Euclidean, Projective, Conformal: Choosing a Geometric Algebra for Equivariant Transformers

Pim de Haan PIM@QTI.QUALCOMM.COM
Taco Cohen TACOS@QTI.QUALCOMM.COM
Johann Brehmer JBREHMER@QTI.QUALCOMM.COM
Qualcomm AI Research[*]

Abstract

The Geometric Algebra Transformer (GATr) is a versatile architecture for geometric deep learning based on projective geometric algebra. We generalize this architecture into a blueprint that allows one to construct a scalable transformer architecture given *any* geometric (or Clifford) algebra. We study versions of this architecture for Euclidean, projective, and conformal algebras, all of which are suited to represent 3D data, and evaluate them in theory and practice. The simplest Euclidean architecture is computationally cheap, but has a smaller symmetry group and is not as sample-efficient, while the projective model is not sufficiently expressive. Both the conformal algebra and an improved version of the projective algebra define powerful, performant architectures.

1. Introduction

Geometric problems require geometric solutions, such as those developed under the umbrella of geometric deep learning (Bronstein et al., 2021). The primary design principle of this field is equivariance to symmetry groups (Cohen and Welling, 2016): network outputs should transform consistently under symmetry transformations of the inputs. This idea has sparked architectures successfully deployed to problems from molecular modelling to robotics.

In parallel to the development of modern geometric deep learning, the transformer (Vaswani et al., 2017) rose to become the de-facto standard architecture across a wide range of domains. Transformers are expressive, versatile, and exhibit stable training dynamics. Crucially, they scale well to large systems, mostly thanks to the computation of pairwise interactions through a plain dot product and the existence of highly optimized implementations (Rabe and Staats, 2021; Dao et al., 2022).

Only recently have these two threads been woven together. While different equivariant transformer architectures have been proposed (Fuchs et al., 2020; Jumper et al., 2021; Liao and Smidt, 2022), most involve expensive pairwise interactions that either require restricted receptive fields or limit the scalability to large systems. Brehmer et al. (2023) introduced the Geometric Algebra Transformer (GATr), to the best of our knowledge the first equivariant transformer architecture based purely on dot-product attention. The key enabling idea is the representation of data in a geometric (or Clifford) algebra (GA), having an expressive invariant inner product. While multiple suitable GAs exist, Brehmer et al. (2023) chose a particular projective algebra. Concurrently, the use of GAs in equivariant networks was also explored by Ruhe et al. (2023) in the context of graph neural networks.

[*] Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

In this paper, we discuss and compare different geometric algebras that may be used in an equivariant transformer for 3D modelling problems. We show how geometric data can be represented in Euclidean and conformal GAs in addition to the projective GA used by Brehmer et al. (2023). We then construct new variations of the GATr architectures based on Euclidean and conformal GA representations. These architectures are compared both theoretically and in an experiment. All three prove viable, with unique strengths.

2. Geometric Algebras

Geometric algebra We start with a brief introduction to geometric algebra (GA). An algebra is a vector space that is equipped with an associative bilinear product $V \times V \to V$.

Given a vector space V with a symmetric bilinear inner product, the geometric or Clifford algebra $\mathcal{G}(V)$ can be constructed in the following way: choose an orthogonal basis e_i of the original d-dimensional vector space V. Then, the algebra has 2^d dimensions with a basis given by elements $e_{j_1}e_{j_2}...e_{j_k} =: e_{j_1 j_2 ... j_k}$, with $1 \leq j_1 < j_2 < ... < j_k \leq d$, $0 \leq k \leq d$. For example, for $V = \mathbb{R}^3$, with orthonormal basis e_1, e_2, e_3, a basis for the algebra $\mathcal{G}(\mathbb{R}^3)$ is

$$1, e_1, e_2, e_3, e_{12}, e_{13}, e_{23}, e_{123}. \tag{1}$$

An algebra element spanned by basis elements with k indices is called a k-vector or a vector of *grade* k. A generic element whose basis elements can have varying grades is called a *multivector*. A multivector x can be projected to a k-vector with the grade projection $\langle x \rangle_k$.

The product on the algebra, called the geometric product, is defined to satisfy $e_i e_j = -e_j e_i$ if $i \neq j$ and $e_i e_i = \langle e_i, e_i \rangle$, which by bilinearity and associativity fully specifies the algebra. As an example, for $\mathcal{G}(\mathbb{R}^3)$, we can work out the following product:

$$e_{23}e_{12} = (e_2e_3)(e_1e_2) = (-e_3e_2)(-e_2e_1) = e_3(e_2e_2)e_1 = e_3\langle e_2, e_2\rangle e_1 = e_3e_1 = -e_1e_3 = -e_{13}.$$

GAs are equipped with a linear bijection $\widehat{e_{j_1 j_2 ... j_k}} = (-1)^k e_{j_1 j_2 ... j_k}$, called the grade involution, a linear bijection $\widetilde{e_{j_1 j_2 ... j_k}} = e_{j_k ... j_2 j_1}$, called the reversal, an inner product $\langle x, y \rangle = \langle x\widetilde{y} \rangle_0$, and an inverse $x^{-1} = \widetilde{x}/\langle x, x \rangle$, defined if the denominator is nonzero. From the geometric product, another associative bilinear product can be defined, the wedge product $\wedge$. For k-vector x and l-vector y, this is defined as $x \wedge y = \langle xy \rangle_{k+l}$.

Given a algebra $\mathcal{G}(V)$, there is a group $\mathrm{Pin}(V)$ that is generated by the 1-vectors in the algebra with norm ± 1, and whose group product is the geometric product. A group element $u \in \mathrm{Pin}(V)$ acts on an algebra element $x \in \mathcal{G}(V)$ as $u[x] = uxu^{-1}$ if $u \in \mathrm{Spin}(V)$ and $u[x] = u\widehat{x}u^{-1}$ otherwise. This action is linear, making $\mathcal{G}(V)$ a representation of $\mathrm{Pin}(V)$.

