Geometric Algebra Transformers (GATr)

NN learns from symmetric data.

- **Symmetric data:** transform data without changing the context.
- Want NN built to be *equivariant* to transformations.
- **Equivariance:** exists a transformation for ***each*** input transformation.
  - Feature mapping $f: X \to Y$ is equivariant to a group of transformations if:
    - Every transformation $\pi \in \Pi$ of the input $x \in \mathrm{X}$
    - Is associable with a transformation $\psi \in \Psi$ of the features $f(x)$

$$\boxed{\boldsymbol{\psi}[\boldsymbol{f}(\boldsymbol{x})] = \boldsymbol{f}(\boldsymbol{\pi}[\boldsymbol{x}])}$$

- **Invariance:** exists a transformation for ***every*** input transformation
  - The feature transformation is not different for different input transformation.

E (3) symmetry = Euclidean 3D symmetry

- Symmetry to rigid transformations.
  - **Rigid transformations:** rotations, translations, flips.
    - **SE (3) symmetry:** E (3) without flips.
- **N-body experiment:** simulate movements of N point masses moving according to gravitation laws.
  - Law of physics don't depend on global translation and rotation.
  - With E (3) the physics law is invariant with respect to spatial translations.

GATr is an equivariant NN.

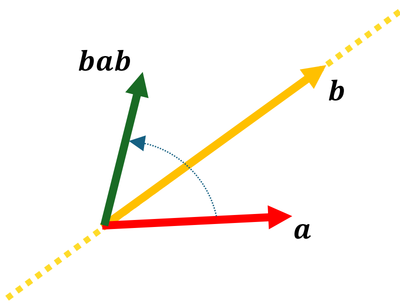
- **Input:** multivectors and additional scalars
  - **Multivectors:** object from geometric algebra that manages geometric objects at once.
    - Can be seen as array of 16 element.
    - Depending on which element it is activated, it represents a different object.
  - **Additional scalars:** lightweight positional embeddings to avoid additional overhead.
    - Multivectors can be noticeably big.
      - 10 multivectors $\cong$ 1 traditional transformer model
- Works as a transformer but embedded in geometric algebra context.
  - **Linear and Bilinear layers:** process individual sequence elements
  - **Attention layer:** mix between the sequence elements.
  - **Equivariant normalization:** consider norm of induvial vectors.
  - **Gated nonlinearity:** generalized nonlinearity.

Object / operator	Scalar 1	Vector e_0	e_i	Bivector e_{0i}	e_{ij}	Trivector e_{0ij}	e_{123}	PS e_{0123}
Scalar $\lambda \in \mathbb{R}$	λ	0	0	0	0	0	0	0
Plane w/ normal $n \in \mathbb{R}^3$, origin shift $d \in \mathbb{R}$	0	d	n	0	0	0	0	0
Line w/ direction $n \in \mathbb{R}^3$, orthogonal shift $s \in \mathbb{R}^3$	0	0	0	s	n	0	0	0
Point $p \in \mathbb{R}^3$	0	0	0	0	0	p	1	0
Pseudoscalar $\mu \in \mathbb{R}$	0	0	0	0	0	0	0	μ
Reflection through plane w/ normal $n \in \mathbb{R}^3$, origin shift $d \in \mathbb{R}$	0	d	n	0	0	0	0	0
Translation $t \in \mathbb{R}^3$	1	0	0	$\frac{1}{2}t$	0	0	0	0
Rotation expressed as quaternion $q \in \mathbb{R}^4$	q_0	0	0	0	q_i	0	0	0
Point reflection through $p \in \mathbb{R}^3$	0	0	0	0	0	p	1	0

