

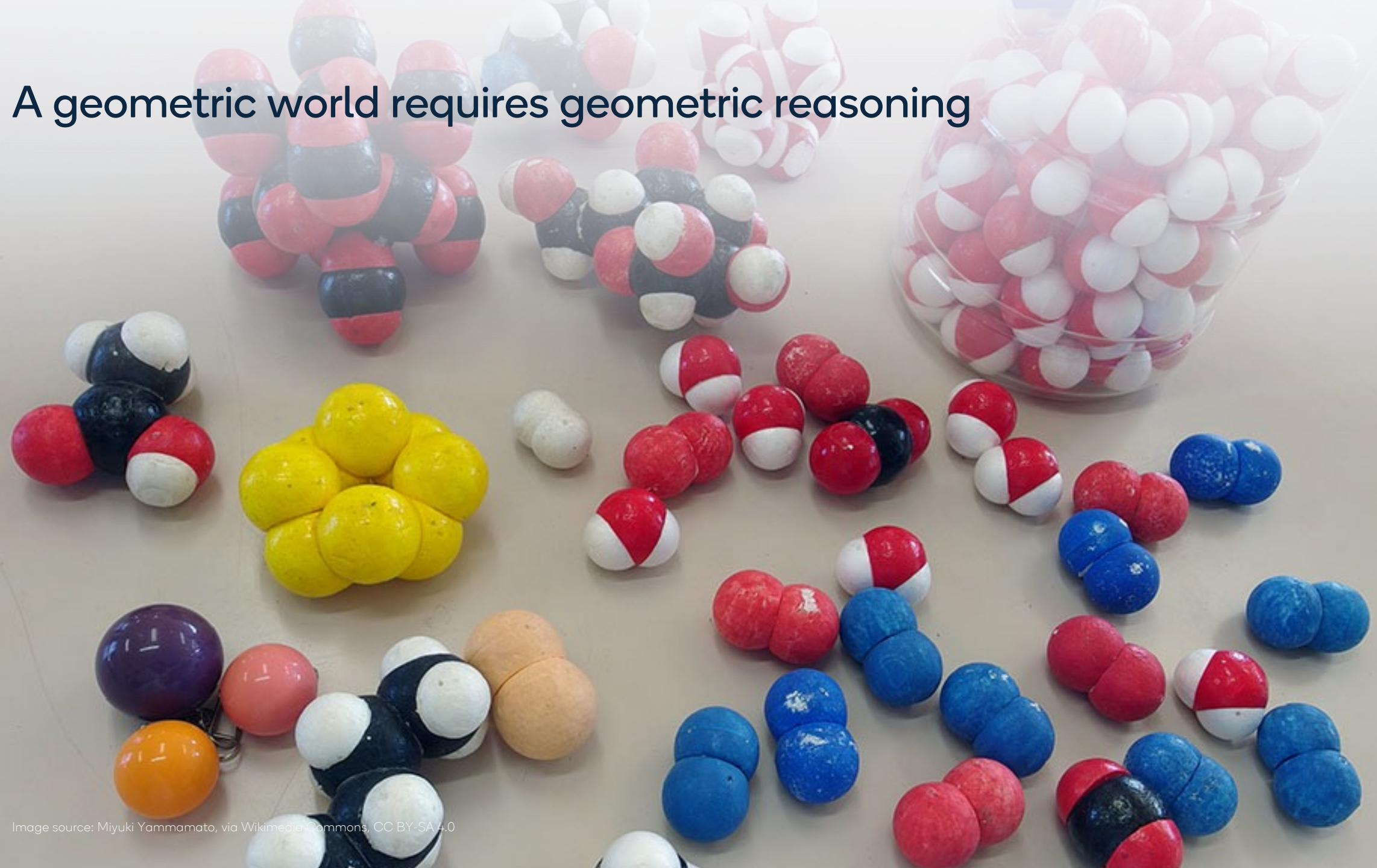
Geometric Algebra Transformers

Johann Brehmer

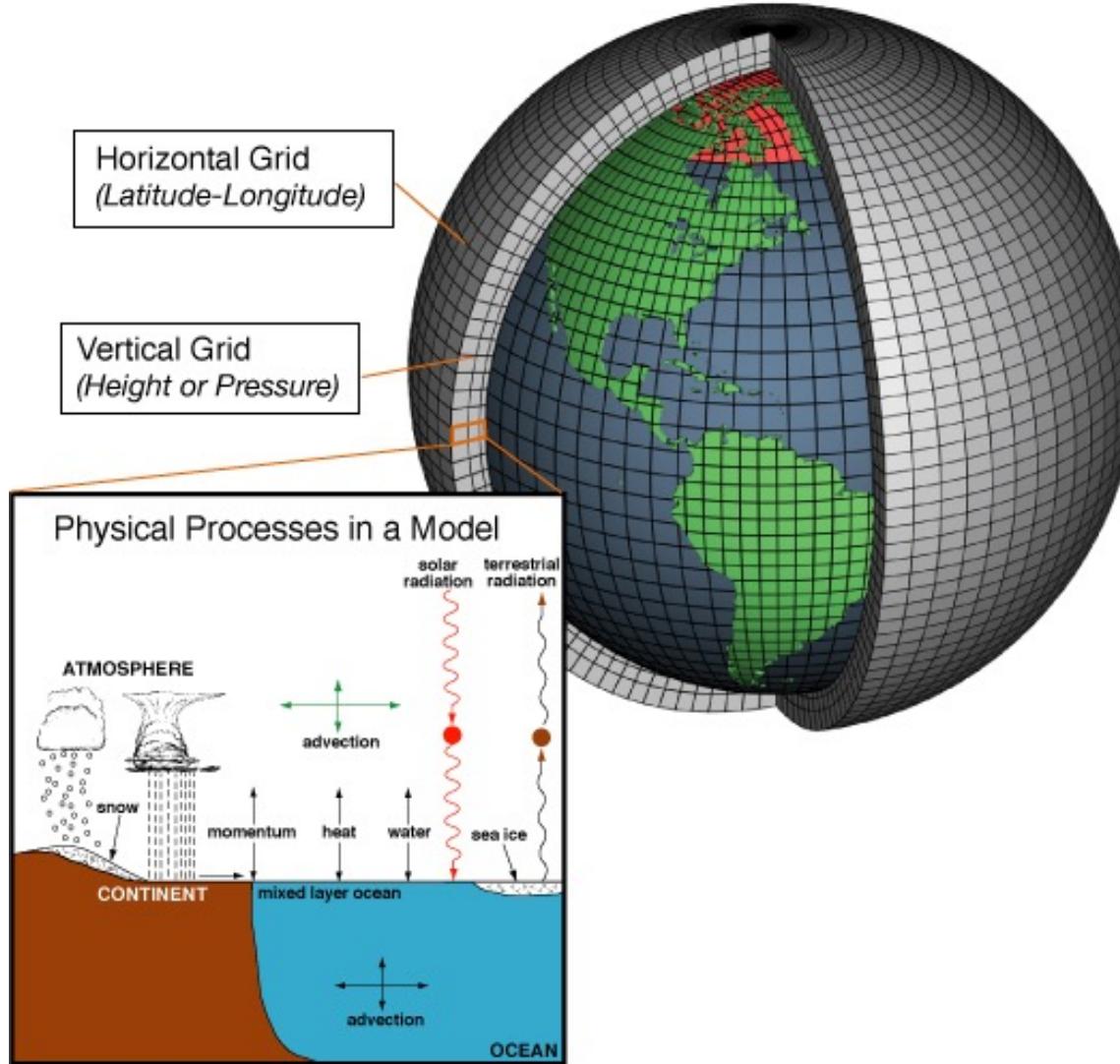
Qualcomm AI Research

Work with Pim de Haan, Sönke Behrends, Taco Cohen

A geometric world requires geometric reasoning

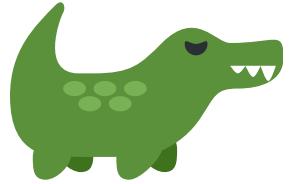


A geometric world requires geometric reasoning

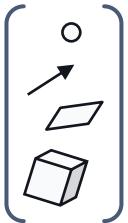


A geometric world requires geometric reasoning

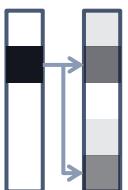




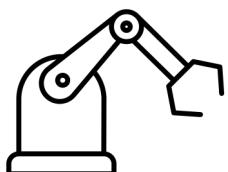
We introduce the **Geometric Algebra Transformer**,
a **versatile architecture for geometric data**



GATr takes into account **geometric structure** through geometric algebra representations and equivariance...



