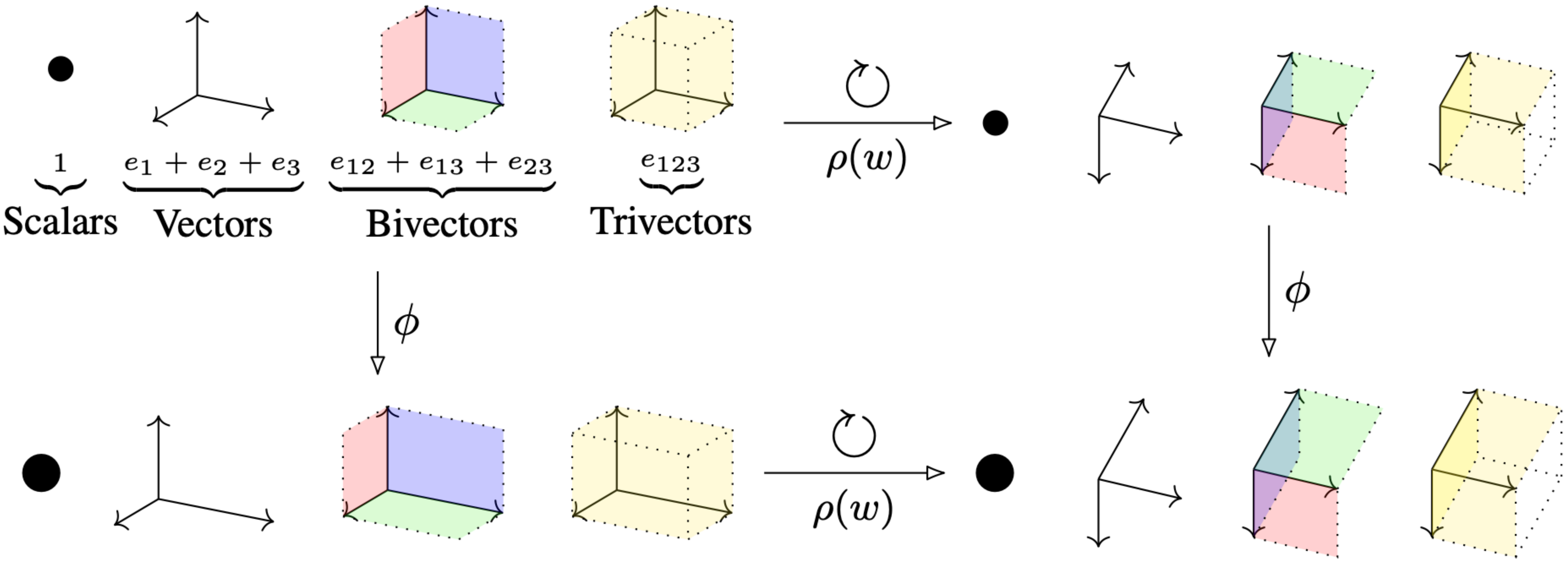






Introduction

Clifford Group Equivariant Networks



Theoretical Results

- The Clifford subspaces are not basis-dependent.
  - Even in the degenerate case.

- Clifford Group:
  - $\Gamma(V,q) := \left\{ w \in \mathrm{Cl}^{\times}(V,q) \cap \left(\mathrm{Cl}^{[0]}(V,q) \cup \mathrm{Cl}^{[1]}(V,q)\right) \mid \forall v \in V, \rho(w)(v) \in V \right\}$
  - Quotient is isomorphic to $O(V,q)$ in general?

Theoretical Results

The Clifford Group

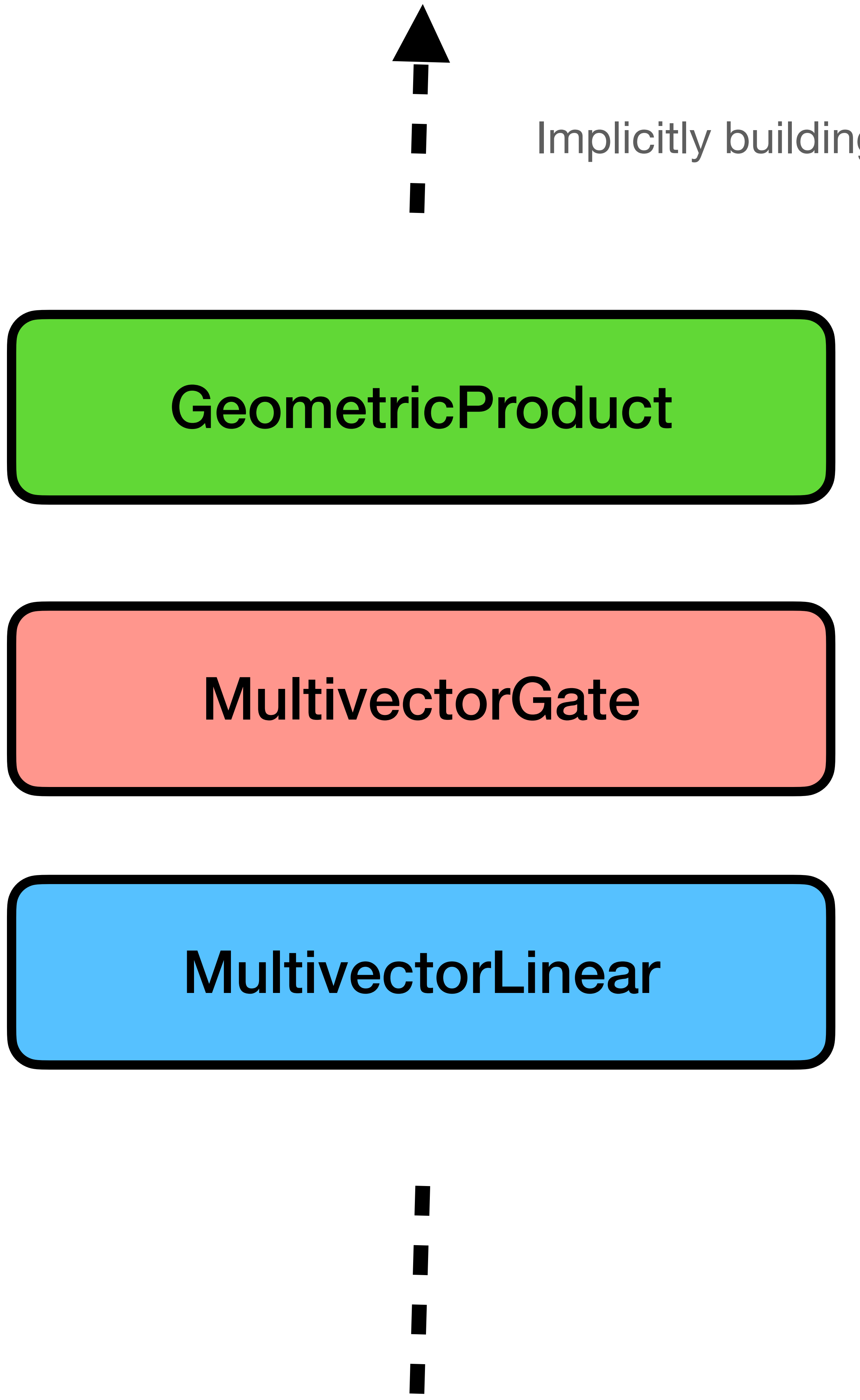
$O(V, q)$ multivector representation.