Geometric algebra

- **Basis vector (e_i):** orthogonal vectors that represent multivector coordinates.
  - **Orthogonality:** dot product is zero
  - $\|e_i^2\| = 1$
- **Geometric product:** xy
  - Square must be equal to squared norm: $v * v = < v, v > = \|v\|^2$
    - **Special case:** $\|e_i * e_j\| = 2$
  - Antisymmetric: $e_i * e_j = -(e_j * e_i)$
    - **Proof:** $e_i e_j + e_j e_i = (e_i + e_j)^2 - e_i^2 - e_j^2 = 0$ ■
      - **Number of basis (d):** number of e_i.
    - It does not matter what order e_i are multiplied.
      - The output will have the same base vectors with different sign.
      - So, there will be $\sum_{i=0}^{d} \binom{d}{i} = 2^d$ base combinations (products)
        - **Number of basis (d):** count of e_i vectors
        - **Geometric algebra dimension:** $\sum_{i=0}^{d} \binom{d}{i} = 2^d$
- **Multivector:** $x = x_s + x_1 e_1 + x_2 e_2 + x_3 e_3 + x_{12} e_1 e_2 + x_{13} e_1 e_3 + x_{23} e_2 e_3 + x_{123} e_1 e_2 e_3$
  - This formulation is specifically for 3D Euclidean geometry.
  - **Grade:** number of basis associate to x_i
- **Inner product:** $< x, y > = (xy + yx)/2$
  - Symmetric part of the geometric product
  - Computes the similarity between vectors.
- **Outer product:** $x \wedge y = (xy - yx)/2$
  - Antisymmetric part of the geometric product
  - Represents the weighted and oriented subspace spanned by the vector.
  - $< x, y > + x \wedge y = xy$
- **Wedge product:** represent the objects intersection.
- **Dual:** $x \longmapsto x^*$
  - Sends basis vectors to their inverse.
  - As bit flip
    - $e_1 \longmapsto e_{23} \equiv e_{001} \longmapsto e_{110}$
- **Grade involution ($\hat{x}$):** flips the sign of the odd-grade elements of x
- **Sandwich product:** $\rho(x)_u = u[x] = \begin{cases} uxu^{-1} & if\ u\ is\ even \\ u\hat{x}u^{-1} & if\ u\ is\ odd \end{cases}$
  - Reflect the vector around a point.
  - A rotation can be represented as two reflections around vectors.
- **Left contraction:** $x \rfloor y =< xy >_{l-k}$
  - Takes parts geometric product part that corresponds to the difference of vectors grades.
- **Join:** $x, y = (x^* \wedge y^*)^*$
  - Join an object between two elements.
- **Blades:** $x = x_1 \wedge x_2 \wedge x_3 \wedge \ldots \wedge x_k$
  - Wedge product of more vectors of grade i from 1 to k.

2D Geometric algebra ($\mathbb{G}_{2,0,0}$)



- Two coordinates e_1, e_2
  - **Multivector:** $x = x_s + x_1 e_1 + x_2 e_2 + x_{12} e_1 e_2$
    - **Scalar (x_s):** grade 0
    - **Vector ($x_i e_i$):** grade 1 = point
    - **Bivector ($x_{12} e_1 e_2$):** grade 2
  - **Number of bases:** $d = 2$
  - **Geometric algebra dimension:** $2^d = 4$
- **Inner product:** dot product
- **Outer product:** gives a bivector.
  - **Bivector area:** $||u \wedge v|| = ||u|| * ||v|| * \sin(\theta)$
- **Sandwich product:** reflection of a vector on another one

Projective 2D Geometric algebra ($\mathbb{G}_{2,0,1}$)

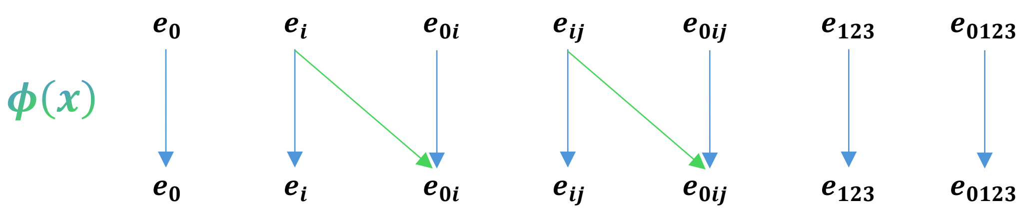
- Three coordinates e_0, e_1, e_2
  - **Multivector:** $x = x_s + x_0 e_0 + x_1 e_1 + x_2 e_2 + x_{01} e_0 e_1 + x_{02} e_0 e_2 + x_{12} e_1 e_2 + x_{012} e_0 e_1 e_2$
    - **Scalar (x_s):** grade 0
    - **Vector ($x_i e_i$):** grade 1 = line
    - **Bivector ($x_{ij} e_i e_j$):** grade 2
    - **Trivector ($x_{012} e_0 e_1 e_2$):** grade 3
    - Describe a line.
  - $e_0^2 = 0$
    - Destroy bivector/trivectors.
  - **Number of bases:** $d = 3$
  - **Geometric algebra dimension:** $2^{d+1} = 8$
- **Join:** the output of joining two objects
  - Two joined points (scalar) give a line (vector).