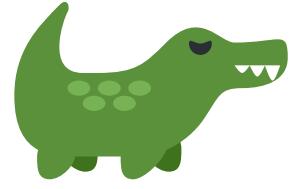
...but has the **scalability** and expressivity of transformers



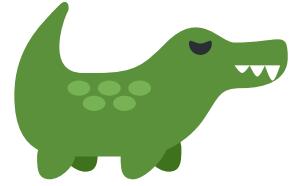
Initial experiments show a **strong performance**, even from little data



GATr 101



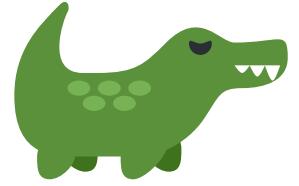
Geometric Algebra
Transformer



=

a **versatile architecture** for **geometric data**

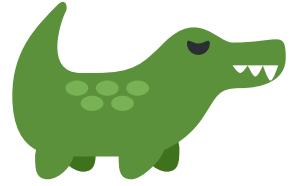
Geometric **A**lgebra
Transformer



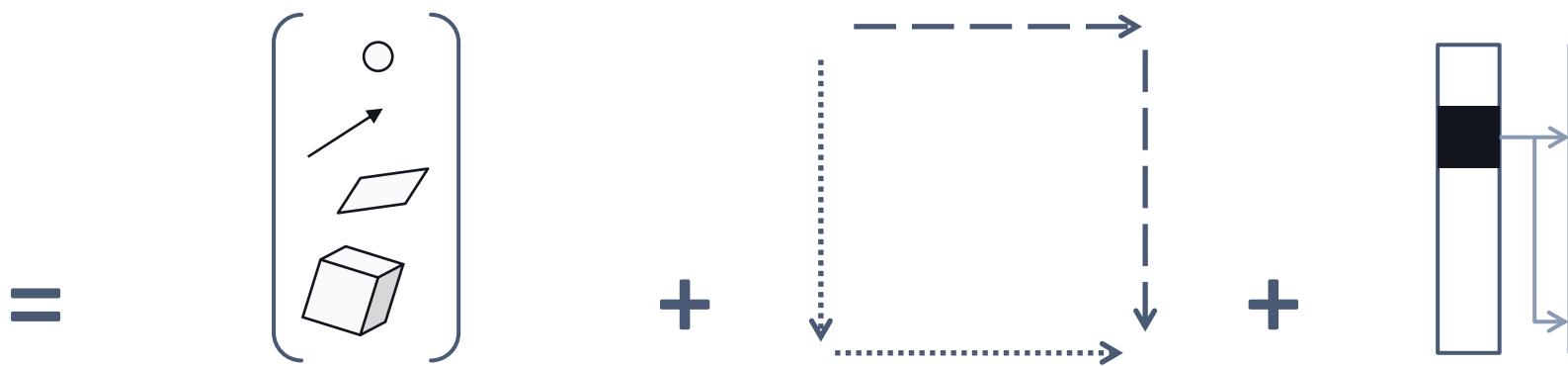
=

geometric deep learning becomes **scalable**

Geometric **A**lgebra
Transformer



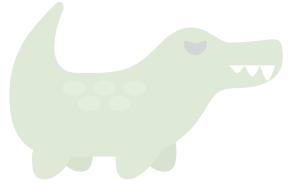
Geometric **A**lgebra
Transformer



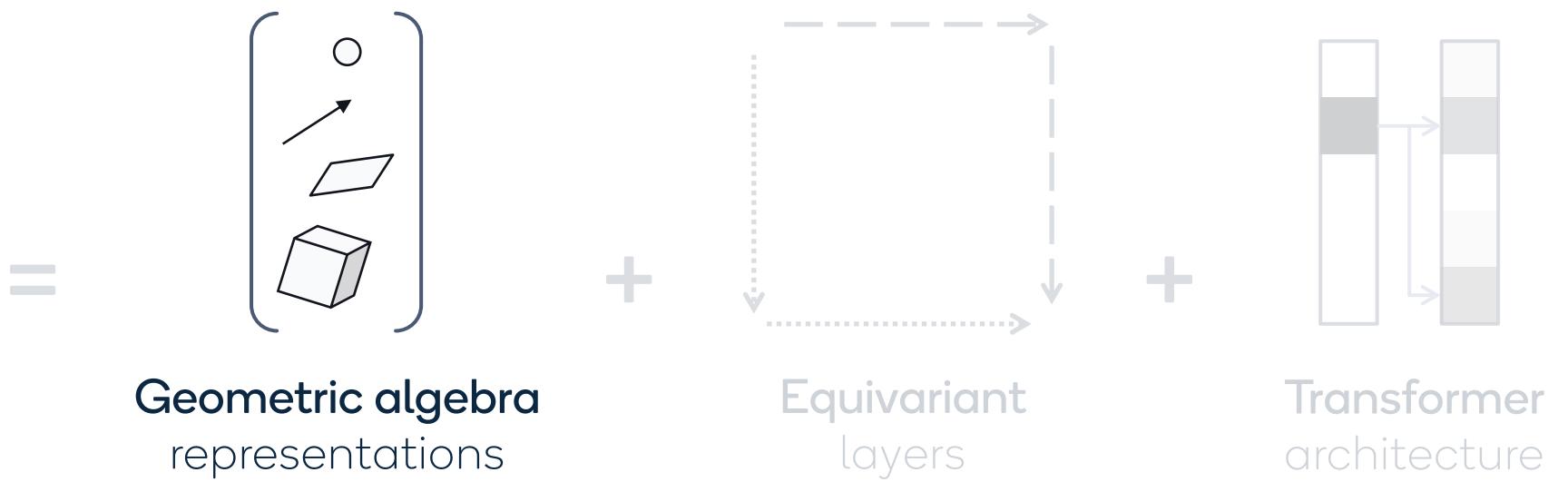
Geometric algebra
representations

Equivariant
layers

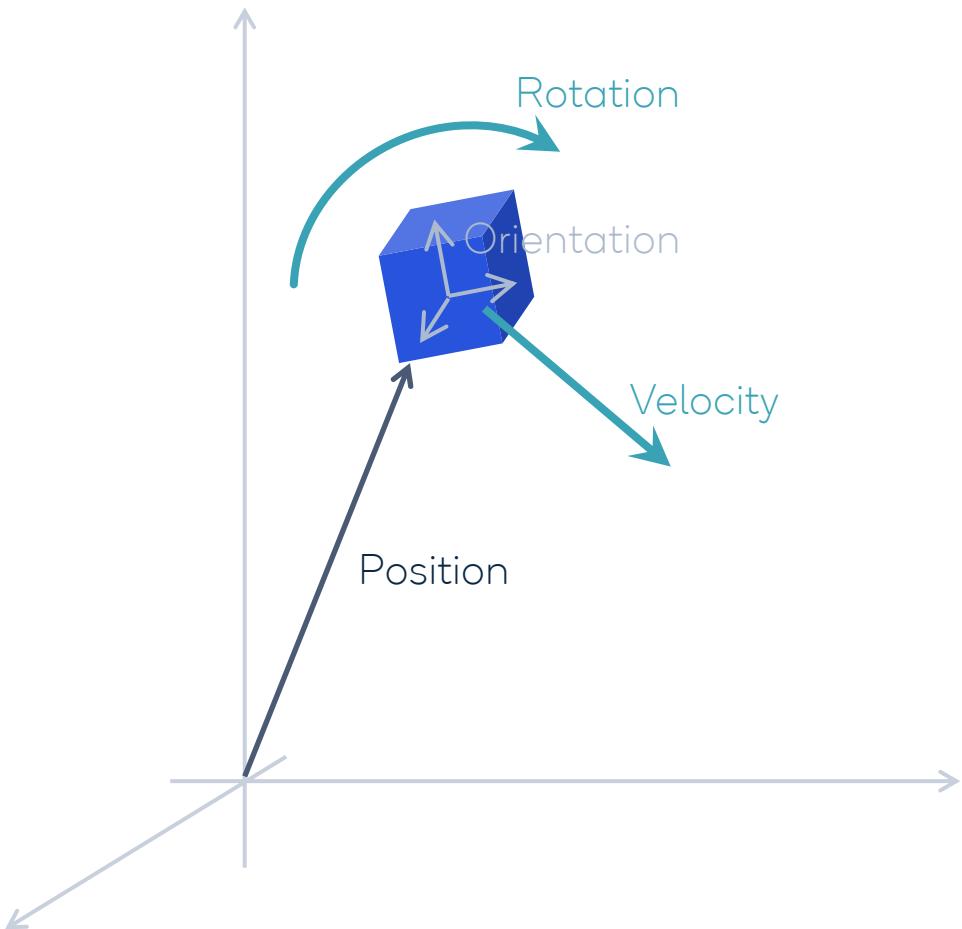
Transformer
architecture



Geometric Algebra
Transformer



Representing geometric data



How do you parameterize this object?

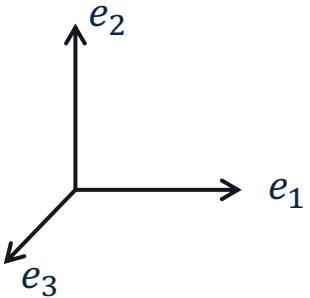
- Standard deep learning: 18 numbers
- Previous **geometric deep learning**: 6 vectors
- **GATr**: 1 position, 3 directions, 1 translation, 1 rotation

Why use different types?