All real inner product spaces are equivalent to a space of the form $\mathbb{R}^{p,q,r}$, with an orthogonal basis with p basis elements that square to $+1$ ($\langle e_i, e_i \rangle = 1$), q that square to -1 and r that square to 0. Similarly, all GAs are equivalent to an an algebra of the form $\mathcal{G}(\mathbb{R}^{p,q,r})$. We'll write $\mathcal{G}(p,q,r) := \mathcal{G}(\mathbb{R}^{p,q,r}), \mathrm{Pin}(p,q,r) := \mathrm{Pin}(\mathbb{R}^{p,q,r})$.

Geometric algebras for 3D space We consider three GAs to model three dimensional geometry. The first is $\mathcal{G}(3,0,0)$, the *Euclidean* GA (EGA), also known as *Vector* GA. The k-vectors have as geometric interpretation respectively: scalar, vectors, pseudovectors, pseudoscalar. A unit vector x in $\mathrm{Pin}(3,0,0)$ represents a mirroring through the plane normal

to x. Combined reflections generate all orthogonal transformation, making the EGA a representation of O(3), or of E(3), invariant to translations.

To represent translation-variant quantities (e.g. positions), we can use $\mathcal{G}(3,0,1)$, the *projective* GA (PGA). Its base vector space $\mathbb{R}^{3,0,1}$ adds to the three Euclidean basis elements, the basis element e_0 which squares to 0 ("homogeneous coordinates"). A unit 1-vector in the PGA is written as $v = \mathbf{n} - \delta e_0$, for a Euclidean unit vector $\mathbf{n} \in \mathbb{R}^3$, and $\delta \in \mathbb{R}$, and represents a plane normal to $\mathbf{n}$, shifted δ from the origin. The 2-vectors represent lines and 3-vectors points (Dorst and De Keninck). The group $\mathrm{Pin}(3,0,1)$ is generated by the unit vectors representing reflections through shifted planes, generating all of E(3), including translations.

The final algebra we consider is $\mathcal{G}(4,1,0)$, the *conformal* GA (CGA, (Dorst et al., 2009)). Its base vector space $\mathbb{R}^{4,0,1}$ adds to the three Euclidean basis elements e_i, the elements e_+ and e_- which square to $+1$ and -1 respectively. Alternatively, it is convenient to choose a non-orthogonal basis $\infty = e_- - e_+$ and $o = (e_- + e_+)/2$, such that $\langle \infty, \infty \rangle = \langle o, o \rangle = 0$ and $\langle \infty, o \rangle = -1$. Planes in the CGA are represented by a 1-vector $\mathbf{n} - \delta\infty$, for a Euclidean unit vector $\mathbf{n}$ and $\delta \in \mathbb{R}$. The Euclidean group E(3) is generated by all such planes, which form a subgroup of $\mathrm{Pin}(4,1,0)$. The CGA contains a point representation by a null 1-vector $p = o + \mathbf{p} + \|\mathbf{p}\|^2\infty/2$, for a Euclidean position vector $\mathbf{p} \in \mathbb{R}^3$.

3. Geometric Algebra Transformer

Given one of the geometric algebras EGA, PGA or CGA, we can construct a geometric algebra transformer (GATr) (Brehmer et al., 2023), respectively E-GATr, P-GATr or C-GATr, that use (many copies of) the algebra as its feature space and is equivariant to E(3) transformations. This requires making the following adjustments to a standard dot-product attention transformer: (1) constrain the linear layers to be equivariant, (2) switch normalization layers and nonlinearities to their equivariant counterparts, (3) in the MLP, let the inputs interact via the geometric product, (4) compute an invariant attention weight between key k and query q via the algebra's inner product $\langle k, q \rangle$.

Any E(3)-equivariant linear map is constructed from linear combinations of grade projections and multiplication with e_0 or ∞ for PGA and CGA, respectively. See Appendix C for a proof. The geometric product is equivariant, because for $u \in \mathrm{Pin}(V), x, y \in \mathcal{G}(V)$, we have that $u[xy] = u[x]u[y]$. The inner product is invariant, because the grade projection and reversal are linear equivariant and the 0-vector, i.e. scalar, is invariant.

The GATr architecture introduced in Brehmer et al. (2023) was based on the PGA, but differs from P-GATr in two key ways: besides the geometric product, the MLP uses another bilinear operation, called the join; and in addition to PGA inner product attention, it uses a map from PGA 3-vectors representing points to CGA 1-vectors representing points and uses the CGA inner product on those. See Brehmer et al. (2023) for details. We refer to this version as improved P-GATr (iP-GATr).

4. Theoretical considerations

Multilinear expressivity To understand better the trade-offs between the GATr variants, we'd like to understand whether they are universal approximators. We will study the slightly simpler question of whether the algebras can express any multilinear map $\mathcal{G}(V)^l \to \mathcal{G}(V)$, a

map from l multivectors to one multivector, linear in each of the inputs. First, we study the case of non-equivariant maps, proven in Appendix B.

Proposition 1 (informal) *The EGA and CGA can express any non-equivariant multilinear map with operations within the algebra, such as geometric products, and constant multivectors. However, due to the degeneracy of its inner product, the PGA can only do so if we also include the join operation.*

For GATr, we are primarily interested in equivariant maps. Here, we don't have a theoretical result, but a conjecture, which we numerically verify up to $l = 4$ (Appendix C).