- $w \in \Gamma\left(V, q\right) \subseteq \mathrm{Cl}\left(V, q\right)$
- $\rho(w)$ satisfies:
  1. $\langle(\rho(w)(x_1), \rho(w)(x_2)\rangle = \langle x_1, x_2\rangle$
  2. Additivity: $\rho(w)(x_1 + x_2) = \rho(w)(x_1) + \rho(w)(x_2)$
  3. Multiplicativity: $\rho(w)(x_1 x_2) = \rho(w)(x_1)\rho(w)(x_2)$
  4. Commutes with scalars: $\rho(w)(\alpha \cdot x) = \alpha \cdot \rho(w)(x)$

All geometric product polynomials are $\Gamma(V, q)$ equivariant.

Network Architectures

Equivariant Layers…



Implicitly building a big polynomial…

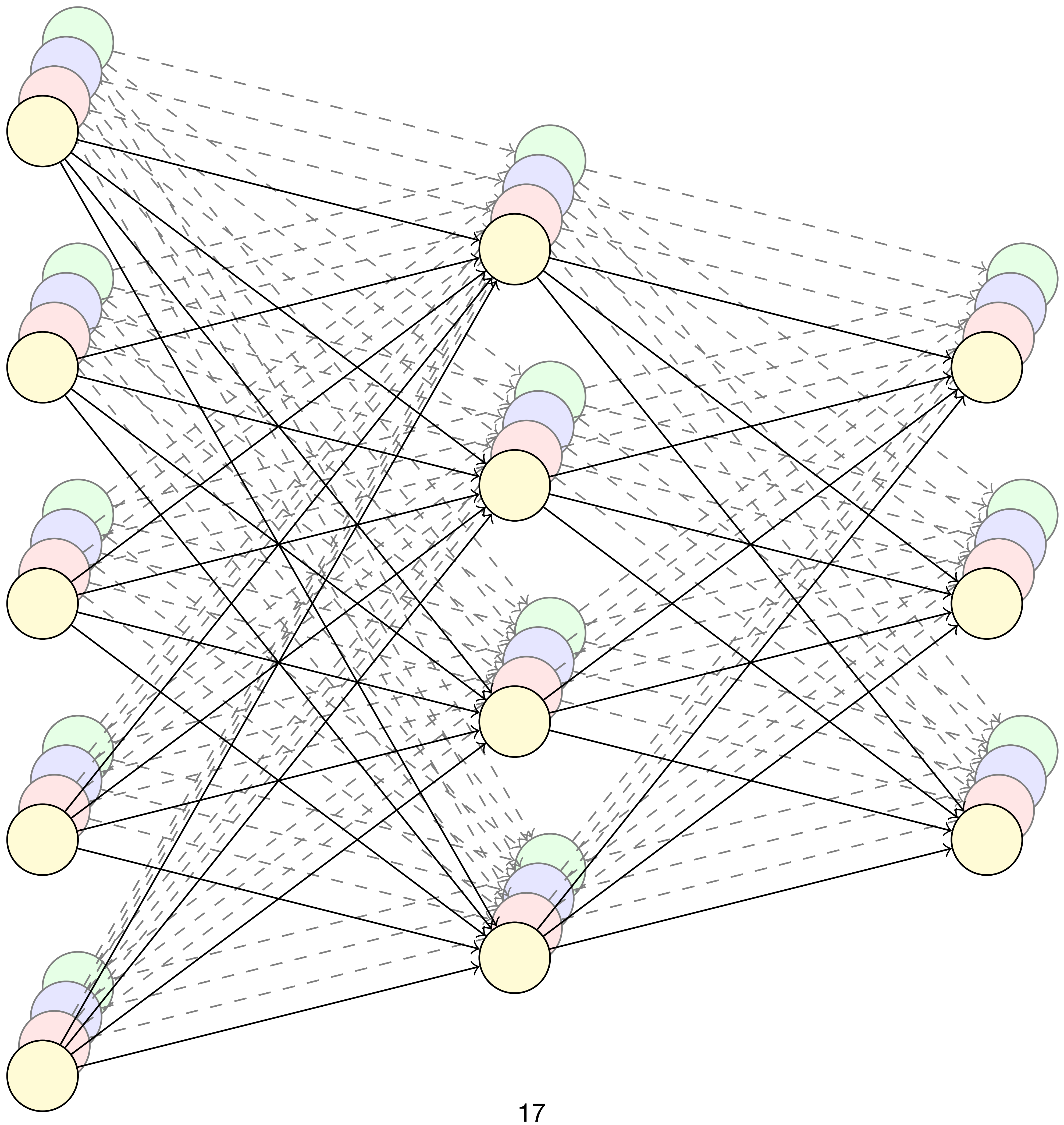
Methodology