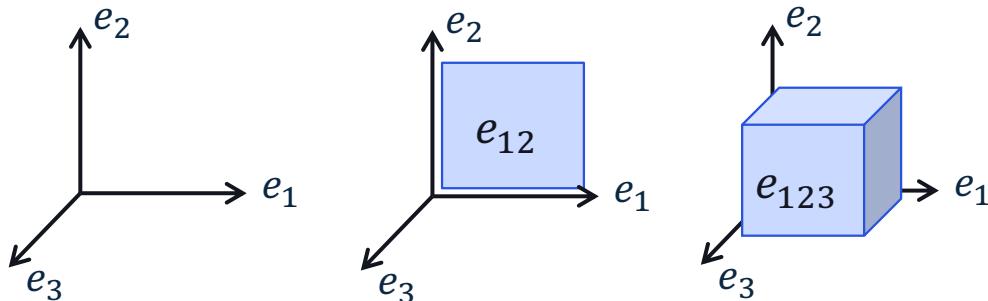
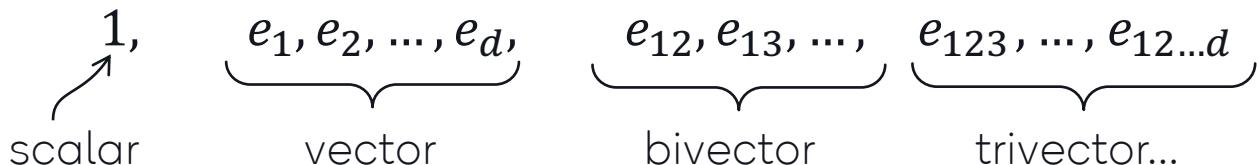
- Types have different **common patterns** (compute distances between positions, but not between direction vectors)
- Types differ in behaviour under **transformations**
- Types provide an inductive bias, potentially improving **sample efficiency** and **generalization**

Geometric algebra

- Vector space V with d dimensions and inner products
 - Basis e_1, e_2, \dots, e_d



- Geometric algebra $\mathcal{G}(V)$ has 2^d dimensions, basis



- Geometric product $\mathcal{G}(V) \times \mathcal{G}(V) \rightarrow \mathcal{G}(V)$
 - Generalizes dot product and cross product



Graßmann



Clifford

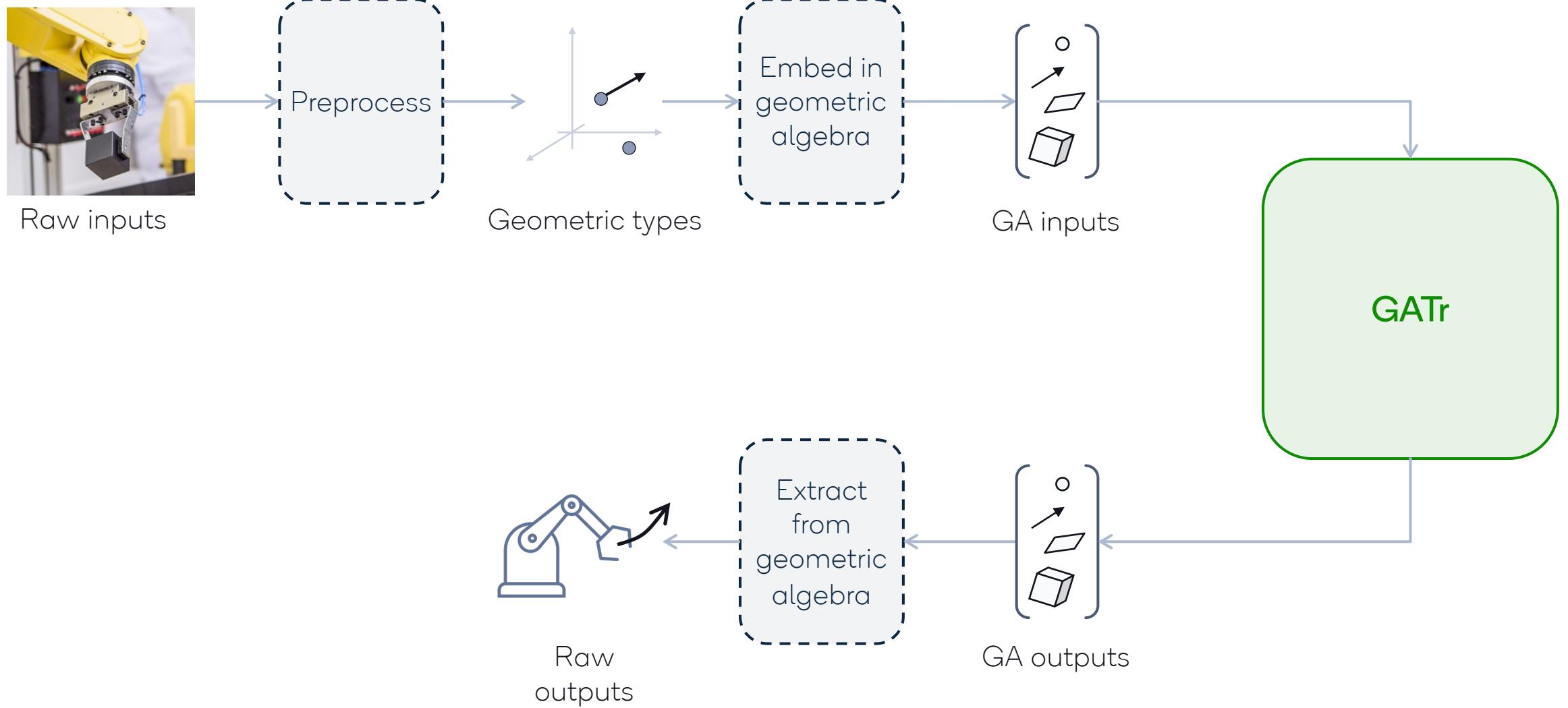
Projective GA as representation for ML

- We use GA representations in addition to the usual unstructured vector space
- Offers **$16n$ -dimensional representation** of 3D geometric data
 - “**Typing**”: a point is not a direction of movement is not the orientation of a plane
 - Established embeddings for **3D primitives and transformations**

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

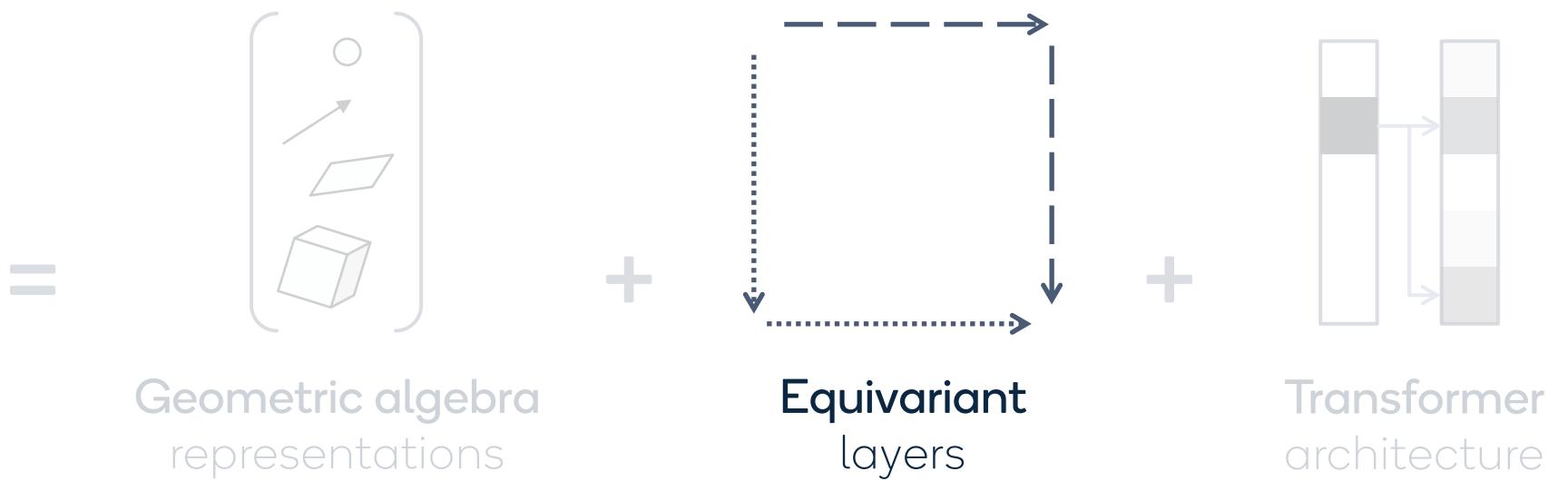
- Geometric product: **canonical operation** on these representations

