VINE-GATR: SCALING GEOMETRIC ALGEBRA TRANSFORMERS WITH VIRTUAL NODE EMBEDDINGS

Julian Suk*
University of Twente, Enschede
Qualcomm AI Research†, Amsterdam

Thomas Hehn, Arash Behboodi, Gabriele Cesa
Qualcomm AI Research†, Amsterdam

ABSTRACT

Equivariant neural networks can effectively model physical systems by naturally handling the underlying geometric quantities and preserving their symmetries, but scaling them to large geometric data remains challenging. Naive downsampling typically disrupts features' transformation laws, limiting their applicability in large scale settings. In this work, we propose a scalable equivariant transformer that efficiently processes geometric data in a coarse-grained latent space while preserving $\mathrm{E}(3)$ symmetries of the problem. In particular, by building on the Geometric Algebra Transformer (GATr) and PerceiverIO architectures, our method learns equivariant latent tokens which allow us to decouple the processing complexity from the input data representation while maintaining global equivariance.

1 INTRODUCTION

Machine learning approaches that adhere to the laws of physics have become an important tool in analyzing data in natural sciences (Raissi et al., 2019). Equivariant neural network architectures (Cohen & Welling, 2016; Weiler et al., 2021; Bronstein et al., 2021) are a prime example of such approaches, as they guarantee that the network output transforms with the input according to the symmetries of the underlying physics. The design principle of equivariant models is to incorporate the prior knowledge about the geometry and the symmetries of the problem directly into the architecture, such that the model does not need to learn them from data during training. As a result, the equivariant design can improve the model's reliability and is especially effective in data-scarce regimes, for example when gathering or even simulating abundant data is too expensive or prohibitive. This is a common scenario in many disciplines in science, where it is crucial to analyze large quantities of geometric data - typically in the form of graphs, meshes or point clouds - with very little training data available. Equivarant models are particularly useful where point clouds cannot be canonically aligned without additional modeling assumptions (for example in cortical surface analysis). Other examples include weather prediction and climate modeling (Andrychowicz et al., 2023; Nguyen et al., 2023; Lam et al., 2022), material design for high-temperature superconductivity (Choudhary & Garrity, 2022; Chen et al., 2024) or fusion power plants (Spangher et al., 2024), analyzing Large Hadron Collider (LHC) data (Plehn et al., 2022; Brehmer et al., 2024), catalyst design (Goldsmith et al., 2018) and even large scale medical data analysis (Arzani et al., 2022; Suk et al., 2024; Dahan et al., 2024). While practitioners in each discipline developed powerful deep learning solutions, leveraging also domain knowledge, these are often tailored to the underlying domain and data representation, and scalability remains a major challenge in many applications. A general strategy to handle large-scale geometric input data, which proved to be effective across many scenarios, is to transition from the fine-grained data space to a coarse-grained latent one, which is more practical for performing computations. Ideally, the coarser latent structure allows computational resources to focus on the most informative parts of the high-resolution data, avoiding the excessive costs associated with modeling the fine patterns at full resolution and, therefore, improving scalability. Similar ideas underpin many successful solutions, but it is not straightforward to implement this within an equivariant design. Indeed, the strict constraints enforced by equivariance require the ex-

*Work done during an internship at Qualcomm AI Research, Amsterdam.
†Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