Conjecture 2 *Let $l \geq 2$. For the EGA and the CGA, and not for the PGA, any* E(3)*-equivariant (resp.* SE(3)*-equivariant) multilinear map $\mathcal{G}(p,q,r)^l \to \mathcal{G}(p,q,r)$ can be constructed out of a combination of the geometric product and* E(3)*-equivariant (resp.* SE(3)*-equivariant) linear maps. For PGA, any* SE(3)*-equivariant multilinear map can be expressed using equivariant linear maps, the geometric product and the join.*

These results suggest that the EGA and CGA, and PGA with the join are sufficiently expressive, while the PGA without the join is not.

Absolute positions The PGA and CGA can represent the absolute position of points: multivectors that are invariant to exactly one rotational SO(3) subgroup of SE(3) – rotations around that point. In contrast, the multivectors of the EGA are invariant to translations, so it can represent directions but not positions. A typical work-around, which we use in our EGA experiments, is to not use absolute positions, but positions relative to some special point, such as the center of mass of a point cloud, which are translation-invariant. This has as downside that the interactions between point-pairs depends on the center of mass. Alternatively, positions can be treated not as generic features in the network, but get special treatment, so that only the position difference between points is used. However, this design decision precludes using efficient dot-product attention in transformers (Brehmer et al., 2023).

Distance-based attention It is desirable when using transformers with geometric systems, that the attention weights between objects can be modulated by their distance. In GATr, the attention logits are the GA inner product between a key and query multivector.

Distance-based attention appears most naturally in the C-GATr architecture. In the CGA, a Euclidean position vector $\mathbf{p} \in \mathbb{R}^3$ is represented as $p = o + \mathbf{p} + \|\mathbf{p}\|^2 \infty/2$, and inner products between points directly compute the Euclidean distance. In the E-GATr, using the positions relative to a center of mass, the inner product of a query consisting of three multivectors $(\|\mathbf{q}\|^2, 2\mathbf{q}, 1)$ and a key $(-1, \mathbf{k}, -\|\mathbf{k}\|^2)$ computes negative squared distance.

However, in P-GATr, dot-product attention cannot compute distances, as we prove in Appendix D. In iP-GATr, this is addressed by computing CGA points from PGA points, and using the CGA inner product in the attention; see Brehmer et al. (2023) for details.

5. Experiments

n-body modelling We first benchmark the GATr variants on the problem of predicting the future position of n gravitational bodies of varying mass and initial position and velocity, initialized concentrated around a varying number of clusters. Figure 1 shows the prediction error as a function of the number of training samples. We find that P-GATr performs poorly due to limited expressivity. E-GATr only performs well with sufficient data, due to the fact that it is only equivariant to rotations around the center of mass. The other GATr variants achieve an excellent performance, outperforming or matching the equivariant baselines (Fuchs et al., 2020; Brandstetter et al., 2022).

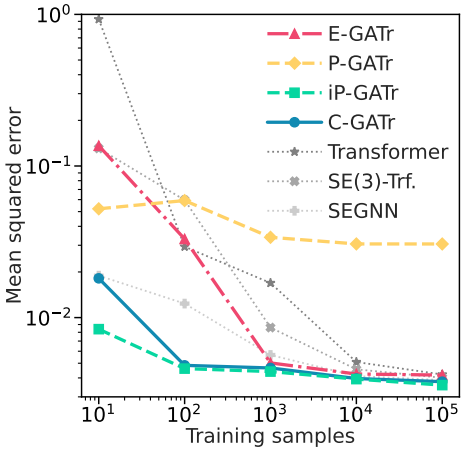

Figure 1: n-body modelling.

Arterial wall-shear-stress estimation Secondly, we test the GATr variants on the problem of predicting the wall shear stress exerted by the blood flow on the arterial wall, using a benchmark dataset proposed by Suk et al. (2022), a challenging problem as there are only 1600 meshes, each begin large ($\approx$ 7000 nodes). We describe the experimental setup in more detail in Appendix F. Table 1 shows our results, the baselines of a regular transformer, PointNet++ (Qi et al., 2017) and GEM-CNN (De Haan et al., 2021). The baseline results are taken from Brehmer et al. (2023) and Suk et al. (2022). All GATrs outperform the baselines. C- & iP-GATr which use distance-aware attention, perform best. We found that C-GATr can suffer from instabilities, see Appendix E.

Method	Approx. error
E-GATr	6.2 %
P-GATr	7.2 %
iP-GATr	**5.5** %
C-GATr	**5.5** %
Transformer	10.5 %
PointNet++	12.3 %
GEM-CNN	7.7 %

Table 1: Arterial experiment.

6. Conclusion

The geometric algebra transformer is a powerful method to build E(3) equivariant models that scale to large problems due to the transformer backend. In this work, we have generalized the original GATr model, which was based on the projective geometic algebra, to the Euclidean and conformal geometric algebra. This involved finding the equivariant linear maps and effective normalization layers. From a theoretical analysis, we found that the Euclidean E-GATr and conformal C-GATr have sufficient expressivity, while the projective P-GATr does not. Addition of the join bilinear, resulting in the improved projective iP-GATr, addresses these issues at the cost of additional complexity. E-GATr can not represent translations or absolute positions, and thus must rely on centering to be E(3) equivariant. This reduces the symmetry group to rotations around the center, and thus sample efficiency. Experimentally, E-GATr has the lowest computational cost, but indeed tends to overfit faster. P-GATr lacks expressivity and doesn't perform well, while the iP-GATr and C-GATr perform best. Of these, C-GATr enjoys the simplicity of just relying on geometric products, while iP-GATr needs the complexity of the join bilinear, as well as a hand-crafted attention method. On the other hand, iP-GATr appears more stable in training than C-GATr.

Acknowledgments

The authors would like to thank Sönke Behrends for his contributions to the implementation on which our experiments were based. We'd like to extend our gratitude to Gabriele Cesa for the helpful discussions.