GA representations in practice





Geometric Algebra
Transformer



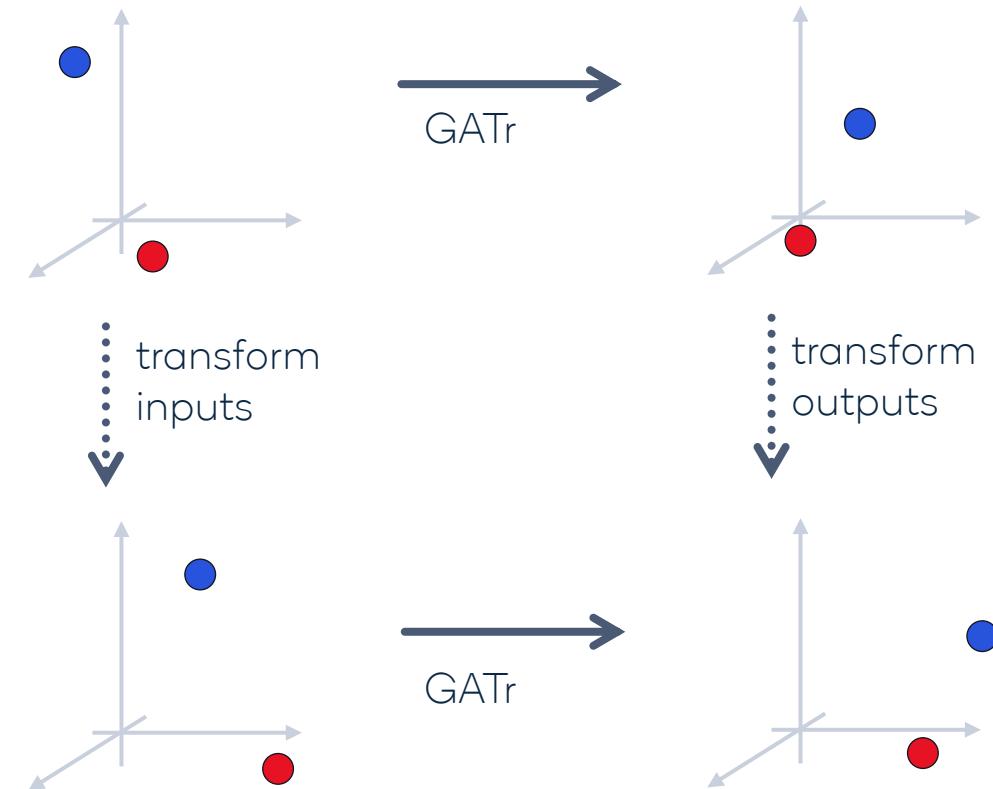
E(3) equivariance

- Goal: **GATr** should behave consistently under input transformations

- Laws of physics, dynamics of robots, ... are the same everywhere in space

- Implementation: **GATr layers are E(3)-equivariant**

- E(3): symmetry group of 3D space (translations and rotations)
- Equivariance: when we transform network inputs, the outputs transform consistently



New E(3)-equivariant layers between GA representations

- We fully characterize how equivariance constrains **linear layers**:

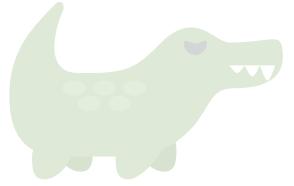
Proposition 1. Any linear map $\phi : \mathbb{G}_{d,0,1} \rightarrow \mathbb{G}_{d,0,1}$ that is equivariant to $\text{Pin}(d, 0, 1)$ is of the form

$$\phi(x) = \sum_{k=0}^{d+1} w_k \langle x \rangle_k + \sum_{k=0}^d v_k e_0 \langle x \rangle_k \quad (4)$$

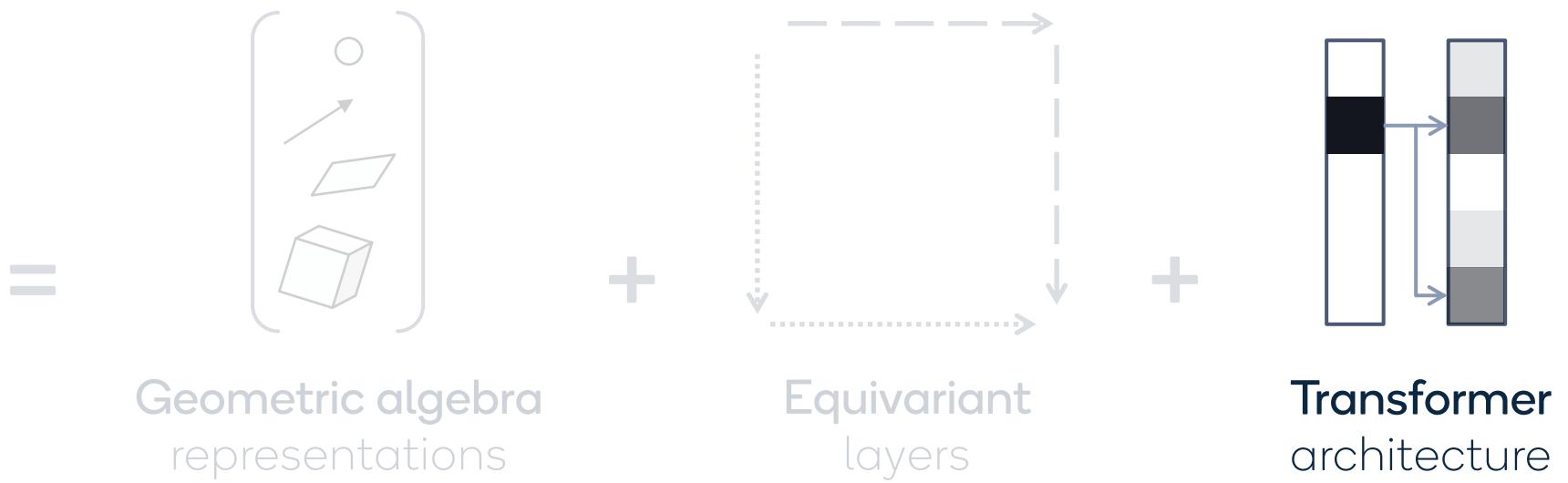
for parameters $w \in \mathbb{R}^{d+2}, v \in \mathbb{R}^{d+1}$. Here $\langle x \rangle_k$ is the blade projection of a multivector, which sets all non-grade- k elements to zero.

essentially E(n)

- Plus: equivariant attention, nonlinearities, normalization...



Geometric **A**lgebra
Transformer

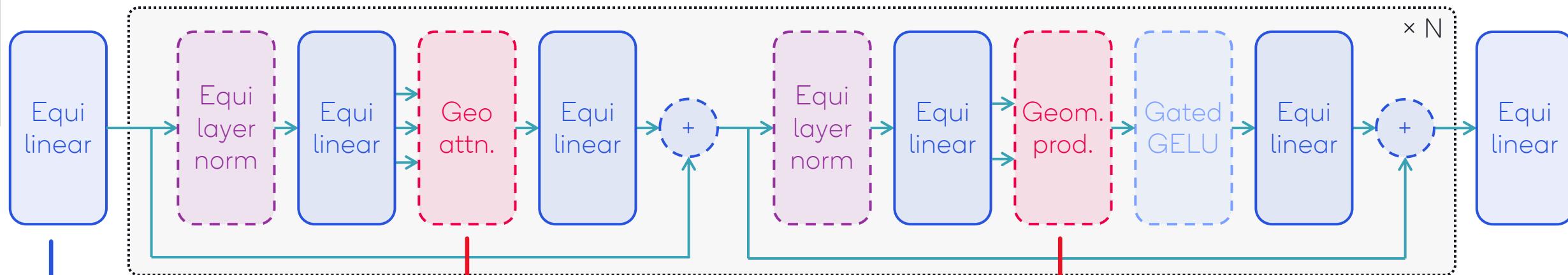


Input and output data

can have one or
multiple token
dimensions

Attention blocks

can be stacked to large depth,
gradients are propagated
efficiently



Linear layers
between GA
representations with
equivariance constraint

Geometric attention
generalizes scaled dot-product attention

Geometric product
allow for construction
of new geometric types

GATr is powered by **dot-product attention**

Message-passing neural networks

$$m_{i \rightarrow j} = \Phi(x_i, x_j, e_{ij})$$

$$x'_j = \Psi\left(x_j, \sum_i m_{i \rightarrow j}\right)$$

GATr (and other Transformers)

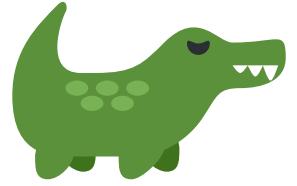
$$V_i, K_i, Q_i = \Phi(x_i)$$

$$x' = \sigma\left(QK^T / \sqrt{d}\right)V$$

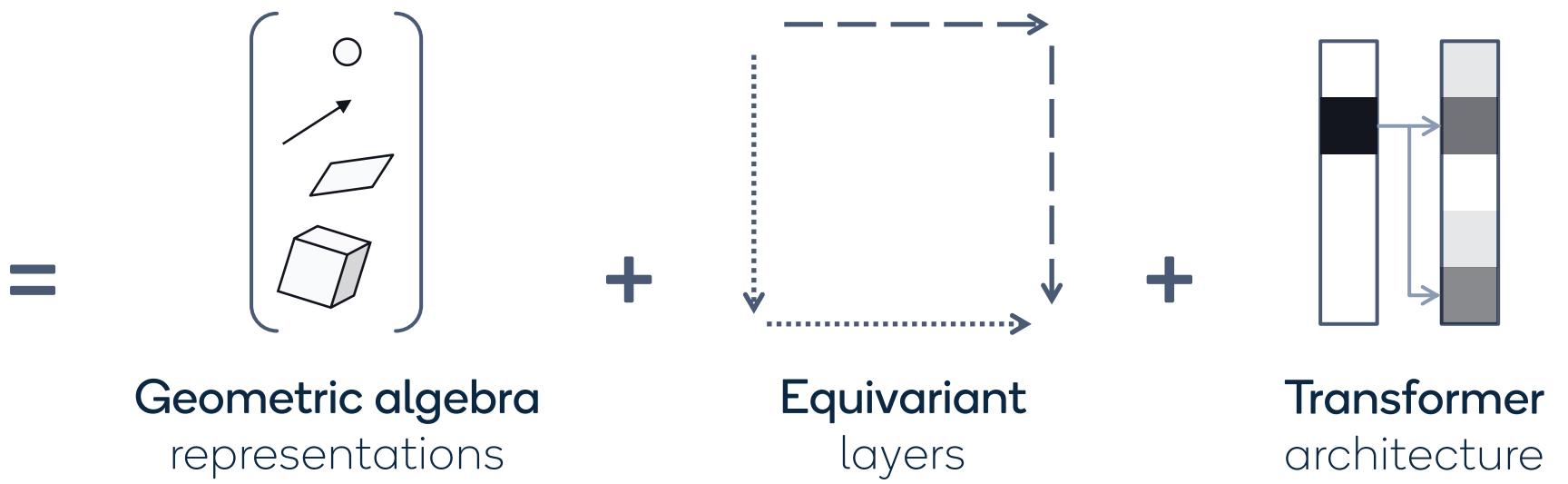
- Node and edge representations
- (Equivariant) network on each edge

