Geometric Algebra Transformer

Kimia Hafezi
La Sapienza
University of Rome

Atra Hossein Tafreshi
La Sapienza
University of Rome

Silvestri Matteo
La Sapienza
University of Rome

Emails: {silvestri.1872908, hafezi.1908578, hosseintafreshi.1908579}@studenti.uniroma1.it

Abstract

In recent years, Transformers have revolutionized the field of deep learning, demonstrating unparalleled performance across a wide range of tasks. However, the representation of data in these models often lacks the geometric structure inherent to many real-world phenomena, potentially limiting the depth of understanding and inference capabilities of the models. This research is based on the Geometric Algebra Transformer (GATr) architecture, an approach that integrates geometric algebra into the transformer framework to leverage the rich mathematical framework for representing complex geometric structures and operations within deep learning models. Dealing with geometric data is very often nowadays, It occurs in a lot of fields such as Chemistry, Physics, Medicine, etc. In this paper, we delve into the GATr architecture for artery classification on the Vessel dataset [1]. Additionally we do some modifications to have light-weight model with less parameters to have more efficient model.

1 Introduction

In this first section, we explore the Geometric Algebra background and the dataset for the task. This paper is a short report of our code implementation available on the link

https://colab.research.google.com/drive/1mcE6-DXg_bGVB4ltffEVXUd1mzcSR31T?usp=sharing

1.1 Projective Geometric Algebra

Since we want to represent geometric objects of $\mathbb{R}^3$, we need a multidimensional space where we can represent easily different types such as points, lines, planes. Therefore, we work in the clifford $\mathbb{G}_{3,0,1}$ PGA; this is a 16-dimensional space, in which the basis elements are $\{1, e_0, e_1, e_2, e_3, e_{01}, e_{02}, e_{03}, e_{12}, e_{13}, e_{23}, e_{012}, e_{013}, e_{023}, e_{123}, e_{0123}\}$. The element of grade 0 is called *scalar*, then we have *vectors*, *bivectors*, *trivectors* and the last element is called *pseudoscalar*.

Operations The bivector e_{01}, for instance, is the combination of the vectors e_0, e_1 through the so-called *geometric product*. This corresponds to the main operation between multivectors and is composed of a symmetrical part, the *inner product* $\langle \cdot, \cdot \rangle$, and an antisymmetrical part, the *outer product* $\wedge$. The former calculates the similarity, while the latter generates the subspace generated by the two multivectors. With an abuse of notation, we can represent the three operations as follows

$$xy = \langle x, y \rangle + x \wedge y \tag{1}$$

In a very crude way, one can think of the inner product as a dot product and the outer product as a cross product. Indeed, we have $\langle e_i, e_j \rangle = \delta_{ij}$ and $e_i \wedge e_i = 0$ as we are normally used to. In particular, we have that $vv = v^2 = \langle v, v \rangle$ so the square of a vector is its squared norm. In the algebra of our interests, the indices $\{3, 0, 1\}$ specify that there are three vectors with norm 1 and one vector with norm 0; in our notation, we have $e_0^2 = 0$ and $e_1^2 = e_2^2 = e_3^2 = 1$.
There are other two important operations within $\mathbb{G}_{3,0,1}$ algebra : the *dual* operator $*$ and the *join* operator $\vee$. The former exchanges "empty" dimensions for "full" dimensions; for instance, $*e_0 = e_{123}$ or $*e_{01} = e_{23}$. The latter is given by

$$x \vee y = (x^* \wedge y^*)^* \tag{2}$$

Equivariance We analyse the stability of the models with respect to Euclidean transformations $\mathbf{E}(3)$. To test this property, we work with versors $u = u_1 ... u_k$ where u_i is a reflection, since any transformation is equal to a sequence of reflections. These elements form a group called $Pin(3,0,1)$, that cover $\mathbf{E}(3)$ within $\mathbb{G}_{3,0,1}$ algebra. If we consider only versors defined by an even number of reflections, then we have the group $Spin(3,0,1)$ that is related to **SE**(3). We will say that a function $f : \mathbb{G}_{3,0,1} \to \mathbb{G}_{3,0,1}$ is *Pin-equivariant* if $\forall x \in \mathbb{G}_{3,0,1}$ and $\forall u \in Pin(3,0,1)$ it holds that

$$f(\rho_u(x)) = \rho_u(f(x)) \tag{3}$$

where ρ_u is the so-called *sandwich product*. This operation is given by

$$\rho_u(x) = \begin{cases} uxu^{-1} & \text{if } u \text{ is even} \\ u\hat{x}u^{-1} & \text{if } u \text{ is odd} \end{cases} \tag{4}$$

where $\hat{x}$ it's the *grade involution* of multivector x, meaning that it has flipped odd-grade components.