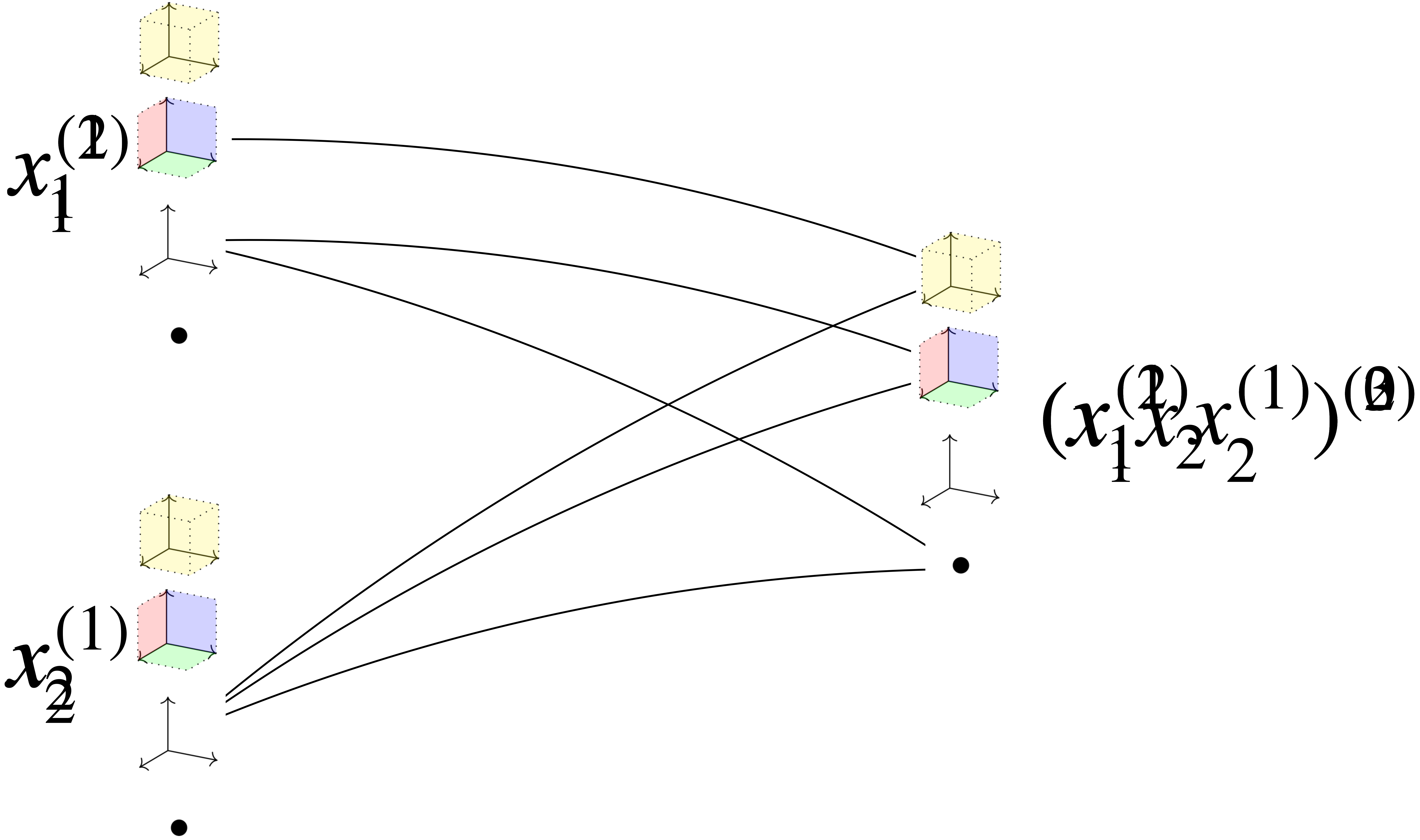
Linear Layers

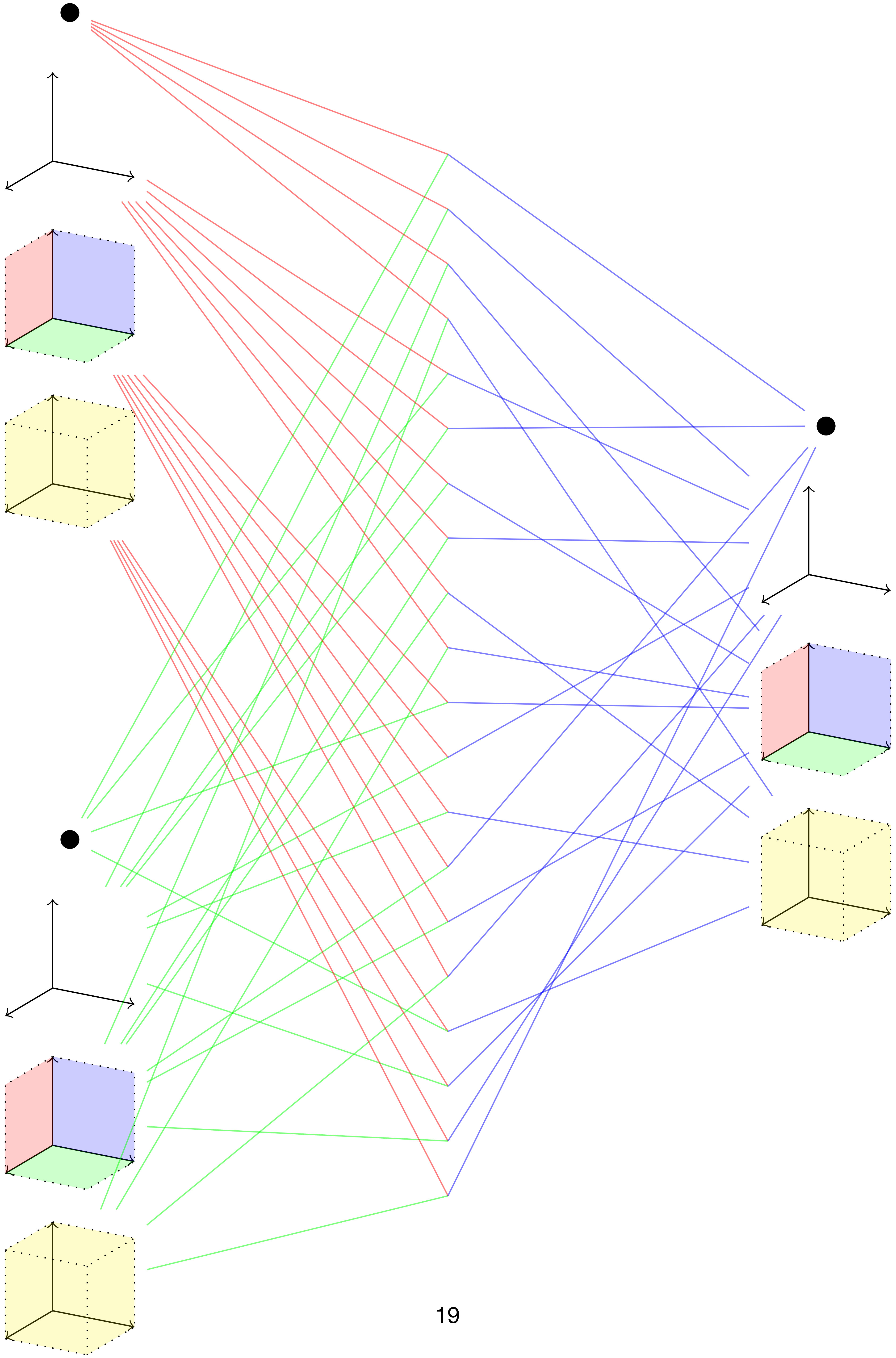
- Let $x_1, \ldots, x_{c_{\text{in}}}$ denote a set of multivectors.
- We can linearly combine them using $T^{\text{lin}}_{\phi_{c_{\text{out}}}}(x_1, \ldots, x_{c_{\text{in}}}) := \sum_{l=1}^{c_{\text{in}}} \phi_{c_{\text{out}} c_l} x_{c_l}$
- $\phi_{c_{\text{out}} c_{\text{in}}} \in \mathbb{R}$
- Or more densely: $T^{\text{lin}}_{\phi_{c_{\text{out}}}}(x_1, \ldots, x_{c_{\text{in}}})^{(k)} := \sum_{l=1}^{c_{\text{in}}} \phi_{c_{\text{out}} c_{in} k} x_{c_l}^{(k)}$

Methodology

Linear Layers "Multivector Neurons"