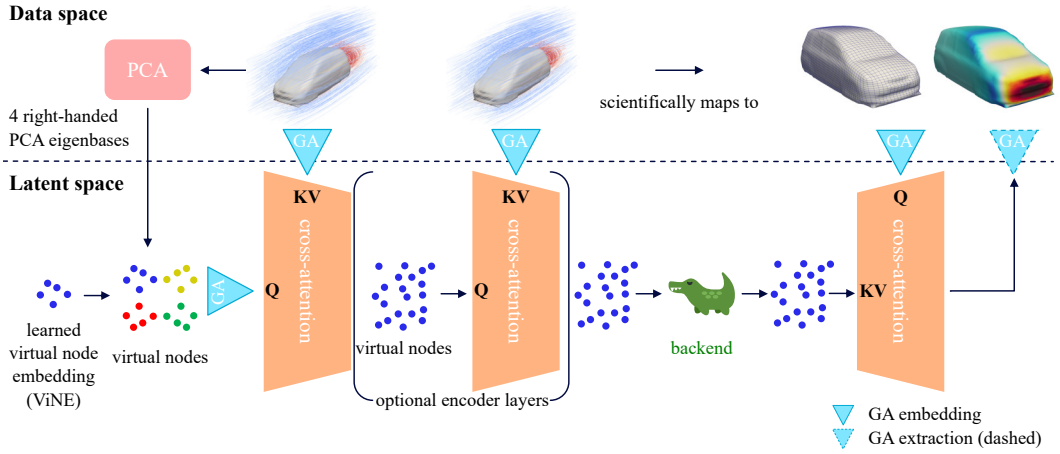

Figure 1: **ViNE-GATr:** input tokens in *data space* are processed into geometric features via PCA. These augment $V/4$ invariant coordinates to construct V virtual nodes (*latent space*). The encoder includes one (or more) cross-attention modules, allowing virtual nodes to query the high-resolution inputs. The resulting latent features are processed by a standard GATr backend. Finally, an output neural field is generated via cross-attention by querying virtual nodes at each output token (or spatial location).

act conservation of the data's transformation law and naive approaches to downsampling are known to be problematic (Zhang, 2019). Preserving the right geometric structure and the input symmetries during down-sampling poses non-trivial challenges, which are subject of many previous works such as (Xu et al., 2021; Chaman & Dokmanic, 2021; Rojas-Gomez et al., 2022; Rahman & Yeh, 2025).

Alkin et al. (2024) proposed a flexible framework addressing similar problems by elegantly decoupling the modeling of dynamics within latent tokens from the input data representation, substantially improving scalability. The method builds on PerceiverIO (Jaegle et al.) to construct a hourglass architecture that learns an independent set of latent query tokens to process the input data efficiently in three stages: first, the latent query tokens extract relevant features from the input via cross-attention, second, the latent tokens are processed by a transformer architecture and, finally, an output conditional neural field is reconstructed at the output tokens, which aggregate features via cross-attention on the latent space. Unfortunately, incorporating equivariance in this framework is still not straightforward since latent query tokens are inherently invariant, preventing them from capturing geometric information from the input keys equivariantly through the simple inner products in cross-attention layers. Hence, in an equivariant design, it is desirable for the latent tokens to transform equivariantly too; see Sec. 3. In this paper, building on the Geometric Algebra Transformer (GATr) framework from Brehmer et al. (2023), we introduce ViNE-GATr, a scalable transformer architecture that preserves the $\mathrm{E}(3)$-symmetries in the data space while efficiently processing it in a latent token space.

2 Background: Geometric Algebra Transformers

Geometric Algebra Transformers (GATr) (Brehmer et al., 2023) is a recent architecture which achieves equivariance to the group $\mathrm{E}(3)$ of isometries of the 3-dimensional Euclidean space, i.e. translations, rotations and mirroring. GATr leverages the *projective geometric algebra* (or *Clifford algebra*) (PGA) $\boldsymbol{G}(3,0,1)$ to represent its input, output and intermediate activations as (multiple) 16-dimensional feature vectors - the so-called *multivectors* - which encode geometric objects such as scalars, points, lines or planes as well as certain geometric operators like rotations or translations.

To better understand how PGA is used to represent 3D geometry, note that by including a 4-th homogeneous coordinate e_0 to the standard basis (e_1, e_2, e_3) of $\mathbb{R}^3$, one can model translations as linear operations too. Then, by introducing a certain associative, non-commutative *geometric product*

between these vectors[1], one generates the 16-dimensional vector space of multivectors:

$$x = (\underbrace{x_s}_{\text{scalar}}, \underbrace{x_0, x_1, x_2, x_3}_{\text{vectors}}, \underbrace{x_{01}, x_{02}, x_{03}, x_{12}, x_{13}, x_{23}}_{\text{bi-vectors}}, \underbrace{x_{012}, x_{013}, x_{023}, x_{123}}_{\text{tri-vectors}}, \underbrace{x_{0123}}_{\text{pseudo-scalar}}) \in \boldsymbol{G}(3, 0, 1)$$

The geometric product is a bilinear operator which generalizes well known operations like the inner product or the cross product between vectors, but also implements other geometric transformations (e.g. combine two multivectors representing a translation and a point to obtain the translated point, expressed as another multivector). This motivates the use of the geometric algebra as a principled and practical tool for geometrical reasoning. We refer to Apx. B and Brehmer et al. (2023) for more details about PGA and the mapping of geometric objects and operators to multivectors.