Embeddings About how we embed geometric objects into the clifford algebra, we used the reference [2].

- A *scalar* $\lambda \in \mathbb{R}$ becomes a multivector given by $[\lambda, 0, ..., 0]$
- A *point* $x = x_1 e_1 + x_2 e_2 + x_3 e_3 \in \mathbb{R}^3$ becomes a multivector X given by
$$X = (1 + e_0 x)e_{123} = e_{123} + x_1 e_{023} - x_2 e_{013} + x_3 e_{012} \tag{5}$$
- A *plane* in space is determined by an equation $ax + by + cz + d = 0$, where $n = (a, b, c)$ it's the normal and d it's the distance from the origin (dn is a location on the plane). In algebra $\mathbb{G}_{3,0,1}$, this plane X becomes a multivector given by
$$X = de_0 + ae_1 + be_2 + ce_3 \tag{6}$$
- A *line* X defined by a direction $v \in \mathbb{R}^3$ and a location $p \in \mathbb{R}^3$ is given by
$$\begin{aligned} X &= (v + e_0(p \wedge v))e_{123} = ve_{123} + e_0(p \wedge v)e_{123} \\ &= \Big[v_1 e_{23} - v_2 e_{13} + v_3 e_{12}\Big] + \Big[(p_3 v_2 - p_2 v_3)e_{01} + (p_1 v_3 - p_3 v_1)e_{02} + (p_2 v_1 - p_1 v_2)e_{03}\Big] \end{aligned} \tag{7}$$

1.2 Dataset

The dataset consists of 4000 arteries: 2000 of the single type with stenosis, 2000 of the bifurcated type. Each sample consists of five fields:

- *pos*: positions of the points that make up the artery
- *wss*: wall shear stress vector for each point
- *pressure*: pressure value for each point
- *face*: triples of numbers corresponding to the points forming a triangle in the mesh
- *inlet idcs*: location(in terms of points) of inlets to other arteries

We embed *pos* as points, *wss* as lines, *pressure* as scalar. About the variable *face*, given three points P,Q,R the normal will be $n = \vec{PQ} \times \vec{PR}$ while the distance from the origin will be $d = n \cdot P$. We didn't use *inlet idcs* for the task.
In the colab notebook, it is clearly shown how the data belonging to the two classes are very distinct from each other. For example, if only the average pressure is taken as a variable, a simple linear classifier would perform well. Also for this, the results obtained with the models described in the next section are very good.

2 Architectures

We decided to analyse four different models: the first is the *GATr* model as presented in the reference [3]; the second, is a variant using an attention method called *Light Weight* introduced in [4]; finally, we wanted to compare these models with the *baseline Transformer* and the baseline transformer with light weight attention.

2.1 GATr

Let's explain the architecture shown in figure 1. As you can see, the model is able to work with multivector and scalar input in parallel. Moreover, we work these inputs separately except in the Equilinear layer, where the scalar component of the multivector and scalar input are connected to each other.

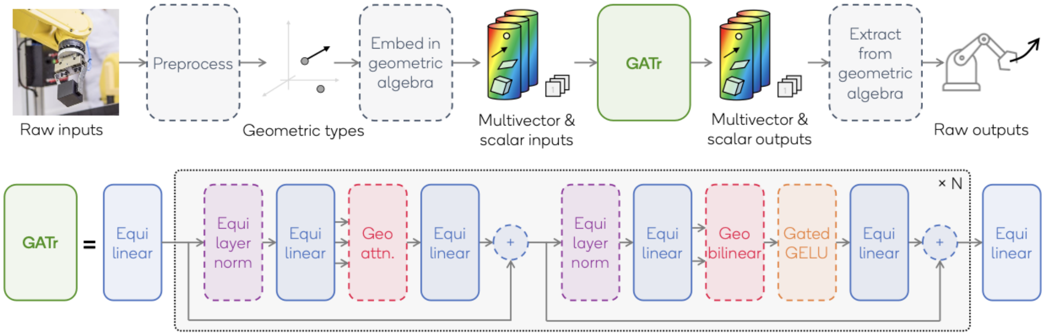


Figure 1: GATr architecture

Linear layer We want to construct a linear layer that is Pin-equivariant according to the previous definition. There are two types of equivariant function: one is the so-called *blade* $f(x) = \langle x \rangle_k$, that takes a multivector as input and put all its components of grade $\neq k$ to 0; the other one is $f(x) = e_0 \langle x \rangle_k$. In more detail, it is proved that every linear function f of the algebra $\mathbb{G}_{3,0,1}$ is of the form