References

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve E(3) equivariant message passing. In *International Conference on Learning Representations*, 2022. (Cited on page 5)

Johann Brehmer, Pim de Haan, Sönke Behrends, and Taco Cohen. Geometric algebra transformers. In *Advances in Neural Information Processing Systems*, volume 37, 2023. URL https://arxiv.org/abs/2305.18415. (Cited on pages 1, 2, 3, 4, 5, 11, 13, 14, and 15)

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. 2021. (Cited on page 1)

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016. (Cited on page 1)

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022. (Cited on page 1)

Pim De Haan, Maurice Weiler, Taco Cohen, and Max Welling. Gauge equivariant mesh CNNs: Anisotropic convolutions on geometric graphs. *International Conference on Learning Representations*, 2021. (Cited on page 5)

Leo Dorst and Steven De Keninck. A guided tour to the plane-based geometric algebra PGA. https://bivector.net/PGA4CS.pdf. URL https://bivector.net/PGA4CS.pdf. Accessed: 2023-4-28. (Cited on pages 3 and 9)

Leo Dorst, Stephen Mann, and Daniel Fontijne. *Geometric Algebra for Computer Science*. 2009. (Cited on page 3)

Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. April 2021. URL http://arxiv.org/abs/2104.09459. (Cited on page 9)

Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. SE(3)-Transformers: 3D Roto-Translation equivariant attention networks. In *Advances in Neural Information Processing Systems*, 2020. (Cited on pages 1 and 5)

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. (Cited on page 1)

Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. *arXiv:2206.11990*, 2022. (Cited on page 1)

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017. (Cited on page 5)

Markus N Rabe and Charles Staats. Self-attention does not need $O(n^2)$ memory. *arXiv:2112.05682*, 2021. (Cited on page 1)

David Ruhe, Johannes Brandstetter, and Patrick Forré. Clifford group equivariant neural networks. *arXiv:2305.11141*, 2023. (Cited on page 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. *arXiv:2212.05023*, 2022. (Cited on pages 5, 14, and 15)

Twemoji. Open source emojis. URL https://twemoji.twitter.com. Accessed: 2023-11-3. (Cited on page 8)

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. (Cited on page 1)

	E-GATr	P-GATr	iP-GATr	C-GATr
Simplicity	🐃🐃🐃🐃	🐃🐃🐃	🐃	🐃🐃🐃
Representational richness	🐃	🐃🐃🐃	🐃🐃🐃	🐃🐃🐃🐃
Expressivity	🐃🐃🐃🐃	🐃	🐃🐃🐃🐃	🐃🐃🐃🐃
Memory efficiency	🐃🐃🐃🐃	🐃🐃🐃	🐃🐃🐃	🐃
Training stability	🐃🐃🐃🐃	🐃🐃🐃🐃	🐃🐃🐃🐃	🐃
Experiment performance	🐃🐃🐃	🐃	🐃🐃🐃🐃	🐃🐃🐃🐃

Table 2: GATr variants ranked from 🐃🐃🐃🐃 (best) to 🐃 (worst) along theoretical qualities (top) and empirical observations (bottom).[2]

Appendix A. Scorecard

In Table 2, we provide a subjective scoring of the nuanced trade-off between the variants.

Appendix B. Constructing generic multilinear maps

Proposition 3 *Let $l \geq 1$.*

(1) If and only if the inner product of $\mathbb{R}^{p,q,r}$ is non-degenerate ($r = 0$), any multilinear map $\mathcal{G}(p,q,r)^l \to \mathcal{G}(p,q,r)$ can be constructed from addition, geometric products, grade projections and constant multivectors.

(2) Furthermore, any multilinear map $\mathcal{G}(p,0,1)^l \to \mathcal{G}(p,0,1)$ can be constructed from addition, geometric products, the join bilinear, grade projections and constant multivectors.

Proof *Proof of (1), "⇒":* First, let $r = 0$. Then let e_i be an orthogonal basis of $\mathbb{R}^{p,q,0}$ where each e_i squares to ± 1. This gives a basis $e_\mathbf{i}$, with multi-index $\mathbf{i} \in 2^{p+q}$, of the algebra $\mathcal{G}(p,q,0)$. This basis is also orthogonal and each element $e_{i_1 i_2 \ldots i_k}$ squares to $\langle e_{i_1 i_2 \ldots i_k}, e_{i_1 i_2 \ldots i_k} \rangle = \langle e_{i_1 i_2 \ldots i_k} \widetilde{e_{i_1 i_2 \ldots i_k}} \rangle_0 = e_{i_1} e_{i_2} \ldots e_{i_k} e_{i_k} \ldots e_{i_2} e_{i_1} = \prod_k \langle e_k, e_k \rangle = \pm 1$.

Now, let $\phi : \mathcal{G}(p,q,0) \to \mathcal{G}(p,q,0)$ be any linear map. For each basis element of the algebra, let $x_\mathbf{i} := \phi(e_\mathbf{i}) / \langle e_\mathbf{i}, e_\mathbf{i} \rangle$. Then ϕ can then be written as:

$$\psi(w) = \sum_{\mathbf{i} \in 2^{p+q+r}} x_\mathbf{i} \langle w \, \widetilde{e_\mathbf{i}} \rangle_0$$

It is easy to see that for any basis element $e_\mathbf{i}$, $\phi(e_\mathbf{i}) = \psi(e_\mathbf{i})$, hence the linear maps coincide.