Like typical equivariant networks, multivector features carry an action of the equivariance group $\mathrm{E}(3)$ and GATr includes layers which map between these features in an equivariant way, in particular: linear layers, bilinear layers (which, loosely speaking, encode the geometric product between multivectors) and the attention layer, as well as other non-linear layers. Finally, GATr has already been applied to a wide range of scientific domains (Suk et al., 2024; Hehn et al., 2025; Brehmer et al., 2024). We believe its versatility makes GATr a good architecture for our problem.

3 E(3)-EQUIVARIANT VIRTUAL NODES EMBEDDINGS (VINE)

In a *generic equivariant scenario*, we assume the input tokens jointly transform according to a certain group G (e.g $\mathrm{E}(3)$) which models the symmetries of the task: $g \in G : \{n_j\}_{j=1}^N \mapsto \{g.n_j\}_{j=1}^N$. The typical cross-attention mechanism leverages the inner product $\langle \cdot, \cdot \rangle$ between key and query vectors. As a result, when using fixed (hence, invariant) query tokens but the input transforms under $g \in G$, the attention scores are given by

$$\langle \boldsymbol{q}(v_i), \boldsymbol{k}(g.n_j) \rangle = \langle \boldsymbol{q}(v_i), \rho(g)\boldsymbol{k}(n_j) \rangle \qquad \neq \langle \boldsymbol{q}(v_i), \boldsymbol{k}(n_j) \rangle \qquad \forall g \in G \tag{1}$$

where ρ is a representation of G modeling its action on the key feature space and $\boldsymbol{k}$ consists of primitives equivariant to the actions of G. However, cross-attention is G-equivariant only if these scores are G-invariant. This holds only for a choice of trivial representation $\rho(g) = 1$, such that the inequality above turns into an equality, that is, the latent queries can only attend to invariant keys, missing a lot of geometric information in the input data. Conversely, if we equip the latent tokens with some geometric feature f which transforms equivariantly to the input data, we can construct equivariant query vectors too, which enable expressive attention[2]:

$$\langle \boldsymbol{q}(v_i, g.f), \boldsymbol{k}(g.n_j) \rangle = \langle \rho(g)\boldsymbol{q}(v_i, f), \rho(g)\boldsymbol{k}(n_j) \rangle \qquad = \langle \boldsymbol{q}(v_i, f), \boldsymbol{k}(n_j) \rangle \qquad \forall g \in G \tag{2}$$

Inspired by recent works on Virtual Nodes for equivariant graph convolution (Sestak et al., 2024; Zhang et al., 2024), we propose to augment the latent tokens with simple global geometric features carrying the equivariance property of the input to allow for more expressive queries.

ViNE via PCA features A efficient but effective choice is given by the Principal Component Analysis (PCA) of the input tokens, which provides a center-of-mass vector and 3 orthogonal eigenvectors (weighted by their eigenvalues). Note that the *center-of-mass* $\boldsymbol{c} \in \mathbb{R}^3$ is a quantity transforming like a point (translates and rotates when the input does), while *eigenvectors* $W = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3) \in \mathbb{R}^{3\times 3}$ rotate but are translation-invariant. As such, they can be *encoded* respectively as a *point* and *translation operators multivectors*[3]: see Apx. B, Tab. 3. The global geometric feature $f = (\boldsymbol{c}, \boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3)$ comprising these 4 multivectors is concatenated to the invariant query tokens to build our **Virtual Node Embeddings** (**ViNE**). We argue this simple PCA features are effective for our purpose, while still being extremely efficient to compute. Indeed, in Apx. A, we show that ViNE allows the model to learn a *canonicalized* virtual point cloud, which can effectively leverage the distance-aware attention. In Apx. A, we also discuss the analogy with *canonicalization* methods.

[1] Assuming the orthonormal basis $\{e_i\}_{i=0}^3$, the geometric product is defined such that $e_0 e_0 = 0$ and $e_i e_i = 1$ for $i = 1, 2, 3$ and $e_i e_j = -e_j e_i$ for $i \neq j$.

[2] Here we assumed orthogonal ρ, which is not typically faithful to the whole $\mathrm{E}(3)$ group. This is the case also in GATr, which only uses the translation invariant coefficients to compute the attention weights, but introduces positional bias via a separate component.