$$f(x) = \sum_{k=0}^{4} w_k \langle x \rangle_k + \sum_{k=0}^{3} v_k e_0 \langle x \rangle_k \tag{8}$$

for parameters $w \in \mathbb{R}^5, v \in \mathbb{R}^4$.

Norm layer We define a Pin-equivariant norm layer on multivectors given by the operation

$$\text{layerNorm}(x) = \frac{x}{\sqrt{\mathbb{E}_c \langle x, x \rangle}} \tag{9}$$

where $\mathbb{E}_c$ is the expectation over channels and we use the inner product of $\mathbb{G}_{3,0,1}$ algebra. This implies $\mathbb{E}_c||\text{inputs}||^2 = 1$.

Attention Just as in the baseline transformer, we compute scalar attention weights with a scaled dot product. The difference is that we use the inner product of $\mathbb{G}_{3,0,1}$ algebra, which is the standard dot product ignoring the dimensions containing e_0.

Geometric bilinears Because of the very limited grade mixing, equivariant linear maps are not sufficient to build expressive networks capable of build geometric features from existing ones. For this reason we introduce this layer given by

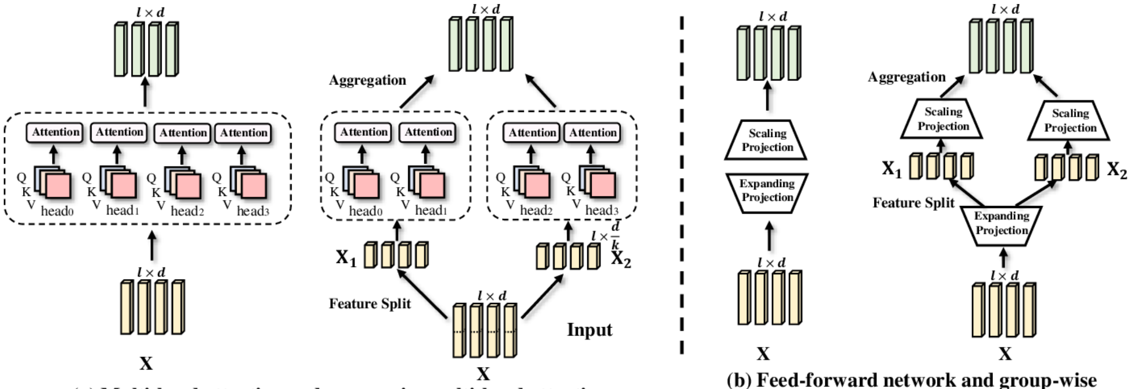
$$\text{Geometric}(x, y; z) = \text{Concatenate}_c(xy, \text{Equijoin}(x, y; z)) \tag{10}$$

where $\text{Equijoin}(x, y; z) = z_{0123}(x \vee y)$ and z it the average of all the input to the network.

GatedGELU We use scalar-gated GELU nonlinearities given by $\text{GatedGELU}(x) = \text{GELU}(x_1)x$ where x_1 is the scalar component of the multivector x.

2.2 Light Weight GATr

For implementing the attention mechanism for GATr architecture, we follow three mechanism: first is *multi-head* attention that is proposed in the reference paper, second one is *multi-query* attention, and third is *light-weight* attention. Multi-head attention provides the model with the flexibility to focus on different parts of the input sequence and leads to better model performance, but the model with multi-head attention would have more parameters and computational cost. For this reason authors test the GATr also with multi-query attention. Whenever multi-query attention is enough for our purpose based on the complexity of the task, it would result in lower number of parameters. The third approach that we use and is group-wise multi-head attention. The proposed approach is consistent with MHA, which can ensure LW-Transformer to learn similar attention patterns as original Transformer, while reducing parameters and computational costs a lot. In figure 2, we show that given the input features X, we first divide it into k groups before projection. Those features of different groups are transformed and the multi-head attention will be compute within each



(a) Multi-head attention and group-wise multi-head attention.