For a multilinear map $\phi : \mathcal{G}(p,q,0)^l \to \mathcal{G}(p,q,0)$, a similar construction can be made:

$$\phi(w_1, \ldots, w_l) = \sum_{\mathbf{i}_1 \in 2^{p+q+r}} \ldots \sum_{\mathbf{i}_l \in 2^{p+q+r}} x_{\mathbf{i}_1, \ldots, \mathbf{i}_l} \langle w_1 \, \widetilde{e_{\mathbf{i}_1}} \rangle_0 \ldots \langle w_l \, \widetilde{e_{\mathbf{i}_l}} \rangle_0$$

$$\text{with} \quad x_{\mathbf{i}_1, \ldots, \mathbf{i}_l} = \frac{\phi(e_{\mathbf{i}_1}, \ldots, e_{\mathbf{i}_l})}{\langle e_{\mathbf{i}_1}, e_{\mathbf{i}_1} \rangle \ldots \langle e_{\mathbf{i}_l}, e_{\mathbf{i}_l} \rangle}$$

2. Symbol by Twemoji, used under a CC BY-4.0 license.

Proof of (1), "⇐": Let $r > 0$. Let $e_0 \in \mathbb{R}^{p,q,r}$ denote a nonzero radical vector, meaning that for all $x \in \mathbb{R}^{p,q,r}$, $\langle e_0, x\rangle = 0$. Consider the multilinear map $\phi : \mathcal{G}(p,q,r)^l \to \mathcal{G}(p,q,r)$ sending input $(e_0, ..., e_0) \mapsto 1$ and all other inputs to 0. This map can not be constructed from within the algebra. To see this, consider any nonzero k-vector $e_0 \wedge y$ for a $(k-1)$-vector y. The only way of mapping $e_0 \wedge y$ to a scalar involves multipication with e_0, which results in a zero scalar component.

Proof of (2): Now consider the projective algebra $\mathcal{G}(p,0,1)$ equipped with the join $\vee$, a bilinear operation $\mathcal{G}(p,0,1) \times \mathcal{G}(p,0,1) \to \mathcal{G}(p,0,1)$ mapping algebra basis elements $e_\mathbf{i} \vee e_\mathbf{j}$ to $\pm e_\mathbf{k}$, where $\mathbf{k}$ contains all indices that occur in both $\mathbf{i}$ and $\mathbf{j}$, as long as all $p+1$ indices are present as at least once in either $\mathbf{i}$ or $\mathbf{j}$. Otherwise, $e_\mathbf{i} \vee e_\mathbf{j} = 0$. See Dorst and De Keninck for details. In particular, the join satisfies $e_{012...p} \vee 1 = 1$.

With the join in hand, any linear map $\phi : \mathcal{G}(p,0,1) \to \mathcal{G}(p,0,1)$ can be written as:

$$\psi(w) = \sum_{\mathbf{i} \in 2^{p+1}} x_\mathbf{i} \langle (w \wedge e_{\setminus \mathbf{i}}) \vee 1 \rangle_0$$

where $x_\mathbf{i} := \phi(e_\mathbf{i})$ and $e_{\setminus \mathbf{i}}$ contains all indices absent in $\mathbf{i}$, in an order such that $e_\mathbf{i} \wedge e_{\setminus \mathbf{i}} = e_{012...p}$. For any basis element $e_\mathbf{j}$, $\langle (e_\mathbf{j} \wedge e_{\setminus \mathbf{i}}) \vee 1 \rangle_0 = 1$ if $\mathbf{j} = \mathbf{i}$ and 0 otherwise, because if $\mathbf{j}$ lacks any index in $\mathbf{i}$, the join yields a zero, and if it $\mathbf{j}$ has any indices not in $\mathbf{i}$, the join results in a non-scalar, which becomes zero with the grade projection. Therefore, $\psi(e_\mathbf{i}) = \phi(e_\mathbf{i})$ for all basis elements $e_\mathbf{i}$, and the linear maps are equal. As before, this construction easily generalizes to multi-linear maps. ■

Appendix C. Numerically computing equivariant multilinear maps

C.1. Lie group equivariance constraint solving via Lie algebras

First, let's discuss in generality how to solve group equivariance constraints via the Lie algebra, akin to Finzi et al. (2021).

Let G be a Lie group, $\mathfrak{g}$ be its algebra. Let $\exp : \mathfrak{g} \to G$ be the Lie group exponential map.

A group representation (ρ, V) induces a Lie algebra representation: $d\rho : \mathfrak{g} \to \mathfrak{gl}(V)$, linearly sending $X \in \mathfrak{g}$ to a linear map $d\rho(X) : V \to V$, satisfying $\rho(\exp(X)) = \exp(d\rho(X))$, where the latter exp is the matrix exponential.

Given a real Lie algebra representation (ρ, V), there is a dual representation (ρ^*, V^*) satisfying $\rho^*(g) = \rho(g^{-1})^T$. It is easy to see that $d\rho^*(X) = -d\rho(X)^T$.

For two group representations (ρ_1, V_1) and (ρ_2, V_2), there is a tensor representation $(\rho_1 \otimes \rho_2, V_1 \otimes V_2)$ with Lie algebra representation $d(\rho_1 \otimes \rho_2) = 1_{V_1} \otimes d\rho_2 + d\rho_1 \otimes 1_{V_2}$.

(ρ_1, V_1) and (ρ_2, V_2), a linear map $\phi : V_1 \to V_2$ is equivariant if and only if ϕ is invariant to the group representation $\rho_2 \otimes \rho_1^*$, when flattening $\text{vec}(\phi) \in V_2 \otimes V_1^*$: for all $g \in G$,

$$\rho_2(g)\phi = \phi\rho_1(g) \iff (\rho_2 \otimes \rho_1^*)(g)\text{vec}(\phi) = \text{vec}(\phi)$$

Any Lie group G is equal to a semi-direct product $G^0 \rtimes D$, for $G^0 \subseteq G$ the subgroup connected to the identity and D a discrete group. Let B be a set of basis elements of the Lie algebra. Then $\exp(\text{span}(B)) = G^0$.