[3] For minor improved expressivity, due to the structure of the equivariant linear maps, in practice we prefer encoding eigenvectors as vectors, which can be mapped to translations via a single linear map.

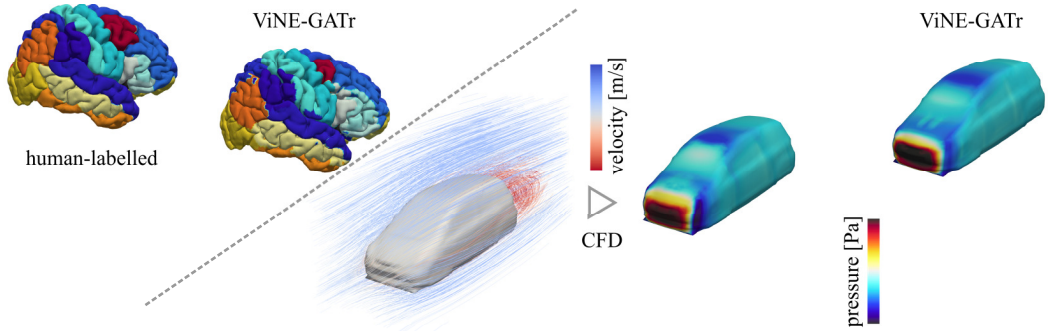

Figure 2: We apply ViNE-GATr to cortical surface parcellation (left) in the Mindboggle-101 dataset and estimation of surface pressure caused by airflow around car bodies (right) (ShapeNet-Car).

Table 1: **Comparison of inference time and accuracy** between LaB-GATr, ViNE-GATr and (†) LaB-GATr using random instead of farthest point sampling. Times are averaged over 101 cortical surface meshes (ca. 300K vertices each) from the Mindboggle-101 dataset. We report in brackets ($*$) the percentage that is due to pre-processing of the inputs. For the accuracy, we report mean $\pm$ standard deviation across the test split.

Model	**Inference [ms]** ($*$)	**Accuracy** $\uparrow$ **[%]**
LaB-GATr	2076.5 (86.2 %)	**79.4** $\pm$ 1.8
ViNE-GATr	390.1 (1.8 %)	77.3 $\pm$ 2.7
LaB-GATr†	**386.1** (0.0 %)	66.6 $\pm$ 2.2

Table 2: **ShapeNet-Car** benchmark comparison to baselines (values from (Alkin et al., 2024)).

Model	**Error** $\times$1e2 $\downarrow$
GINO	**2.14**
UPT	2.24
FNO	3.26
ViNE-GATr	3.85

Sign-Ambiguity of eigenvectors A problem with the PCA features is that eigenvectors are only defined up to a sign ambiguity. To handle this artificial symmetry, we compute 4 different global features $\{f_j\}_{j=1}^4$, one for each matrix with positive determinant among the 2^3 obtained by flipping the signs of the columns of W. Then, we only learn $V/4$ invariant embeddings but combine them with each of these global features to obtain V ViNE tokens. In other words, this strategy introduces some redundancy by inputting all possible 4 frames and leverages the permutation-equivariant attention to automatically handle the resulting set symmetry. In practice, in some experiments, we found beneficial for improved performance and stable convergence to ignore this artifact and simply use all sign-flipped eigenvectors in a single global feature $f = (\boldsymbol{c}, W, -W)$ for all virtual nodes.

4 Experiments

We evaluate ViNE-GATr on two tasks, see Fig. 2: cortical surface parcellation, i.e. the segmentation of functional brain regions (Mindboggle-101 (Klein et al., 2016)) and estimation of pressure exerted by airflow on the body of cars (ShapeNet-Car (Umetani & Bickel, 2018)).

Inference time study on large-scale cortical surface data We investigate the trade-off between inference time and accuracy using the Mindboggle-101 dataset. We create an ablation of LaB-GATr (Suk et al., 2024) ("large-scale biomedical GATr") which uses random subsampling of mesh vertices instead of farthest point sampling (FPS) which is the inference bottleneck in LaB-GATr.[4] Results are presented in Tab. 1. We choose the best-performing configuration of LaB- and ViNE-GATr which used (approximately) 1K hidden tokens and $V = 750$ virtual nodes, respectively. LaB-GATr achieves the highest accuracy albeit for the highest inference cost due to FPS. Ablating FPS for random sampling incurs an accuracy drop of 12.8% but greatly increases inference speed. ViNE-GATr presents a favorable accuracy trade-off by 2.1% under negligible increase in inference time.