- Only node representations
- Dot-product per edge,
with highly optimized implementations
(e.g. flash attention)

Transformers have same theoretical complexity, but dramatically more efficient in practice



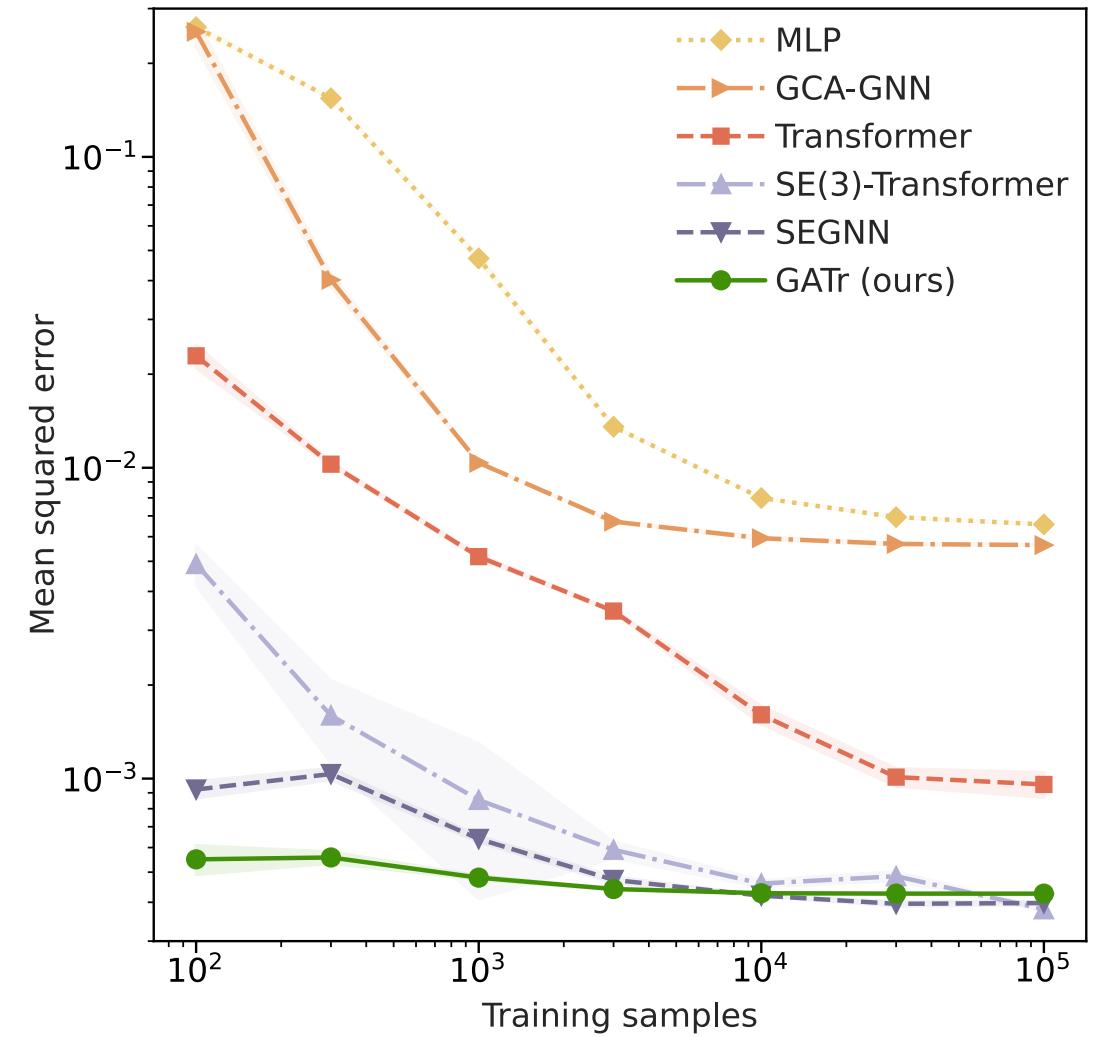
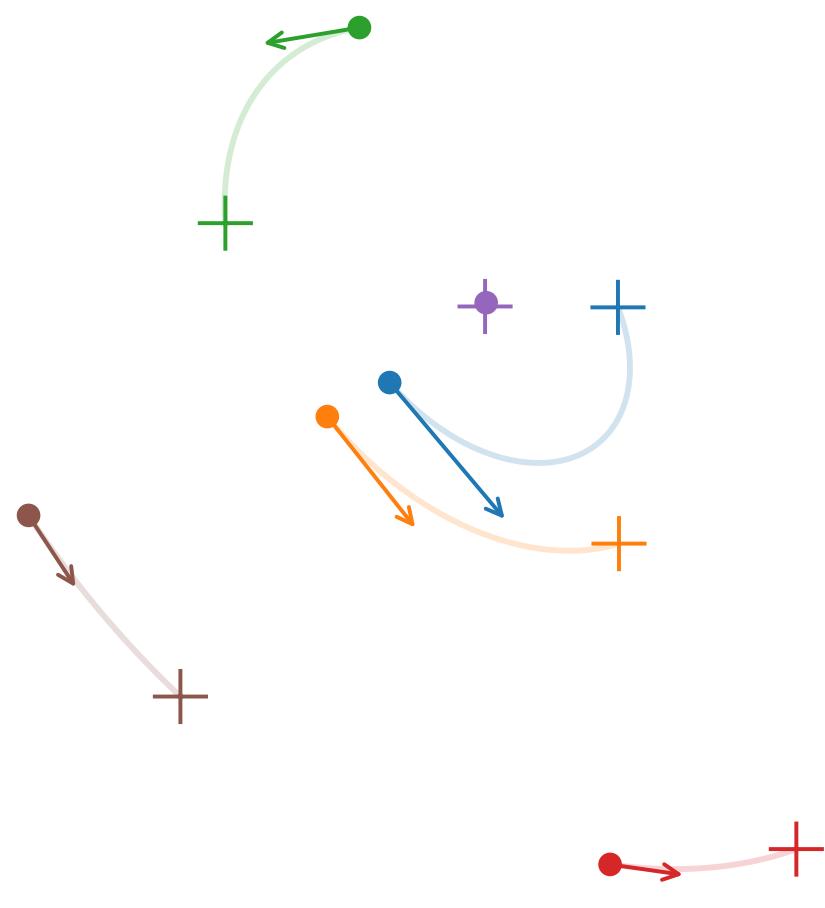
Geometric **A**lgebra
Transformer