Projective 3D Geometric algebra ($\mathbb{G}_{3,0,1}$)

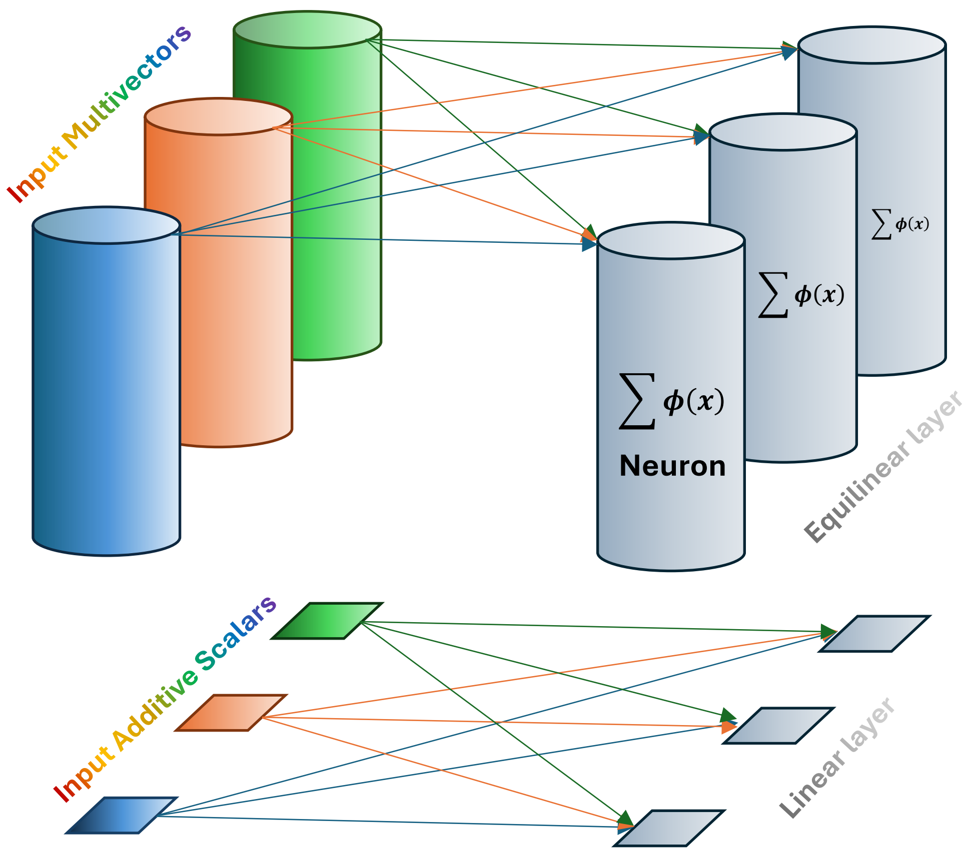
- The geometric algebra used by GATr.
- Four coordinates e_0, e_1, e_2, e_3
    - **Multivector:** $x = x_s + \sum_{i=0}^{3} x_i e_i + \sum_{i=0}^{2} \sum_{j=i+1}^{3} x_{ij} e_i e_j + \sum_{i=0}^{1} \sum_{j=i+1}^{2} \sum_{k=j+1}^{3} x_{ijk} e_i e_j e_k + x_{0123} e_0 e_1 e_2 e_3$
    - **Scalar (x_s):** grade 0
    - **Vector ($x_i e_i$):** grade 1 = plane
    - **Bivector ($x_{ij} e_i e_j$):** grade 2
    - **Trivector ($x_{012} e_0 e_1 e_2$):** grade 3 = points
    - Describe a plane.
  - $e_0^2 = 0$
    - Destroy bivector/trivectors.
  - **Number of bases:** $d = 3$
  - **Geometric algebra dimension:** $2^{d+1} = 16$
- **Wedge product:** the output of intersect two objects.
  - Two intersected planes (vectors) give a line (bivector).
  - Is a geometric product for orthogonal planes (vectors).
  - Opposite of joint

Equivariant linear layer: perform blade projection.