[4]FPS is a iterative algorithm that requires distances to all points in each step ($\mathcal{O}(NV)$) and – due to its sequential nature – is challenging to parallelize.

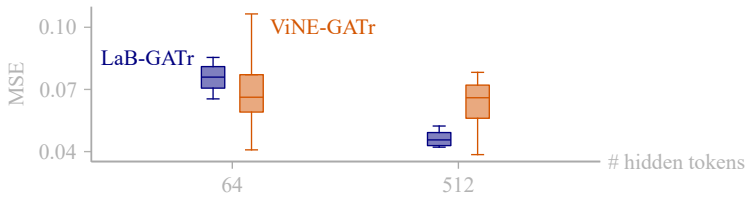

Figure 3: **MSE ↓ over allocated token budget** of LaB-GATr (blue) and ViNE-GATr (orange). Shown are mean, std, min. and max. error over 4 runs per budget.

Ablation on number of virtual nodes in cars airflow data We present preliminary results in which we train LaB- and ViNE-GATr with varying latent token budget and compare its influence on accuracy, see Fig. 3. We follow the setup of Alkin et al. (2024) and estimate car body pressure. Flow direction is encoded at every vertex in the mesh by oriented planes in $\mathbf{G}(3, 0, 1)$; the surface normal of the car body is also encoded as planes. We find that increased token budgets benefit both LaB- and ViNE-GATr while the scaling of ViNE-GATr is more moderate. However, ViNE-GATr performs better at the very small budget of $V = 64$. Tab. 2 shows how ViNE-GATr ($V = 512$) compares to the three models in Alkin et al. (2024), where we observe competitive performance. We did not perform any hyperparameter optimisation for ViNE-GATr in this study.

5 Discussion

Our preliminary results are a promising prospect for the utility of ViNE-GATr in the context of scientific problems on intricate geometric shapes. Compared to baselines, ViNE-GATr affords additional inference compute which could be spent e.g. on uncertainty quantification. ViNE-GATr also decouples the task from its geometry and generates predictions based on a latent space of virtual nodes, independently of the input spatial resolution (in contrast to LaB-GATr). We find that increased numbers of virtual nodes improve the accuracy on the ShapeNet-Car dataset, which hints at decoupling of model expressivity and computational cost due to the input size. See limitations in Apx. D. In future work, we aim to compare ViNE-GATr to different strategies to achieve $\mathrm{E}(3)$ equivariance, e.g. based on frame averaging (Puny et al., 2022), and non-equivariant baselines such as UPT (Alkin et al., 2024).

References

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers. *arXiv preprint arXiv:2402.12365*, 2024. (Cited on pages 2, 4, and 5)

Marcin Andrychowicz, Lasse Espeholt, Di Li, Samier Merchant, Alexander Merose, Fred Zyda, Shreya Agrawal, and Nal Kalchbrenner. Deep learning for day forecasts from sparse observations. *arXiv preprint arXiv:2306.06079*, 2023. (Cited on page 1)

Amirhossein Arzani, Jian-Xun Wang, Michael S Sacks, and Shawn C Shadden. Machine learning for cardiovascular biomechanics modeling: challenges and beyond. *Annals of Biomedical Engineering*, 50(6):615–627, 2022. (Cited on page 1)

Johann Brehmer, Pim de Haan, Sönke Behrends, and Taco S Cohen. Geometric algebra transformer. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 35472–35496. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6f6dd92b03ff9be7468a6104611c9187-Paper-Conference.pdf. (Cited on pages 2, 3, 8, and 9)

Johann Brehmer, Víctor Bresó, Pim de Haan, Tilman Plehn, Huilin Qu, Jonas Spinner, and Jesse Thaler. A lorentz-equivariant transformer for all of the lhc. *arXiv preprint arXiv:2411.00446*, 2024. (Cited on pages 1 and 3)

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. (Cited on page 1)

Anadi Chaman and Ivan Dokmanic. Truly shift-invariant convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3773–3783, June 2021. (Cited on page 2)

Pin Chen, Luoxuan Peng, Rui Jiao, Qing Mo, WANG Zhen, Wenbing Huang, Yang Liu, and Yutong Lu. Learning superconductivity from ordered and disordered material structures. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. (Cited on page 1)