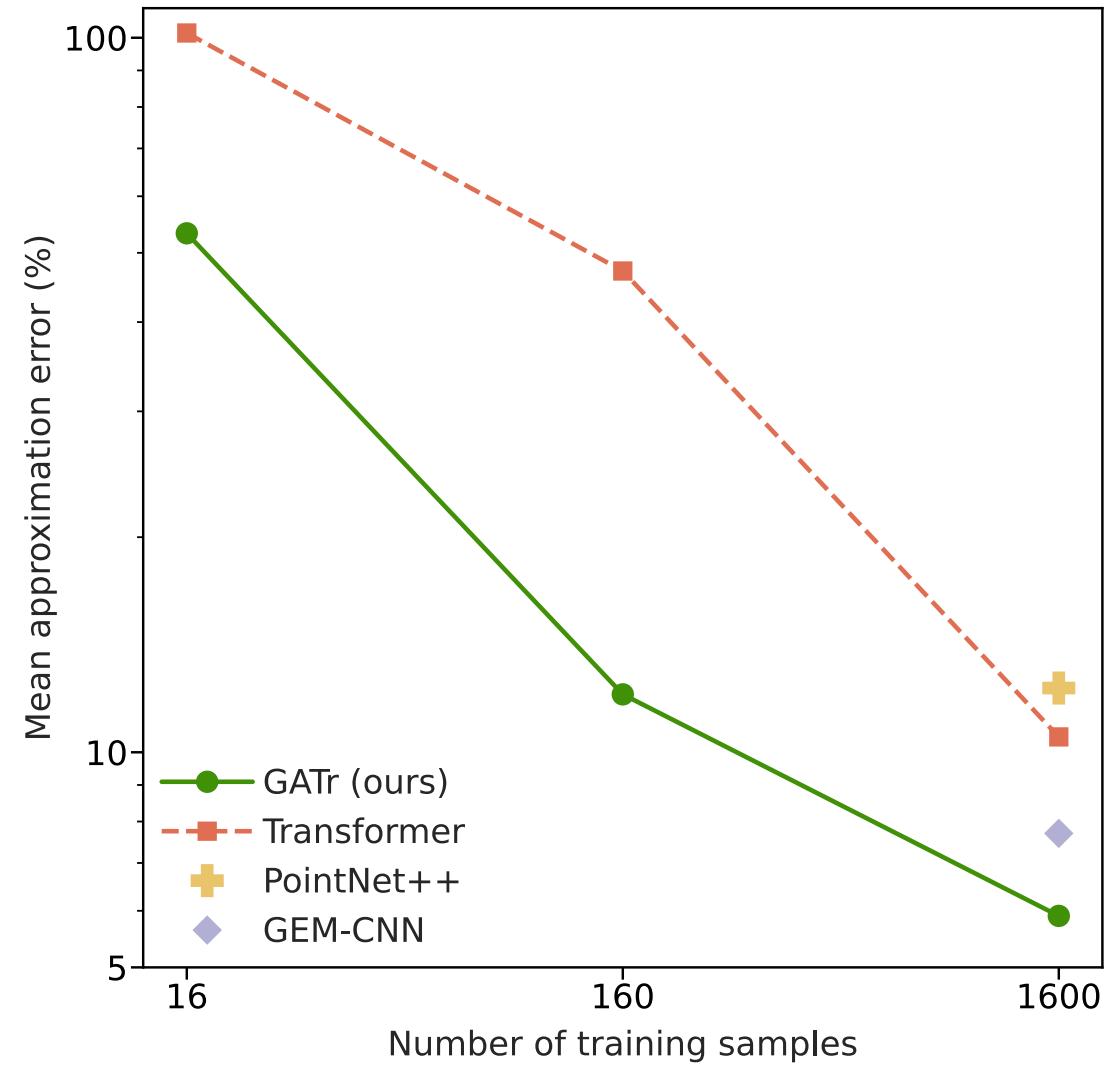
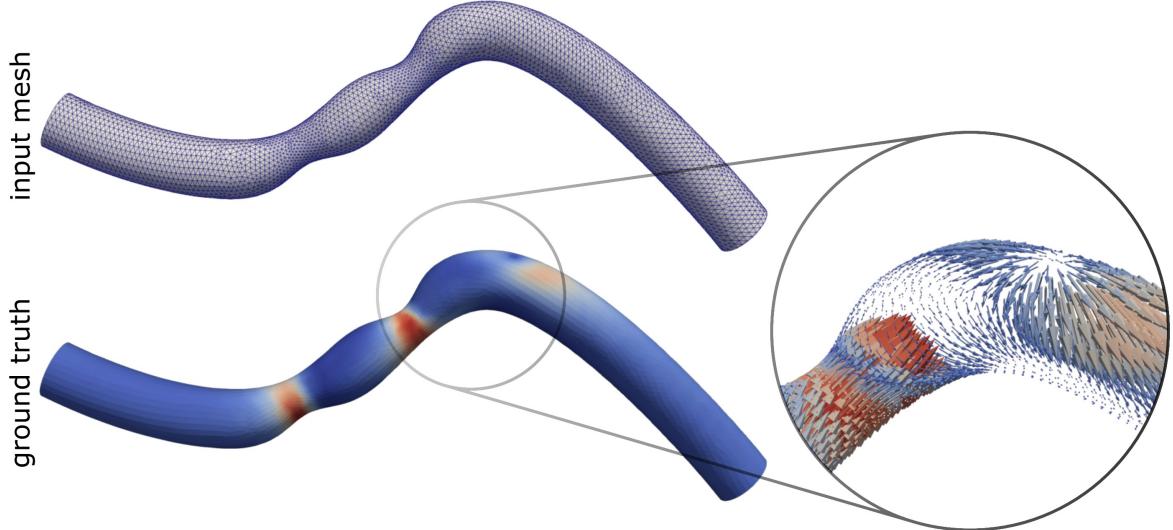
GATr in the wild



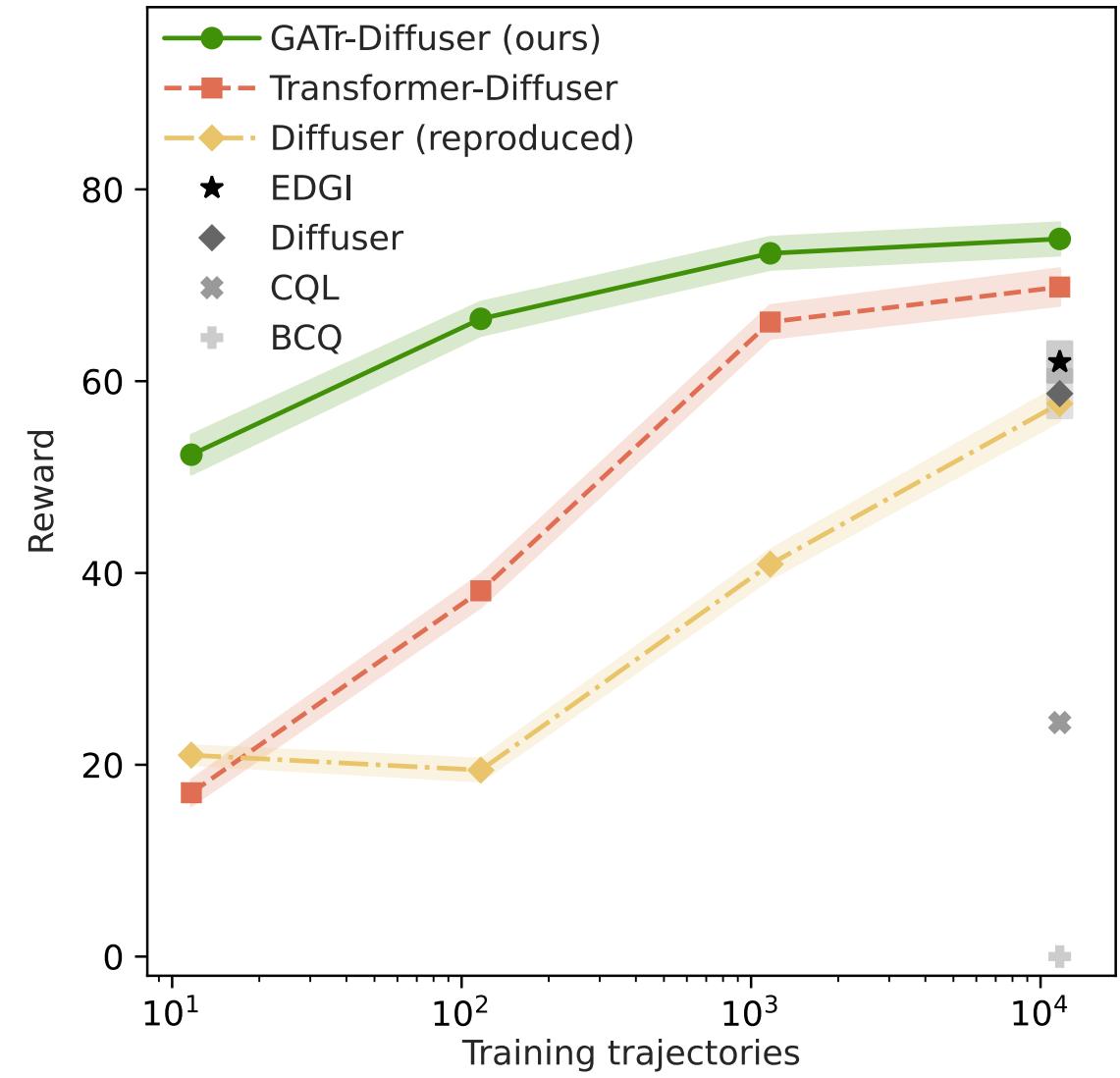
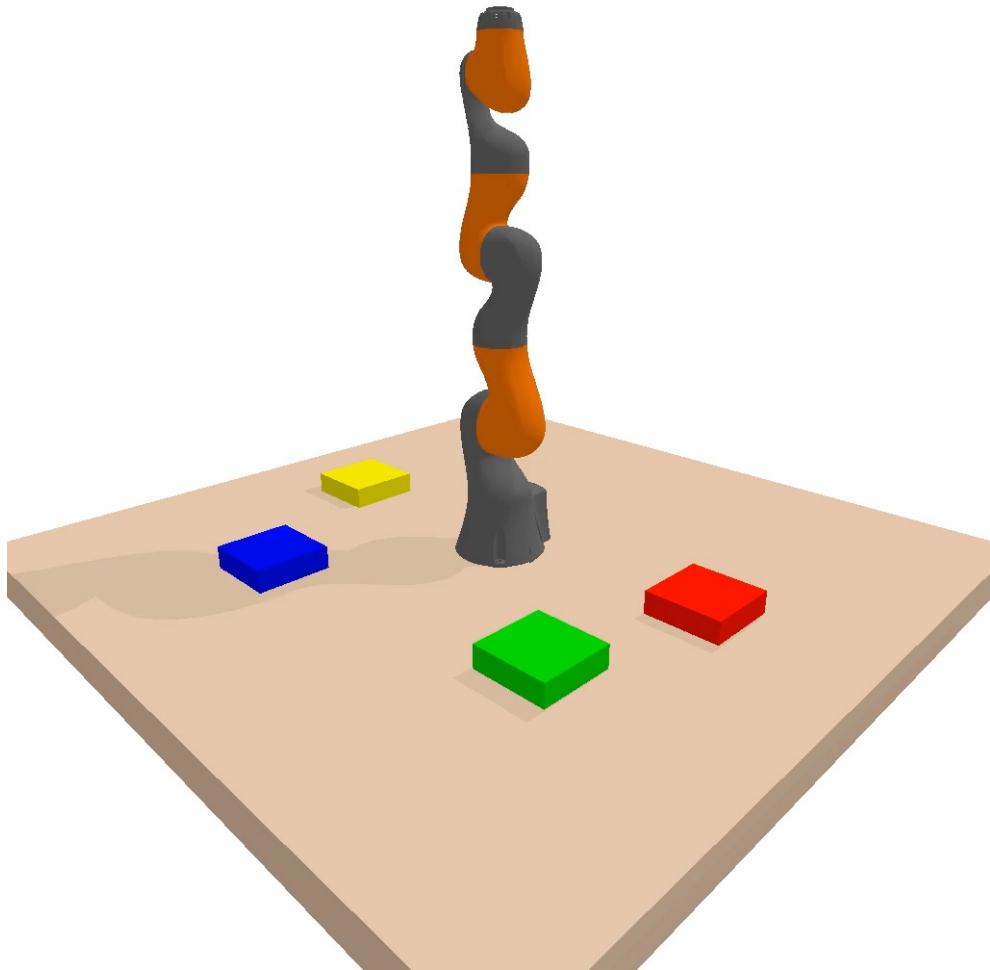
Benchmarking: n-body modelling