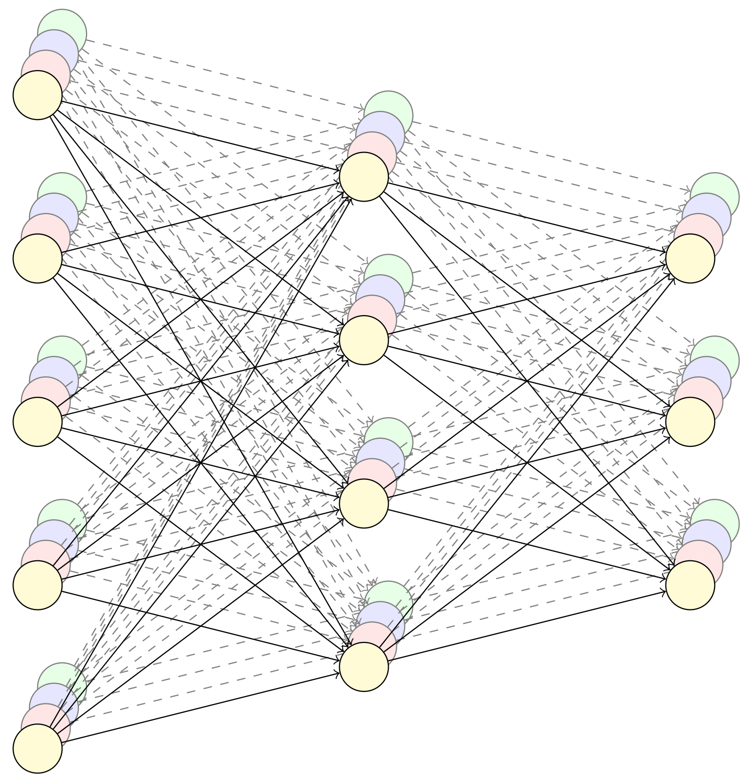




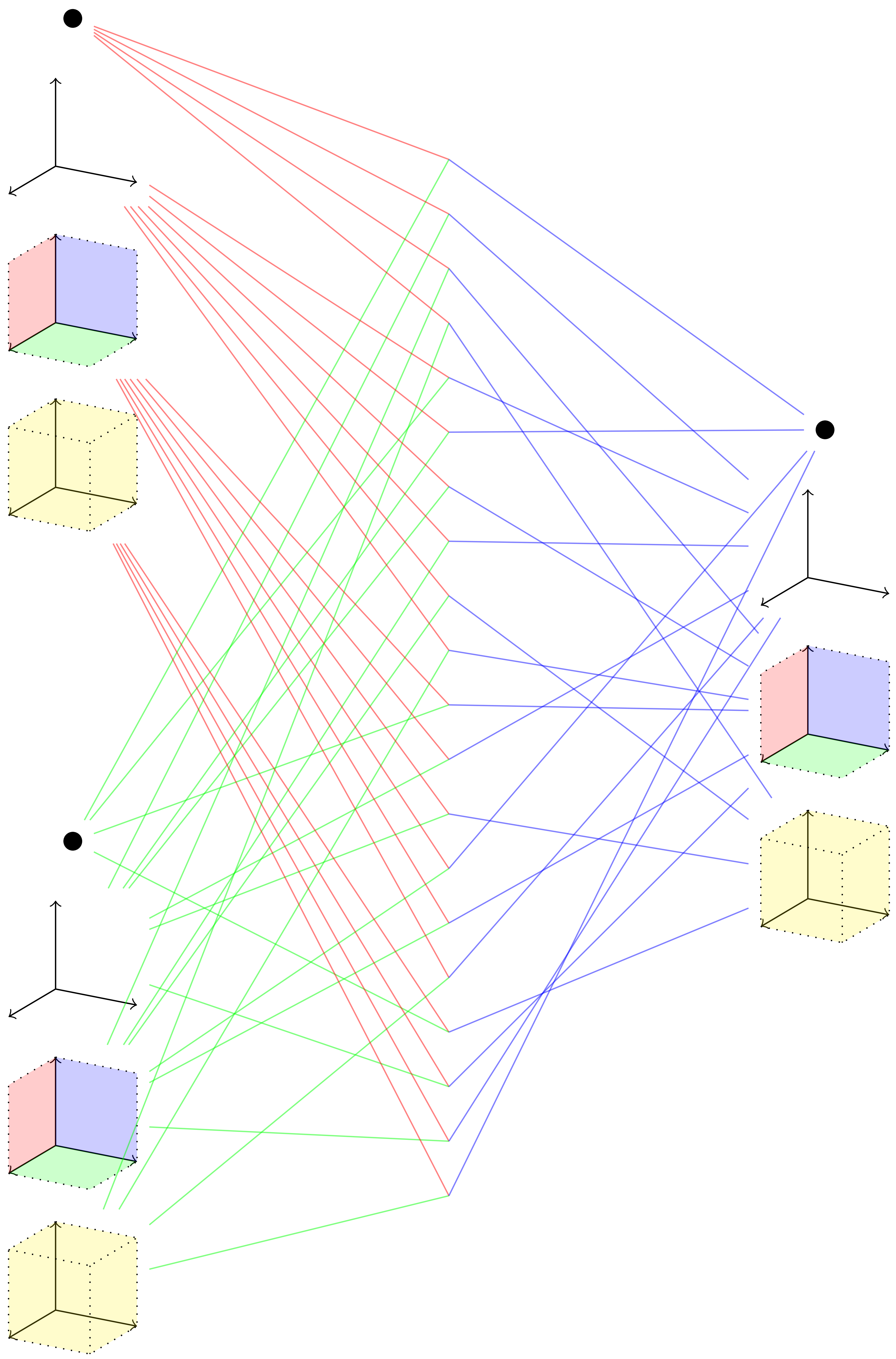
Methodology

Parameterized Geometric Product

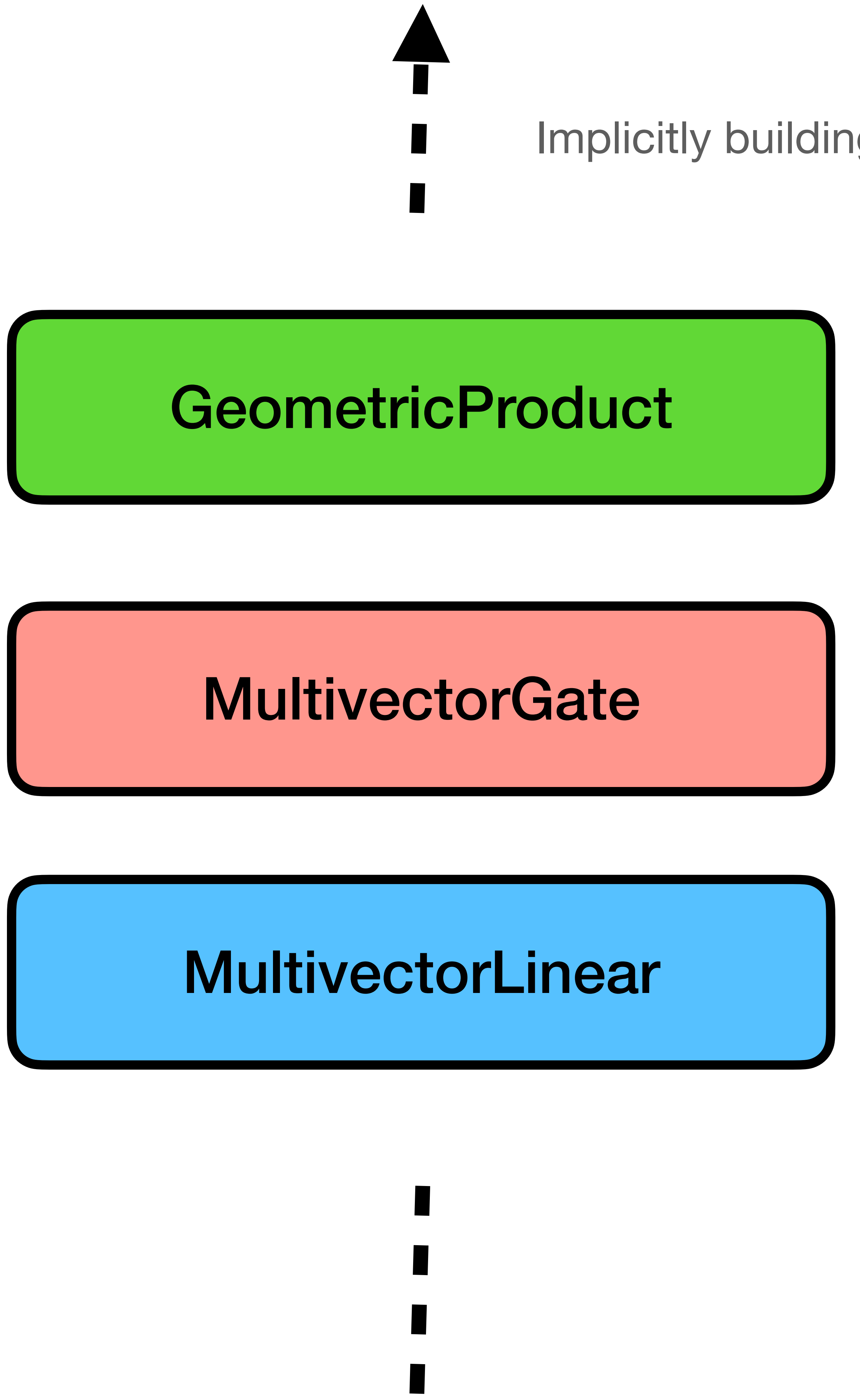