First, consider a connected Lie group G^0, and a basis B of the Lie algebra, and a representation (ρ, V). Then

$$\begin{aligned} & \forall g \in G^0,\ \rho(g)v = v \\ \iff \quad & \forall X \in \mathfrak{g},\ \rho(\exp(X))v = v \\ \iff \quad & \forall X \in \mathfrak{g},\ \exp(d\rho(X))v = v \\ \iff \quad & \forall X \in \mathfrak{g},\ d\rho(X)v = 0 \\ \iff \quad & \forall X \in B,\ d\rho(X)v = 0 \end{aligned}$$

where in for the final step, we note that $d\rho$ is linear, so linearly dependent algebra vectors generate linearly dependent csontraints, and just constraining by a basis of the algebra suffices.

To test invariance to a non-connected Lie group, we need to additionally constrain for the discrete group D, generated by subgroup D', leading to:

$$\forall g \in G\, \rho(g)v = v \iff \begin{cases} d\rho(X)v = 0 & \forall X \in B \\ (\rho(g) - 1_V)v = 0 & \forall g \in D' \end{cases}$$

If V is d-dimensional There are thus $|B| + |D'|$ $d \times d$ matrices and v needs to be in the null space of each of these, or equivalently in the null space of the concatenated $((|B| + |D'|)d) \times d$ matrix. This can be done numerically via e.g. `scipy.linalg.nullspace`. When ρ can be decomposed into subrepresentations $(\rho_a \oplus \rho_b, V_a \oplus V_b)$, the invariant vectors can be found separately, making computing the null space more efficient.

Combining the framing of equivariance as invariance, and finding invariant vectors via a null space, we can find the linear equivariant maps $\phi : V_1 \to V_2$ by finding the nullspace of:

$$\forall g \in G\, \rho_2(g)\phi = \phi\rho_1(g) \iff \begin{cases} (d\rho_2 \otimes 1_{V_1^*} - 1_{V_2} \otimes d\rho_1^T)(X)\mathrm{vec}\phi = 0 & \forall X \in B \\ ((\rho_2 \otimes \rho_1^*)(g) - 1_V)\mathrm{vec}\phi = 0 & \forall g \in D' \end{cases}$$

To find equivariant multilinear maps $\phi : V_{i_1} \otimes V_{i_2} \otimes ... \otimes V_{i_l} \to V$, we simply set $\rho_1 = \rho_{i_1} \otimes \rho_{i_2} \otimes ... \otimes \rho_{i_l}$, with $d\rho_1 = d\rho_{i_1} \otimes 1_{V_{i_2}} \otimes ... + 1_{V_{i_1}} \otimes d\rho_{i_2} \otimes ... + ...$

C.2. GA equivariance solving of linear maps

For the GA, we'll consider $V = \mathcal{G}(p, q, r)$ and $\rho(u)(x) = u\widehat{x}u^{-1}$. Define $d\rho(X)(v) = Xv - vX$. Any GA has the exponential map endomorphism, defined through the Taylor series:

$$\exp : \mathcal{G}(p, q, r) \to \mathcal{G}(p, q, r) : x \mapsto 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + ...$$

EGA Now, for the EGA, the bivectors $\mathcal{G}(3, 0, 0)_2$ are the Lie algebra $\mathfrak{spin}(3, 0, 0)$ of the connected Lie group $\mathrm{Spin}(3, 0, 0)$ of even number of reflections. The Lie group exponential map is the GA exponential map. The entire Pin group decomposes as $\mathrm{Pin}(3, 0, 0) = \mathrm{Spin}(3, 0, 0) \rtimes \{1, e_1\}$. The bivectors have a basis $\mathfrak{spin}(3, 0, 0) = \mathcal{G}(3, 0, 0)_2 = \mathrm{span}(e_{12}, e_{23}, e_{13})$. Therefore,

a linear map $\phi : \mathcal{G}(3,0,0) \to \mathcal{G}(3,0,0)$ is equivariant to $\mathrm{Pin}(3,0,0)$, and hence to $\mathrm{O}(3)$, which it doubly covers, and $\mathrm{E}(3)$ with trivial action under translation, if and only if:

$$\begin{cases} (d\rho \otimes 1 - 1 \otimes d\rho^T)(X)\mathrm{vec}\phi = 0 \quad \forall X \in \{e_{12}, e_{23}, e_{13}\} \\ ((\rho \otimes \rho^*)(e_1) - 1_V)\mathrm{vec}\phi = 0 \end{cases}$$

Studying the nullspace, we find that all equivariant linear maps can be written as linear combinations of grade projections, giving 4 independent maps:

$$\phi : \mathcal{G}(3,0,0) \to \mathcal{G}(3,0,0) : x \mapsto \sum_{k=0}^{3} \alpha_k \langle x \rangle_k$$

PGA For the PGA, similarly, $\mathrm{Pin}(3,0,1)$ doubly covers $\mathrm{E}(3)$. The group $\mathrm{Spin}(3,0,1)$ is its connected subgroup, whose algebra are the bivectors, and the Pin group decomposes as the Spin group and a mirroring. A linear map $\phi : \mathcal{G}(3,0,1) \to \mathcal{G}(3,0,1)$ is therefore equivariant to $\mathrm{Pin}(3,0,1)$, and hence $\mathrm{E}(3)$, if and only if:

$$\begin{cases} (d\rho \otimes 1 - 1 \otimes d\rho^T)(X)\mathrm{vec}\phi = 0 \quad \forall X \in \{e_{12}, e_{23}, e_{13}, e_{01}, e_{02}, e_{03}\} \\ ((\rho \otimes \rho^*)(e_1) - 1_V)\mathrm{vec}\phi = 0 \end{cases}$$

Studying the nullspace, we find that all equivariant linear maps can be written as linear combinations of grade projections and multiplications with e_0, leading to 9 independent maps:

$$\phi : \mathcal{G}(3,0,1) \to \mathcal{G}(3,0,1) : x \mapsto \sum_{k=0}^{4} \alpha_k \langle x \rangle_k + \sum_{k=1}^{4} \beta_k \langle e_0 x \rangle_k$$

This result is in accordance with what was shown analytically in Brehmer et al. (2023).