Kamal Choudhary and Kevin Garrity. Designing high-tc superconductors with bcs-inspired screening, density functional theory, and deep-learning. *npj Computational Materials*, 8(1):244, 2022. (Cited on page 1)

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

Simon Dahan, Logan Zane John Williams, Daniel Rueckert, and Emma Claire Robinson. The multiscale surface vision transformer. In *Medical Imaging with Deep Learning*, 2024. (Cited on page 1)

Leo Dorst. A guided tour to the plane-based geometric algebra pga. 2020. URL https://geometricalgebra.org/downloads/PGA4CS.pdf. (Cited on pages 8 and 9)

Bryan R Goldsmith, Jacques Esterhuizen, Jin-Xun Liu, Christopher J Bartel, and Christopher Sutton. Machine learning for heterogeneous catalyst design and discovery. 2018. (Cited on page 1)

Thomas Hehn, Markus Peschl, Tribhuvanesh Orekondy, Arash Behboodi, and Johann Brehmer. Differentiable and learnable wireless simulation with geometric transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9TClCDZXeh. (Cited on pages 3 and 8)

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*. (Cited on page 2)

Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions, 2023. (Cited on page 8)

Arno Klein, Satrajit S. Ghosh, Forrest S. Bao, Joachim Giard, Yrjö Häme, Eliezer Stavsky, Noah Lee, Brian Rossa, Martin Reuter, Elias Chaibub Neto, and Anisha Keshavan. Mindboggling morphometry of human brains. *bioRxiv*, 2016. doi: 10.1101/091322. URL https://www.biorxiv.org/content/early/2016/12/03/091322. (Cited on page 4)

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Graphcast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*, 2022. (Cited on page 1)

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: a25 foundation model for weather and climate. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 25904–25938, 2023. (Cited on page 1)

Tilman Plehn, Anja Butter, Barry Dillon, Theo Heimel, Claudius Krause, and Ramon Winterhalder. Modern machine learning for lhc physicists. *arXiv preprint arXiv:2211.01421*, 2022. (Cited on page 1)

Omri Puny, Matan Atzmon, Edward J. Smith, Ishan Misra, Aditya Grover, Heli Ben-Hamu, and Yaron Lipman. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=zIUyj55nXR. (Cited on page 5)

Md Ashiqur Rahman and Raymond A. Yeh. Group downsampling with equivariant anti-aliasing. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sOte83GogU. (Cited on page 2)

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. (Cited on page 1)

Renan A. Rojas-Gomez, Teck-Yian Lim, Alexander G. Schwing, Minh N. Do, and Raymond A. Yeh. Learnable polyphase sampling for shift invariant and equivariant convolutional networks, 2022. (Cited on page 2)

David Ruhe, Jayesh K Gupta, Steven de Keninck, Max Welling, and Johannes Brandstetter. Geometric clifford algebra networks. In *ICLR*, 2023. (Cited on page 8)

Florian Sestak, Lisa Schneckenreiter, Johannes Brandstetter, Sepp Hochreiter, Andreas Mayr, and Günter Klambauer. Vn-egnn: E (3)-equivariant graph neural networks with virtual nodes enhance protein binding site identification. *arXiv preprint arXiv:2404.07194*, 2024. (Cited on page 3)

Lucas Spangher, Allen M. Wang, Andrew Maris, Myles Stapelberg, Viraj Mehta, Alex Saperstein, Stephen Lane-Walsh, Akshata Kishore Moharir, Alessandro Pau, and Cristina Rea. Position: Opportunities exist for machine learning in magnetic fusion energy. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=arwP5FA2dO. (Cited on page 1)

Julian Suk, Baris Imre, and Jelmer M. Wolterink. LaB-GATr: Geometric algebra transformers for large biomedical surface and volume meshes. In Marius George Linguraru, Qi Dou, Aasa Feragen, Stamatia Giannarou, Ben Glocker, Karim Lekadir, and Julia A. Schnabel (eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, pp. 185–195, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-72390-2. (Cited on pages 1, 3, and 4)

Nobuyuki Umetani and Bernd Bickel. Learning three-dimensional flow for interactive aerodynamic design. *ACM Transactions on Graphics (SIGGRAPH 2018)*, 37(4), 2018. doi: 10.1145/3197517.3201325. URL https://doi.org/10.1145/3197517.3201325. (Cited on page 4)

Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. Coordinate independent convolutional networks–isometry and gauge equivariant convolutions on riemannian manifolds. *arXiv preprint arXiv:2106.06020*, 2021. (Cited on page 1)

David R Wessels, David M Knigge, Samuele Papa, Riccardo Valperga, Sharvaree Vadgama, Efstratios Gavves, and Erik J Bekkers. Grounding continuous representations in geometry: Equivariant neural fields. *arXiv preprint arXiv:2406.05753*, 2024. (Cited on page 8)

Jin Xu, Hyunjik Kim, Tom Rainforth, and Yee Whye Teh. Group equivariant subsampling. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=CtaDl9L0bIQ. (Cited on page 2)

Richard Zhang. Making convolutional networks shift-invariant again. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7324–7334. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/zhang19a.html. (Cited on page 2)

Yuelin Zhang, Jiacheng Cen, Jiaqi Han, Zhiqiang Zhang, Jun Zhou, and Wenbing Huang. Improving equivariant graph neural networks on large geometric graphs via virtual nodes learning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=wWdkNkUY8k. (Cited on page 3)

A On the effectiveness of PCA features and the relation with canonicalization methods

Let $X \in \mathbb{R}^{3\times N}$ be the coordinates of the input N tokens and let $S \in \mathbb{R}^{3\times V}$ be three invariant scalars learned by each of the V virtual tokens; then, $\hat{S}(X) = W(X)S + \boldsymbol{c}(X) \in \mathbb{R}^{3\times V}$ represents the *canonicalized* coordinates of the virtual point cloud S - where we emphasized the dependency of the PCA features on the input X. Coincidentally, this computation is achieved by i) a weighted combination of the three translations $W = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3)$ by the scalars in S followed by ii) the application of this translation on the point $\boldsymbol{c}$, which can be performed by a sequence of two linear and bilinear layers in GATr. See also Tab. 3 for the precise embeddings of these scalar, point and translation features into multivectors. In other words, this construction allows the model to learn a *canonicalized* virtual point cloud, which can effectively leverage the distance-aware attention mechanism.

Comparison with canonicalization methods Previous works studied canonicalization methods, e.g. Kaba et al. (2023), but we emphasize some important differences. First, the canonicalized coordinates enrich the latent queries, but the virtual nodes still attend to the input tokens, which contain the full geometry and pose information; hence, the canonicalization of the virtual scalars is not a bottleneck in passing pose-information to the rest of the model. Second, we still process the latent data with an equivariant architecture, which helps preserving not only the global but also the local isometry equivariance. Additionally, each virtual node (and each attention head) can learn to weight the canonicalized features differently.

We finally draw an analogy with the work of Wessels et al. (2024), which particularly resembles the decoder of our model, although the work focuses more on the equivariant neural field aspect of it.

B Geometric algebra

As representation, GATr uses the projective geometric algebra $\mathbb{G}_{3,0,1}$. Here we summarize key aspects of this algebra. This summary is taken from the appendix of Hehn et al. (2025). For a precise definition and pedagogical introduction, we refer the reader to Dorst (2020).

Geometric algebra. A geometric algebra $\mathbb{G}_{p,q,r}$ consists of a vector space together with a bilinear operation, the *geometric product*, that maps two elements of the vector space to another element of the vector space.

The elements of the vector space are known as *multivectors*. Their space is constructed by extending a base vector space $\mathbb{R}^d$ to lower orders (scalars) and higher-orders (bi-vectors, tri-vectors, ...). The algebra combines all of these orders (or *grades*) in one 2^d-dimensional vector space. From a basis for the base space, for instance (e_1, e_2, e_3), one can construct a basis for the multivector space. A multivector expressed in that basis then reads, for instance for $d = 3$, $x = x_\emptyset + x_1e_1 + x_2e_2 + x_3e_3 + x_{12}e_1e_2 + x_{13}e_1e_3 + x_{23}e_2e_3 + x_{123}e_1e_2e_3$.

The geometric product is fully defined by bilinearity, associativity, and the condition that the geometric product of a vector with itself is equal to its norm. The geometric product generally maps between different grades. For instance, the geometric product of two vectors will consist of a scalar, the inner product between the vectors, and a bivector, which is related to the cross-product of $\mathbb{R}^3$. In particular, the conventional basis elements of grade $k > 1$ are constructed as the geometric product of the vector basis elements e_i. For instance, $e_{12} = e_1e_2$ is a basis bivector. From the defining properties of the geometric products it follows that the geometric product between orthogonal basis elements is antisymmetric, $e_ie_j = -e_je_i$. Thus, for a d-dimensional basis space, there are $\binom{d}{k}$ independent basis elements at grade k.