- $$P_\phi(x_1, x_2)^{(k)} := \sum_{i=0}^{n} \sum_{j=0}^{n} \phi_{ijk}(x_1^{(i)} x_2^{(j)})^{(k)}$$



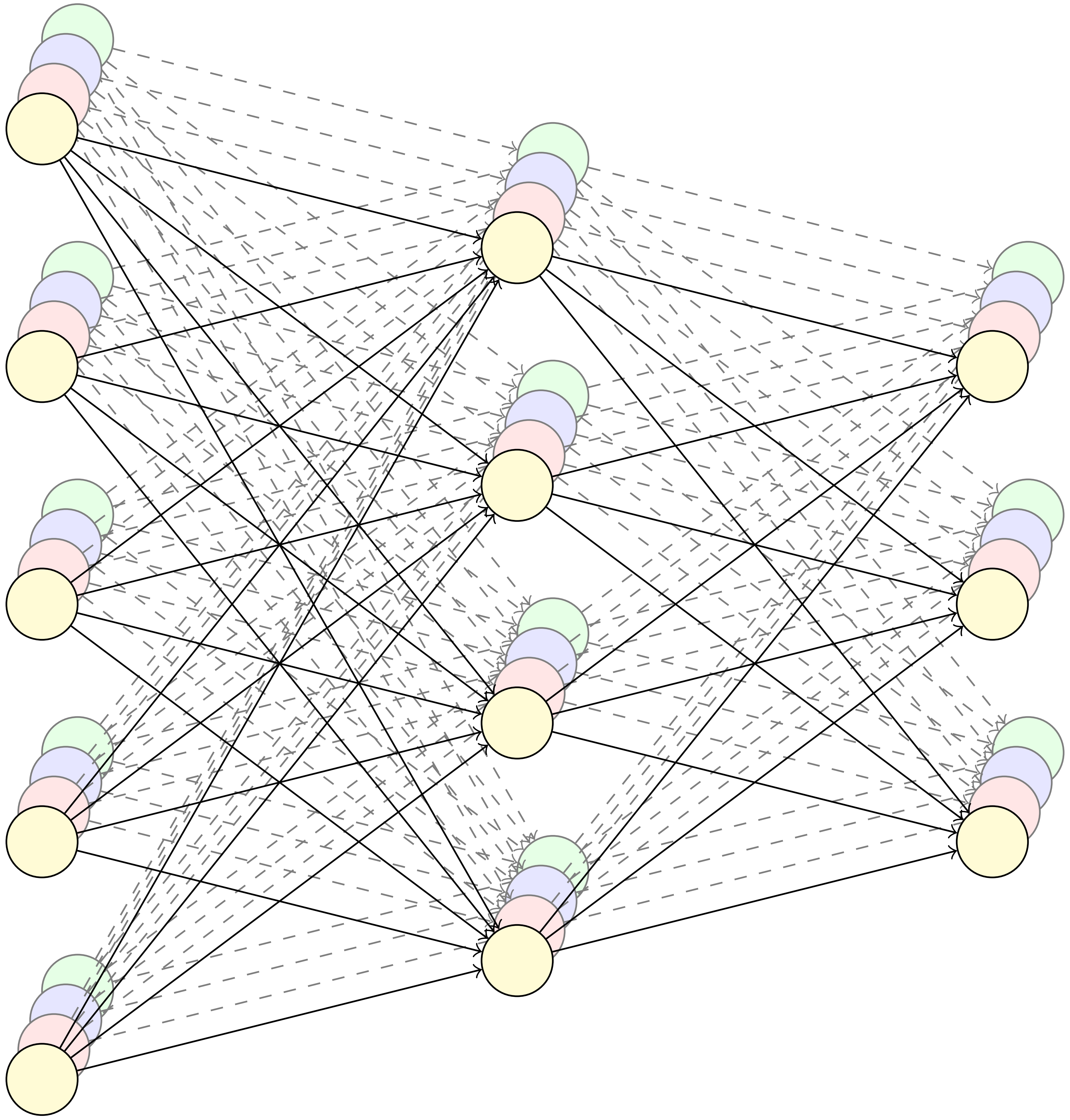
- All products:
  - $$T^{\text{prod}}(x_1, \ldots, x_{c_{\text{in}}})^{(k)} := \sum_{p=1}^{c_{\text{in}}} \sum_{q=1}^{c_{\text{in}}} P_{\phi_{pq}}(x_p, x_q)^{(k)}$$