Scaling to 7k tokens: Arterial wall-shear stress estimation

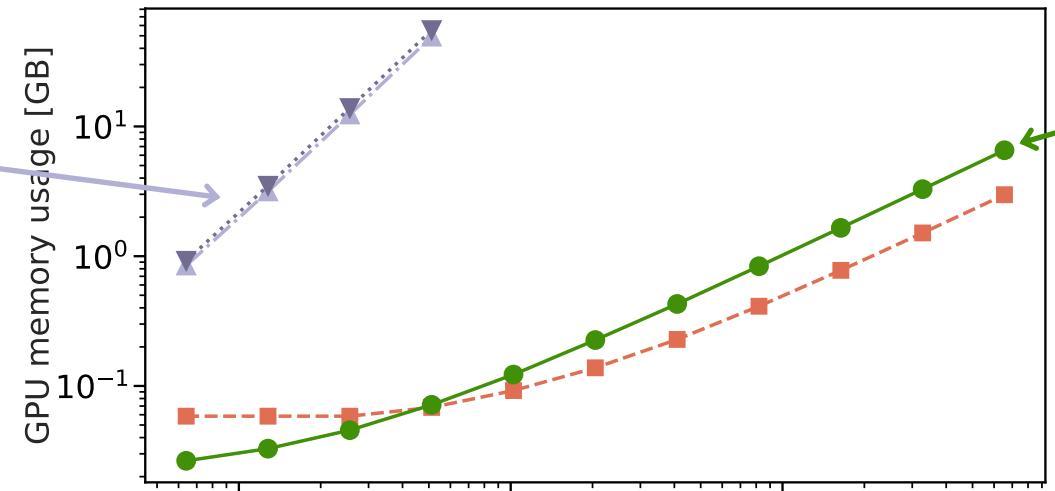


Diffusion-based planning: Robotic control

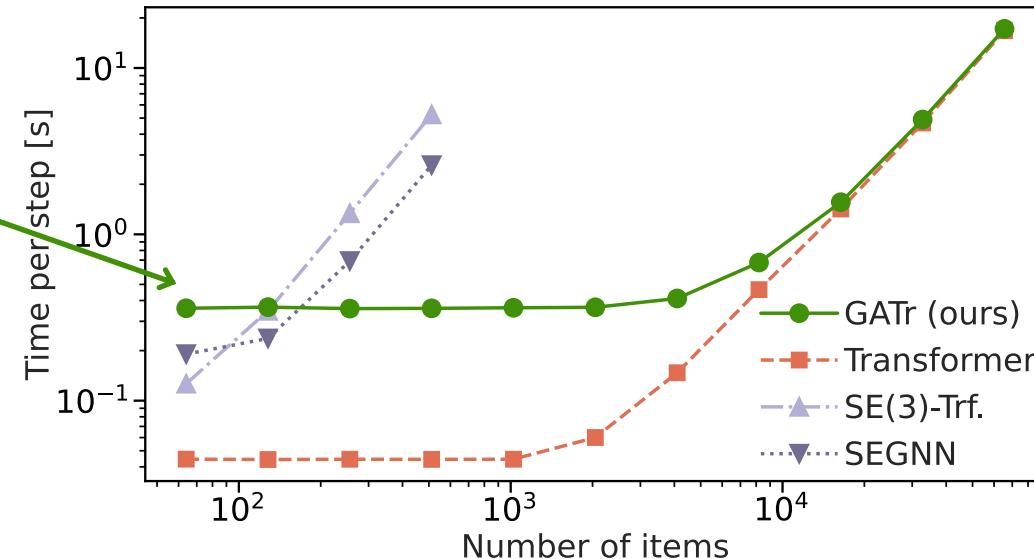


GATr is more scalable than GDL baselines

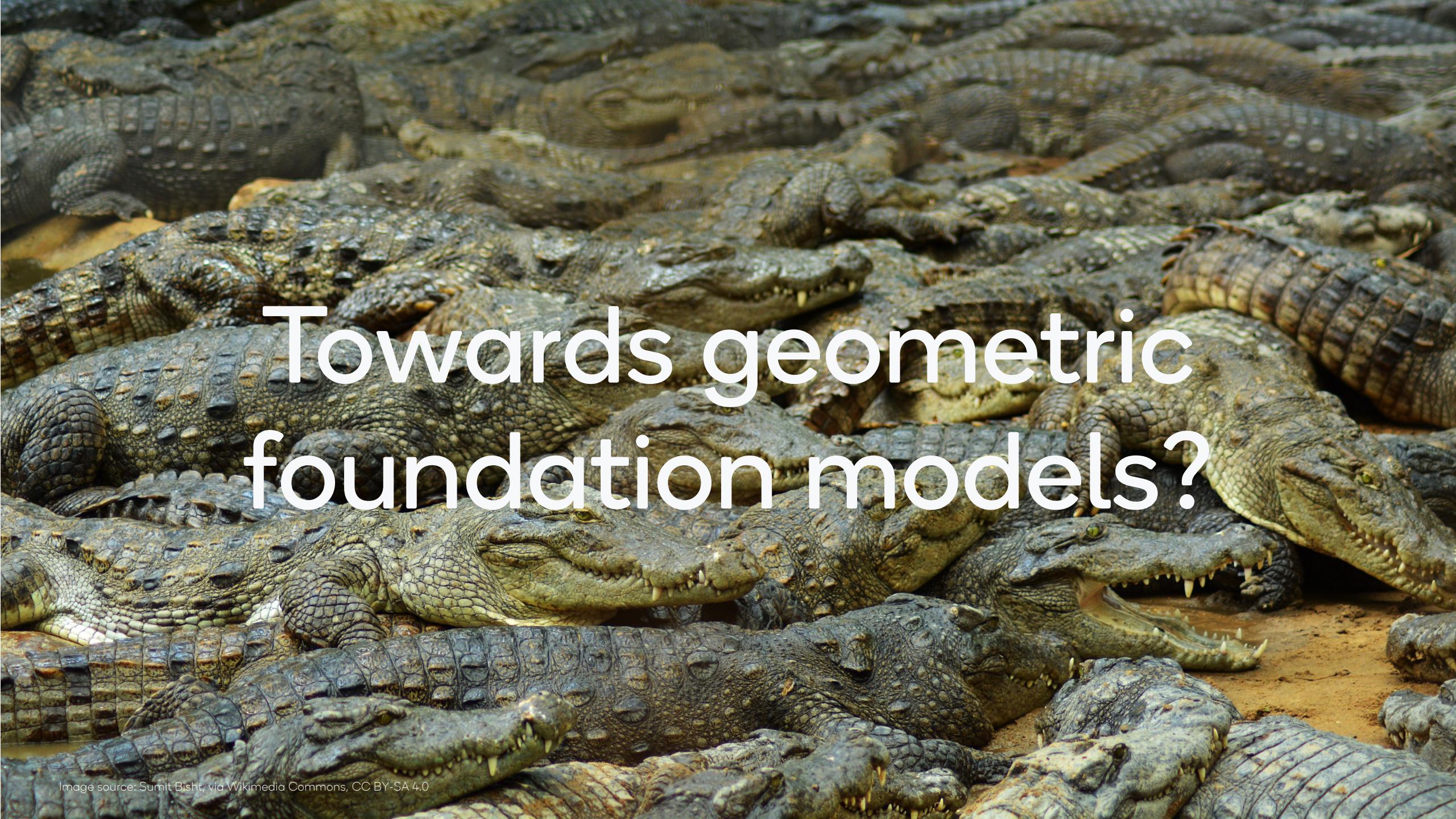
Heavily optimized
Nvidia implementation
of a classic GDL
baseline