- GATr neurons are multivectors.
- **GATr linear layer:** linear mapping from a (input) multivector to a (neuron) multivector
  - One mapping for each multivector.
    - $\boldsymbol{n}^2$ mappings, where $\boldsymbol{n}$ = input multivectors
  - Summed all together at each output neuron.
- An equivariant linear layer has always the form $\boldsymbol{\phi}(\boldsymbol{x}) = \sum_{k=0}^{d+1} \boldsymbol{w}_k < \boldsymbol{x} >_k + \sum_{k=0}^{d} \boldsymbol{v}_k \boldsymbol{e}_0 < \boldsymbol{x} >_k$
  - Sum of weighted blade projections
  - Each weight is a learnable scalar.
    - $\boldsymbol{w}_k$ multiplies each multivector ($< \boldsymbol{x} >_k$) component.
      - $\boldsymbol{e}_i \rightarrow \boldsymbol{e}_i$ map
    - $\boldsymbol{v}_k$ multiplies each multivector ($< \boldsymbol{x} >_k$) that does not have $\boldsymbol{e}_0$ component.
      - Turns $< \boldsymbol{x} >_k$ component into the analogue with $\boldsymbol{e}_0$ component.
      - $\boldsymbol{e}_i \rightarrow \boldsymbol{e}_{0i}$ map
  - **Blade projection ($< \boldsymbol{x} >_k$):** multivector $\boldsymbol{x}$ with only the components of grade $\boldsymbol{k}$
    - So, each component of grade $\boldsymbol{k}$ will have same weights.
    - The geometric algebra linear projection



- Addictive scalars are processed with a classic linear layer.
  - Weight + bias
  - Equilinear layer **CANNOT** have bias.
    - Loss of equivariance



Geometric bilinear layer: perform geometric operations on multivectors.

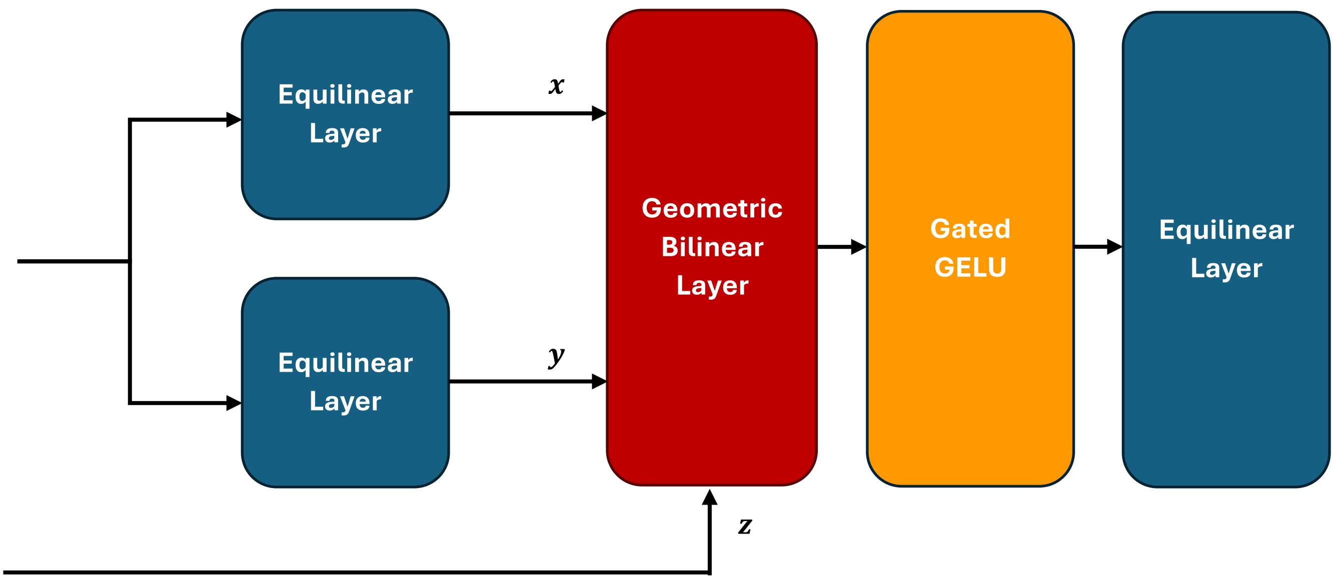
- Works on splitted equilinear layer outputs
- Combines geometric product (equivariant by definition) and (equivariant) joint of the splitted inputs.
  - $\boldsymbol{Geometric(x,y) = Concat_{channels}(xy, EquiJoin(x,y;z))}$
- Joins are necessary to distinguish distances.
  - $\boldsymbol{x,y,z \mapsto EquiJoin(x,y;z) = z_{0123}(x^* + y^*)^*}$
  - Realize (inner) geometric product between the (equivariant) join.
    - Of the splitted outputs and a reference
    - **Reference ($\boldsymbol{z}$)**: pseudoscalar of the splitted inputs mean.
  - Join element "measuring" their distances.
    - Like point-to-point distance
  - The multiplication by $\boldsymbol{z}$ maintain the equivariance.
    - Because joint alone is not equivariant to mirroring
- The output is projected with an Equilinear layer.
  - $\boldsymbol{\phi(Geometric(x,y))}$