Projective geometric algebra. To represent three-dimensional objects including absolute positions, we use a geometric algebra based on a base space with $d = 4$, adding a *homogeneous coordinate* to the 3D space.[5] We use a basis (e_0, e_1, e_2, e_3) with a metric such that $e_0^2 = 0$ and $e_i^2 = 1$ for

[5] A three-dimensional base space is not sufficient to represent absolute positions and translations acting on them in a convenient form. See Dorst (2020); Ruhe et al. (2023); Brehmer et al. (2023) for an in-depth discussion.

Object / operator	Scalar	Vector		Bivector		Trivector		PS
	1	e_0	e_i	e_{0i}	e_{ij}	e_{0ij}	e_{123}	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

Table 3: Embeddings of common geometric objects and transformations into the projective geometric algebra $\mathbb{G}_{3,0,1}$. The columns show different components of the multivectors with the corresponding basis elements, with $i, j \in \{1, 2, 3\}, j \neq i$, i.e. $ij \in \{12, 13, 23\}$. For simplicity, we fix gauge ambiguities (the weight of the multivectors) and leave out signs (which depend on the ordering of indices in the basis elements). This is a copy of Tab. 1 from Brehmer et al. (2023), but we highlighted the entries we used for our PCA embeddings in Sec. 3.

$i = 1, 2, 3$. The multivector space is thus $2^4 = 16$-dimensional. This algebra is known as the projective geometric algebra $\mathbb{G}_{3,0,1}$.

Canonical embedding of geometric primitives. In $\mathbb{G}_{3,0,1}$, we can represent geometric primitives as follows:

- Scalars (data that do not transform under translation, rotations, and reflections) are represented as the scalars of the multivectors (grade $k = 0$).
- Oriented planes are represented as vectors ($k = 1$), encoding the plane normal as well as the distance from the origin.
- Lines or directions are represented as bivectors ($k = 2$), encoding the direction as well as the shift from the origin.
- Points or positions are represented as trivectors ($k = 3$).

For more details, we refer the reader to Tab. 3 (Tab. 1 in Brehmer et al. (2023)) or to Dorst (2020).

C Computational complexity

Assume N input tokens and a fixed set of V Virtual Nodes, independent of the input. The initial cross-attention has cost $\mathcal{O}(NV)$. The majority of transformer layers operate on virtual tokens in the latent with $\mathcal{O}(V^2)$ complexity. Finally, the model decodes information from virtual tokens to the output tokens of interest (usually, the input tokens) with another cross-attention. The overall $\mathcal{O}(V^2 + NV)$ complexity highlights that cost scales linearly with input tokens N, while we note that the computational capacity depends on number of virtual tokens V, which is now disentangled from the input data discretization.

D Limitations

In our experiments we found ViNE-GATr training convergence to be somewhat sensitive to virtual node initialisation, which we are currently investigating. Furthermore, ViNE-GATr achieved peak performance on Mindboggle-101 by breaking the permutation symmetry among PCA eigenvectors. To address these issues, we aim to look into soft (training-time) regularisation of virtual node position to better follow the topology of the input data.

E ADDITIONAL EXPERIMENTAL DETAILS

The cortical surface meshes are heterogeneous and have around 300K vertices, while the airflow domain and cars are represented by volume and surface meshes of ca. 33k vertices. We train all our models under L^1 loss with a batch size of four using Adam with an initial learning rate of $3\mathrm{e}-4$ and exponential decay until we observe convergence on a held-out validation split. All trainings were run on NVIDIA Tesla V100 (32 GB) GPUs and we used gradient accumulation in the case of the large-scale cortical surface data. ViNE-GATr had 360k - 400k trainable parameters, dependent on the specific hyperparameters.

In the large-scale cortical surface data experiment, all models encoded the surface normal as oriented planes and mesh vertices as points in $\mathbf{G}(3,0,1)$. LaB-GATr followed the same layout as ViNE-GATr but uses learned interpolation instead of cross-attention in the decoding step. In these experiment, note that we achieved the presented accuracy of ViNE-GATr by breaking sign-flip symmetry and concatenating all PCA eigenvectors along the channels dimension.