Network Architectures



Implicitly building a big polynomial…



Experiments

E(3) **Experiment: n-body.**

- A benchmark for simulating physical systems using GNNs.
- Given $n = 5$ charged particles' positions and velocities, estimate their positions after 1000 time-steps.

Experiments

E(3) **Experiment: n-body. ($G(3)$)**

- A benchmark for simulating physical systems using GNNs.
- Given $n = 5$ charged particles' positions and velocities, estimate their positions after 1000 time-steps.

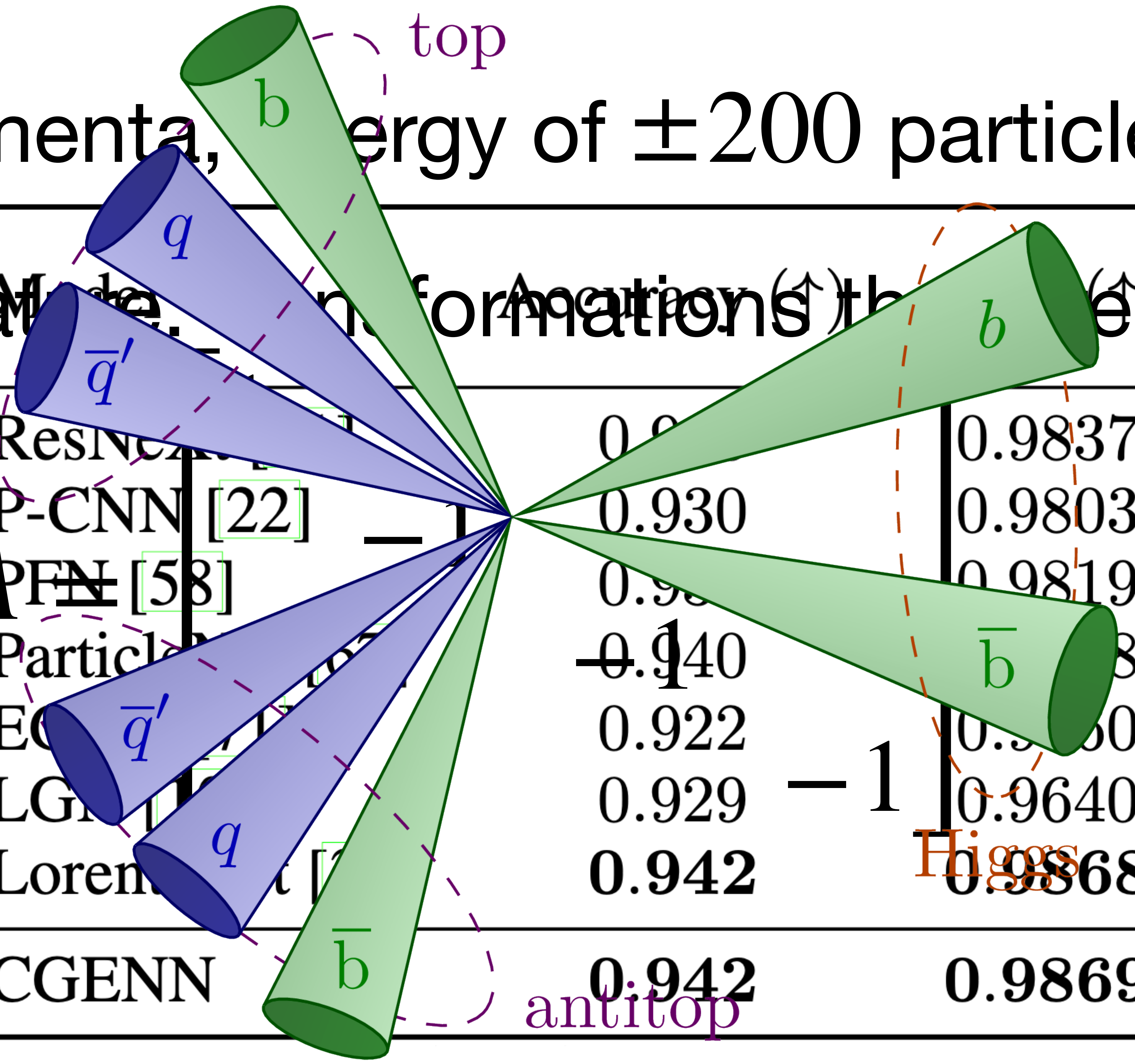
Method	MSE (↓)
SE(3)-Tr.	0.0244
TFN	0.0155
NMP	0.0107
Radial Field	0.0104
EGNN	0.0070
SEGNN	0.0043
CGENN	**0.0039 ± 0.0001**

Table 1: Mean-squared error (MSE) on the n-body system experiment.

Experiments

O(1,3) **Experiment: Top Tagging ($G(1,3)$)**

- Jet tagging: identifying particle jets generated during collisions.
- Top tagging: identifying whether event produced a top quark.
  - Given: momenta, energy of ± 200 particles.
- Relativistic nature. Transformations that preserve space-time distances given by O(1,3).