Layer normalization

- $\boldsymbol{LayerNorm(x) = \frac{x}{\sqrt{\mathbb{E}_c<x,x>}}}$
  - Equivariant approach to find the layer normalization.
  - Denominator = Square root of the average $<\boldsymbol{x},\boldsymbol{x}>$ over the channels
  - Invariant due the invariance of the inner product

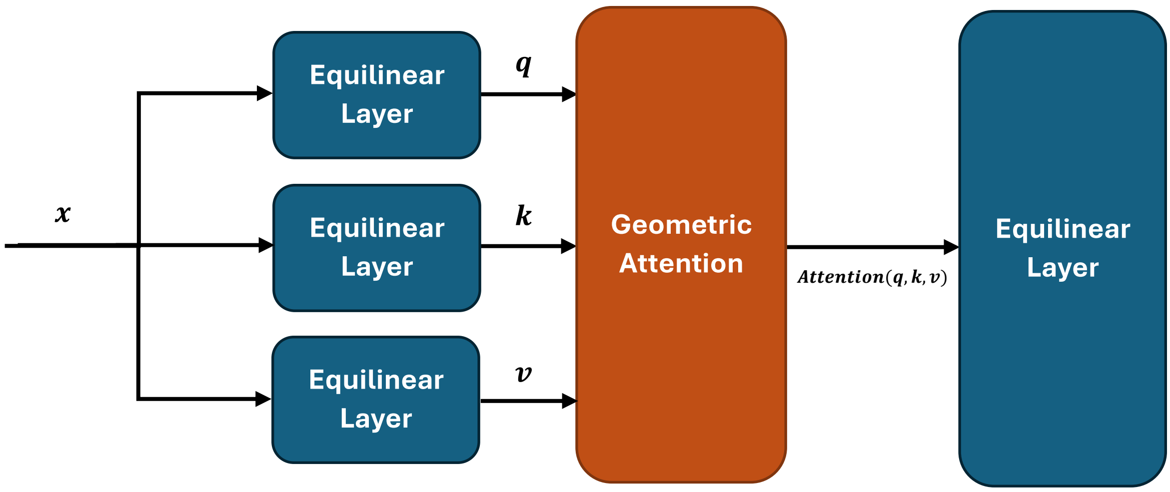
Gated GELU

- $\boldsymbol{GatedGELU(x) = GELU(x_s)x}$
  - Classic GELU computed only on the scalar component of the multivector.
  - Its output is used to gate the whole multivector.