CGA Let $\iota : \mathcal{G}(3,0,1) \to \mathcal{G}(4,0,1)$ be the algebra homomorphism with $\iota(e_i) = e_i, \iota(e_0) = \infty$. For the CGA, $E(3)$ is doubly covered by the subgroup $\iota(\mathrm{Pin}(3,0,1))$ of $\mathrm{Pin}(4,1,0)$, hence a linear map $\phi : \mathcal{G}(4,1,0) \to \mathcal{G}(4,1,0)$ is equivariant to $\iota(\mathrm{Pin}(3,0,1))$, and hence $\mathrm{E}(3)$, if and only if:

$$\begin{cases} (d\rho \otimes 1 - 1 \otimes d\rho^T)(X)\mathrm{vec}\phi = 0 \quad \forall X \in \{e_{12}, e_{23}, e_{13}, e_{\infty 1}, e_{\infty 2}, e_{\infty 3}\} \\ ((\rho \otimes \rho^*)(e_1) - 1_V)\mathrm{vec}\phi = 0 \end{cases}$$

Studying the nullspace, we find that all equivariant linear maps can be written as linear combinations of grade projections and multiplications with ∞, giving 20 independent maps

in total:

$$\begin{aligned}\phi : \mathcal{G}(4,1,0) \to & \mathcal{G}(4,1,0) \\ x \mapsto & \sum_{k=0}^{5} \alpha_k \langle x \rangle_k \\ & + \sum_{k=1}^{5} \beta_k \langle \infty \langle x \rangle_k \rangle_{k-1} \\ & + \sum_{k=0}^{4} \gamma_k \langle \infty \langle x \rangle_k \rangle_{k+1} \\ & + \sum_{k=1}^{4} \delta_k \infty \langle \infty \langle x \rangle_k \rangle_{k-1}\end{aligned}$$

SE(3) **equivariance** To consider SE(3) equivariance, we just have to be equivariant tot the connected part rotational of the Lie group, so remove the mirror constraint in the above equations. For the EGA, PGA and CGA, we find numerically that the SE(3)-equivariant maps are the same as the E(3)-equivariant linear maps, but possibly combined with multiplication with the pseudoscalar: e_{123} for the EGA, e_{0123} for the PGA and $e_{123} \wedge o \wedge \infty$ for the CGA. This is because the pseudoscalar is an invariant, up to a sign flip due to mirroring, thus SE(3) invariant.

C.3. Multilinear equivariant map solving

To find multilinear equivariant maps efficiently, we found it necessary to separate out the grades. For any geometric algebra, the $\mathrm{Pin}(p,q,r)$ representation decomposes into sum of a representation $(\rho_k, \mathcal{G}(p,q,r)^k)$ of k-vectors, for each grade k. Then we use the above procedure to find the equivariant multilinear maps $\phi : \mathcal{G}(p,q,r)^{i_1} \otimes \mathcal{G}(p,q,r)^{i_2} \otimes ... \otimes \mathcal{G}(p,q,r)^{i_l} \to \mathcal{G}(p,q,r)^o$, taking as inputs an i_1-vector, and i_2-vector, ..., and an i_l-vector and outputting an o-vector.

C.4. Numerically testing expressivity

In the above subsections, we show how one can compute all equivariant multilinear maps for a given algebra. In the main paper, we stated the following conjecture:

Conjecture 4 *Let $l \geq 2$. For the EGA and the CGA, and not for the PGA, any* E(3)*-equivariant (resp.* SE(3)*-equivariant) multilinear map $\mathcal{G}(p,q,r)^l \to \mathcal{G}(p,q,r)$ can be constructed out of a combination of the geometric product and* E(3)*-equivariant (resp.* SE(3)*-equivariant) linear maps. For PGA, any* SE(3)*-equivariant multilinear map can be expressed using equivariant linear maps, the geometric product and the join.*

To test this, we explicitly construct all linear maps via the algebra. Let $\phi^\alpha_{\mu\nu}$ be a basis for the linear equivariant maps of an algebra, so that for each α, $y_a = \sum_b \phi^\alpha_{ab} x_b$ is an equivariant linear map, where roman indices enumerate multivector indices. Also, let Φ^β_{abc} be a basis for the bilinears in the algebra, so that for each β, $z_a = \sum_{bc} \Phi^\beta_{abc} x_b y_c$ is a bilinear. For most

algebras, we'll just consider the geometric product, but for the PGA, we can also consider the join, which is only SE(3)-equivariant (Brehmer et al., 2023, Prop 7). Then, for example, for $l = 2$, all bilinear maps constructable for two inputs x^1, x^2 from the linears and bilinears are:

$$\sum_{bc} \Omega_{abc}^{\sigma\alpha\beta\gamma\delta} x_b^1 x_c^2 = \sum_{bcdef} \phi_{ab}^{\alpha} \Phi_{bcd}^{\beta} \left(\phi_{ce}^{\gamma} x_e^{\sigma_1}\right) \left(\phi_{df}^{\delta} x_f^{\sigma_2}\right)$$

where $\sigma \in S_2$ is a permutation over the two inputs. This approach can be recursively applied to construct any multilinear map from the bilinears and linears. As the algebra is not commutative, we need to take care to consider all permutations of the inputs. For computational efficiency to soften the growth in the number of Greek basis indices, during the reduction for multilinear maps, we apply a singluar value decomposition of the basis of maps, re-express the basis in the smallest number of basis maps.