(b) Feed-forward network and group-wise feed-forward network.

Figure 2: Group-wise Multi-Head Attention

group, and at the last the results will be aggregated. Below is its formula, where each X_i is feature of i-th group and $[\cdot]$ shows the concatenation:

$$G\text{-}MHA(X) = \left[\tau(X_1), \ldots, \tau(X_k)\right], \tag{11}$$

where $\forall i \in \{1...k\}$ we have

$$\tau(X_i) = MHA(X_i)$$

2.3 Baseline Transformer

The primary role of the Transformer encoder is to transform the input data into a high-dimensional space where the relationships between different parts of the data are more explicitly modeled through self-attention mechanisms. The decoder part of the Transformer model is primarily responsible for generating output sequences based on the context provided by the encoder. Since classification tasks are about making decisions or predictions based on the input rather than creating new sequences, the decoder is not necessary here in our classification task. Also using only the encoder part, reduces the computational complexity and resources required and it is enough for our goal. Therefore we have implemented the encoder part of the transformer based on the proposed architecture in [5] with a slight modification. In the implemented baseline, after processing the input sequence through the encoder, a classification layer (a fully connected layer with a sigmoid function) is applied directly to the encoder's output to produce the classification labels. This layer can leverage the rich, contextualized representations of the input sequence created by the encoder to perform accurate classification.

3 Equivariance Check

In this section we want to analyse the model's equivariance respect to Pin and $Spin$ transformations. The $Spin$ group includes : translations, linear reflections, rotations about a line, and screw motions(i.e. rotation about a line and a translation along the same line); while the Pin group includes also planar reflections, rotoreflections and point reflections. This table shows that our model is $Spin$ equivariant but not perfectly Pin equivariant and the problem seems to be related to the EquiJoin operation within the Geomtric MLP. Indeed, this problem is confirmed by the results obtained in section 5, since we have different behaviors of the model depending on which group the transformation belongs to.

Layer	Pin transformation	Spin transformation
Linear layer	0.0000489	0.0000441
Norm layer	0.0000472	0.0000247
Geometric MLP	384.87	0.0000872
Attention layer	0.0000567	0.0000424

4 Evaluations on standard dataset

For evaluation, we will show the test accuracy, and F1 score for our task. F1 score is useful when we need a single metric to reflect both precision and recall. A high F1 score indicates that the model has a low rate of false positives and false negatives, suggesting it is robust and performs well across the aspects of precision and recall. As we report also in our colab notebook, data are very distinct between the two classes. This will ensure that all models that will be implemented and analysed will achieve a very high accuracy. Another aspect that is important for evaluate and report, is the model parameters based on using the different approaches (Multi Head GATr and Light Weight GATr). As it is shown below, there is a significant reduce in the number of parameters when using the light weight version of the GATr, while both give us the same accuracy.

Model	Accuracy	F1-score	Num Parameters
Transformer	0.99	0.95	4 K
MH GATr	1	1	258 K
LW GATr	1	1	41 K

5 Evaluations on rotated dataset

Finally, we tested the model by rotating the input data (such as *pos* and *wss*) to check equivariance. To do this, we used a 3D rotation matrix accordint to the reference [Rotation matrix]. Below we report the results applying a $Spin$ trasformation composed of a line reflection and rotation around an axis.

Model	Accuracy	F1-score
Transformer	0.98	0.95
LW GATr	0.99	1

This shows how the GATr model is stable with respect to this transformation. This is not the case for any type of rotation. In fact, by making a rotation of 60 degrees with respect to all axes (Pin transformation), the GATr model is affected by this transformation and the accuracy drops around 0.75.

References

[1] Julian Suk, Pim de Haan, Phillip Lippe, Christoph Brune, and Jelmer M. Wolterink. Mesh neural networks for se(3)-equivariant hemodynamics estimation on the artery wall, 2023.

[2] Leo Dorst. *A Guided Tour to the Plane-Based Geometric Algebra PGA*. University of Amsterdam, 2023.

[3] Johann Brehmer, Pim de Haan, Sönke Behrends, and Taco Cohen. Geometric algebra transformer, 2023.

[4] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Yan Wang, Liujuan Cao, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Towards lightweight transformer via group-wise transformation for vision-and-language tasks. *IEEE Transactions on Image Processing*, 31:3386–3398, 2022.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[3] [2] [4] [1] [5]