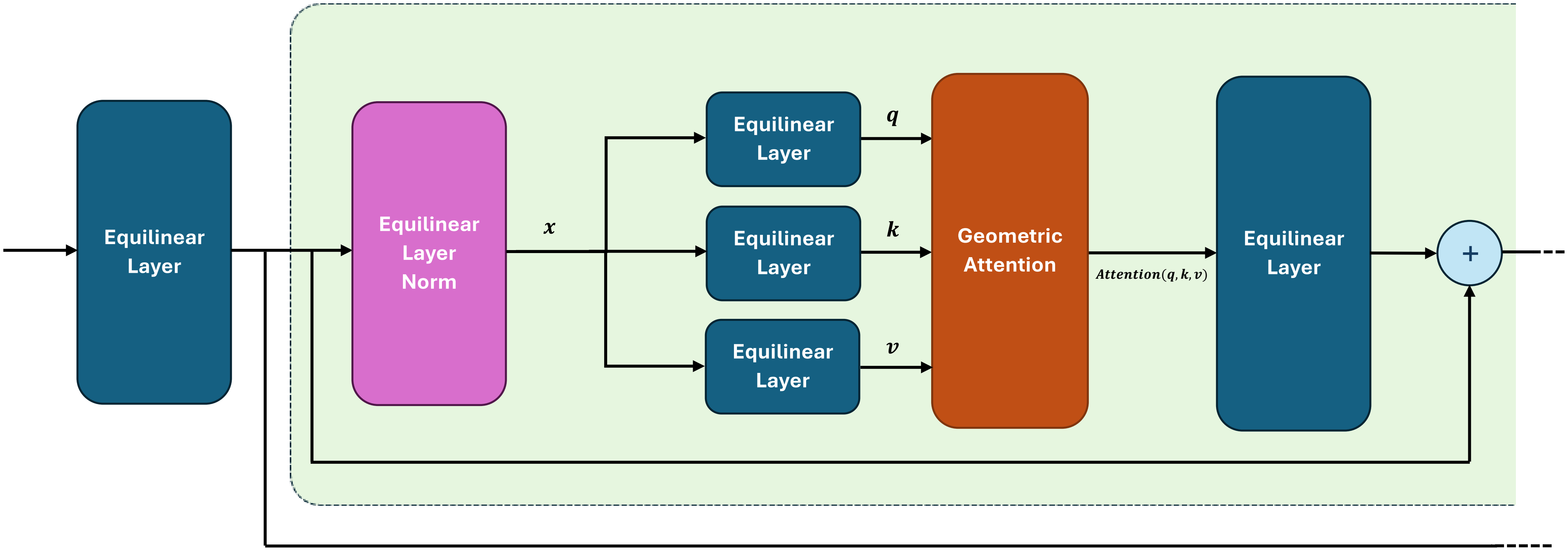


Attention

- $\boldsymbol{Attention(q,k,v)_{i',c'} = \sum_i Sofmax\left(\frac{\sum_c < q_{i',c}, k_{i,c} >}{\sqrt{8n_c}}\right) v_{i,c'}}$
  - Scaled-inner attention.
  - With the inner product, guarantee of invariance.
  - $\boldsymbol{n_c}$ = head dimension = key length
  - Each query, key, value consists in $\boldsymbol{n_i}$ tokens.
- Inner product does not consider the 8 $\boldsymbol{e_{0x}}$ components.
  - Multiplication by two $\boldsymbol{e_0}$ gives zero.
  - $\boldsymbol{8n_c}$ allow to take that into account.
  - They absence is not a numerical problem, thanks to this.
  - But it is a conceptual problem in the attention context.
- **Nonlinear features:** replace the $\boldsymbol{e_{0x}}$ in the attention.
  - $\boldsymbol{< q_{i,c}, k_{i,c} > = < q_{i,c}, k_{i,c} > + \phi(q_{ic}) + \psi(k_{ic})}$
  - $\boldsymbol{\phi(q) = \omega(q_{\backslash 0}) \begin{pmatrix} q_{\backslash 0}^2 \\ \sum_i q_{\backslash i}^2 \\ q_{\backslash 0} q_{\backslash 1} \\ q_{\backslash 0} q_{\backslash 2} \\ q_{\backslash 0} q_{\backslash 3} \end{pmatrix}}$
  - $\boldsymbol{\phi(q) = \omega(k_{\backslash 0}) \begin{pmatrix} -\sum_i k_{\backslash i}^2 \\ -k_{\backslash 0}^2 \\ 2k_{\backslash 0} k_{\backslash 1} \\ 2k_{\backslash 0} k_{\backslash 2} \\ 2k_{\backslash 0} k_{\backslash 3} \end{pmatrix}}$
  - $\boldsymbol{\omega(x) = \frac{x}{x^2 + \epsilon}}$
    - $\boldsymbol{\epsilon}$ avoids numerical instability.
  - $\boldsymbol{q, k}$ are trivectors.
    - $\boldsymbol{q_{\backslash i}, k_{\backslash i}}$ indicate the component of the trivector with all indices but $\boldsymbol{i}$



Full architecture (1)



Full architecture (2)