With this strategy, we were able to verify the above conjecture for $2 \leq l \leq 4$.

Appendix D. Distance-based attention

To show that P-GATr, which uses the PGA inner product as attention, can not have attention weights depend on distance, we prove that taking the inner product of any transformation of point representations, will be constant in the position of the points. Hence, it can not compute distances.

Proposition 5 *Let $\omega : \mathbb{R}^3 \to \mathcal{G}(3,0,1)$, $x \mapsto x_1 e_{032} + x_2 e_{013} + x_3 e_{021} + e_{123}$ be the point representation of the PGA. For all Spin-equivariant maps $\phi, \psi : \mathcal{G}(3,0,1) \to \mathcal{G}(3,0,1)$, for positions $x, y \in \mathbb{R}^3$, the inner product $\langle \phi(\omega(x)), \psi(\omega(y)) \rangle$ is constant in both x and y.*

Proof The inner product in the PGA is equal to the Euclidean inner product on the Euclidean subalgebra $\mathcal{G}(3,0,0)$ (given a basis, this is the subalgebra spanned by elements e_1, e_2, e_3, but not e_0), ignoring the basis elements containing e_0. Translations act invariantly on the the Euclidean subalgebra. Therefore, for any $v \in \mathcal{G}(3,0,1)$, if we consider the map $\mathbb{R}^3 \to \mathbb{R} : x \mapsto \langle \phi(\omega(x)), v \rangle$, this map is invariant to translations, and thus constant. Filling in $v = \phi(\omega(y))$ proves constancy of $\langle \phi(\omega(x)), \psi(\omega(y)) \rangle$ in x. Constancy in y is shown similarly. ■

Appendix E. Normalization and stability

In GATr, as in typical transformers, LayerNorm is used, which normalizes a collection of n multivector channels jointly. The obvious equivariant interpretation of LayerNorm in GATr would be:

$$\mathcal{G}(p,q,r)^n \to \mathcal{G}(p,q,r) : x \mapsto \frac{x}{\sqrt{\frac{1}{n}\sum_{i=1}^n \langle x^i, x^i \rangle + \epsilon}}$$

leaving out the shift to 0 mean used typically in normalization to ensure equivariance. This approach works when $q = r = 0$, as then the inner product is directly related to the magnitued of the multivector coefficients, which the normalization layer is designed to keep controlled. However, for the PGA, with $r = 1$, the 8 dimensions containing e_0 do not contribute to the

inner product, making their magnitudes no longer well-controlled. We found a reasonably high magnitude of $\epsilon = 0.01$ to suffice to stabilize training. For the CGA, with $q = 1$, the situation is worse. Firstly, as the inner products can be negative, the channels can cancel each other out. In a first attempt to address this, we add the absolute value around the inner product:

$$\mathcal{G}(p,q,r)^n \to \mathcal{G}(p,q,r) : x \mapsto \frac{x}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} |\langle x^i, x^i \rangle| + \epsilon}}$$

However, also within one multivector some dimension contribute negatively to the inner product and, for example, a scalar and pseudoscalar can cancel out to give a 0-norm (null) multivector. The coefficients of such a multivector grow by $1/\sqrt{\epsilon}$ with each normalization layer. Setting $\epsilon = 1$ stabilized training, but made the models achieve poor training losses. We found it beneficial to use the following norm in the CGA, which applies the absolute value around each multivector grade separately:

$$\mathcal{G}(p,q,r)^n \to \mathcal{G}(p,q,r) : x \mapsto \frac{x}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} \sum_{k=0}^{5} |\langle \langle x^i \rangle_k, \langle x^i \rangle_k | \rangle + \epsilon}}$$

This approach mostly addressed stability concerns. However, due to the fact that we still can't fully control the magnitude of the coefficients, we still found it necessary to train C-GATr on `float32`, whereas the other GATr variants trained well on `bfloat16`.

Appendix F. Experiment details

n-body modelling dataset We create an n-body modelling dataset, in which the task is to predict the final positions of a number of objects that interact under Newtonian gravity given their initial positions, velocities, and velocities. The dataset is created like the n-body dataset described in Brehmer et al. (2023), with one exception: rather than a single cluster of bodies, we create a variable number of clusters, each with a variable number of bodies, such that the total number of bodies in each sample is 16. This makes the problem more challenging. Each cluster is generated as described in Brehmer et al. (2023), and the clusters have locations and overall velocities relative to each other sampled from Gaussian distributions.

Arterial wall-shear-stress dataset We use the dataset of human arteries with computed wall shear stress by Suk et al. (2022). We use the single-artery version and focus on the non-canonicalized version with randomly rotated arteries. There are 1600 training meshes, 200 validation meshes, and 200 evaluation meshes, each with around 7000 nodes.

Models and training Our GATr variants are discussed in the main paper. We mostly follow the choices used in Brehmer et al. (2023), except for the choice of algebra, attention, and normalization layers. For the linear maps, we evaluated two initialization methods: initialize all basis maps with a Kaiming-like scheme, or initialize the linear maps to be the identity on the algebra, and Kaiming-like in the channels. For iP-GATr and P-GATr, we found that the former worked best, for C-GATr we found the latter to work best and for E-GATr we found no difference.

We choose model and training hyperparameters as in Brehmer et al. (2023), except that for the n-body experiments, we use wider and deeper architectures with 20 transformer blocks, 32 multivector channels, and 128 scalar channels.

Baselines For the n-body modelling experiment, we run Transformer, $SE(3)$-Transformer, and SEGNN experiments, with hyperparameters as discussed in Brehmer et al. (2023).

For the artery experiments, baseline results are taken from Brehmer et al. (2023) and Suk et al. (2022).