Model	Accuracy (↑)	AUC (↑)	$1/\epsilon_B$ (↑) ($\epsilon_S = 0.5$)	$1/\epsilon_B$ (↑) ($\epsilon_S = 0.3$)
ResNeXt	[illegible]	0.9837	302	1147
P-CNN [22]	0.930	0.9803	201	759
PFN [58]	[illegible]	0.9819	247	888
ParticleNet	0.940	[illegible]	397	1615
EGNN	0.922	[illegible]	148	540
LGN	0.929	0.9640	124	435
LorentzNet	**0.942**	**0.9868**	**498**	**2195**
CGENN	**0.942**	**0.9869**	**500**	**2172**

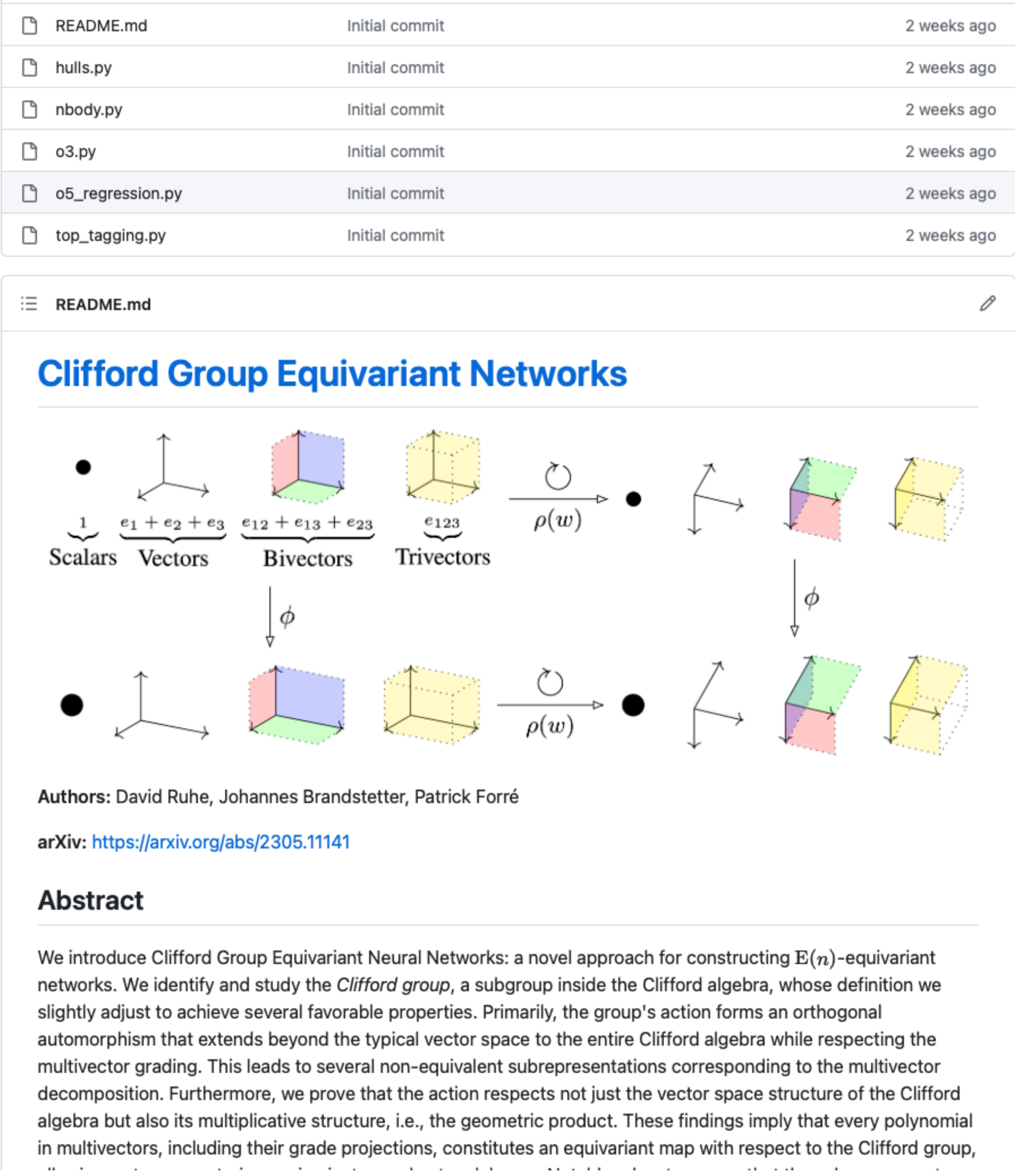
Remarks

- No need for group convolutions.
- We can directly use higher-order (vector) features instead of scalarized ones.
- CGENNs generalize to quadratic spaces of any dimension, can be equivariant to O(n), E(n), and subgroups.
- No spherical harmonics, CG coefficients, etc.

- Do not have all the SO(3) representations: is it a fundamental limitation?
- Are the representations we do have always irreducible?
- Geometric products are all you need in the nondegenerate case, not in the degenerate case.

Code & Efficiency

- Code is available at https://github.com/DavidRuhe/clifford-group-equivariant-neural-networks/
- Massive speed ups in JIT-compiled JAX versions.



Adjacent & Followup Works

- Geometric Algebra Transformer (Brehmer et al., 2023, NeurIPS 2023)
  - Lorentz-Equivariant GATr (Spinner et al., 2024)
- Clifford Simplicial Message Passing (Liu et al., 2024, ICLR 2024)
- Clifford-Steerable CNNs (Zhdanov et al., 2024, ICML 2024)
- Applications in
  - 3D vision (Pepe et al., 2024)
  - (Bio)chemistry (Pepe et al., 2024)
  - Fluid Mechanics (Maruyana et al., 2024).

Thanks