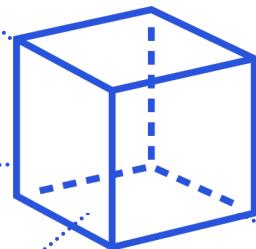
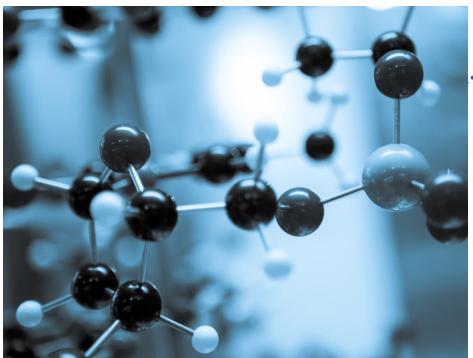
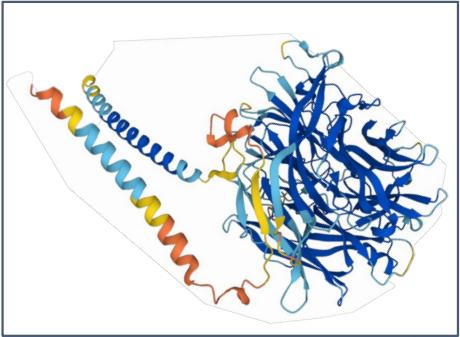
Overhead in small
problems, but still
room for optimization



GATr (with flash
attention) scales like a
transformer, to 10ks
tokens!

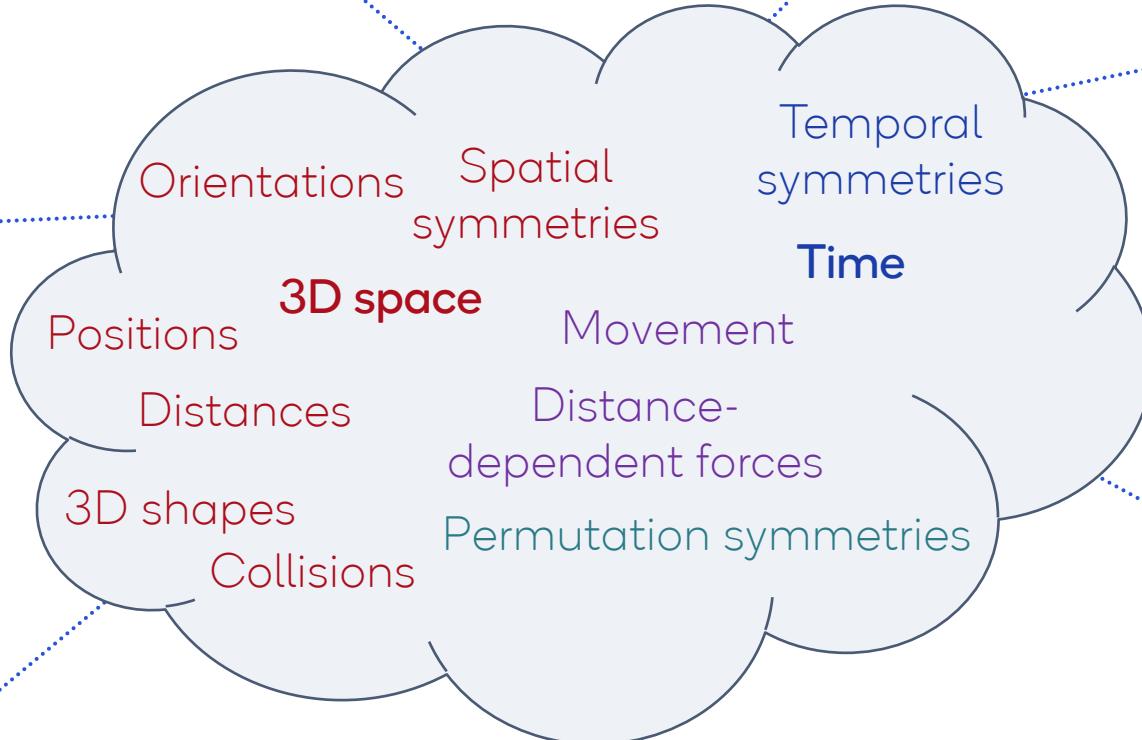
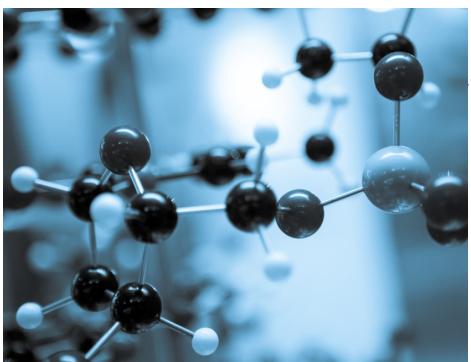
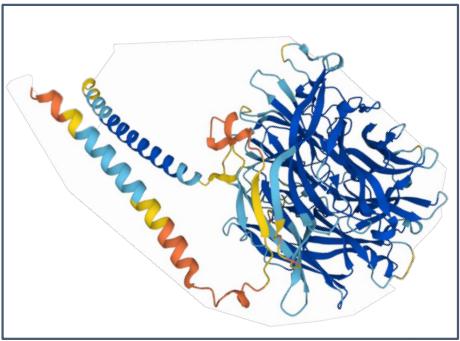
A photograph showing a large number of crocodiles resting on dry, sandy ground. The crocodiles are piled on top of each other, their scaly bodies and heads visible. Some have their mouths open, showing their teeth. The lighting is bright, highlighting the texture of their skin.

Towards geometric foundation models?



Geometric foundation model

- Trained on (and finetuned to) various scientific and engineering problems
- Representations tailored to spatio-temporal data
- Architecture reflects symmetries of 3D space



Foundation model checklist

- Lots of **data** the internet
- Universal **representations** tokenizers and embeddings
- A scalable, expressive **architecture** transformers
- Self-supervised **training** protocol max likelihood, ...
- Multiple **downstream tasks** with shared structure chatbots, ...
- Bonus points for **predictable scaling** neural scaling laws

In language / vision foundation models

Foundation model checklist

- Lots of **data**
- Universal **representations**
- A scalable, expressive **architecture**
- Self-supervised **training** protocol
- Multiple **downstream tasks** with shared structure
- Bonus points for **predictable scaling**

Geometric foundation model



Foundation model checklist

- Lots of **data**
- Universal **representations**
- A scalable, expressive **architecture**
- Self-supervised **training** protocol
- Multiple **downstream tasks** with shared structure
- Bonus points for **predictable scaling**

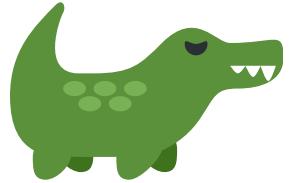
Geometric foundation model

- ?
- could be geometric algebra-based**
- equivariant Transformers**
- ?
- GATr works on very different domains**
- ?

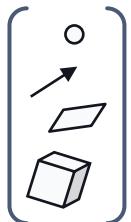
Foundation model checklist

- Lots of **data** data and simulators are out there?
- Universal **representations** could be geometric algebra-based
- A scalable, expressive **architecture** equivariant transformers
- Self-supervised **training** protocol predicting masked-out tokens?
- Multiple **downstream tasks** with shared structure GATr works on very different domains
- Bonus points for **predictable scaling** ?

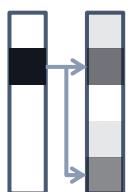
Geometric foundation model



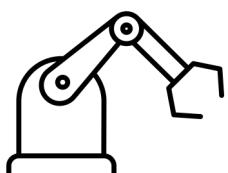
We introduce the **Geometric Algebra Transformer**,
a **versatile architecture for geometric data**



GATr takes into account **geometric structure** through geometric algebra representations and equivariance...



...but has the **scalability** and expressivity of transformers



Initial experiments show a **strong performance**, even from little data

Related and concurrent:

arXiv:2305.11141

Clifford Group Equivariant Neural Networks

David Ruhe

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

Johannes Brandstetter

Microsoft Research
AI4Science
johannesb@microsoft.com

Patrick Forré

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

Abstract

We introduce Clifford Group Equivariant Neural Networks: a novel approach for constructing $E(n)$ -equivariant networks. We identify and study the *Clifford group*, a subgroup inside the Clifford algebra, whose definition we slightly adjust to achieve several favorable properties. Primarily, the group's action forms an orthogonal automorphism that extends beyond the typical vector space to the entire Clifford algebra while respecting the multivector grading. This leads to several non-equivalent subrepresentations corresponding to the multivector decomposition. Furthermore, we prove that the action respects not just the vector space structure of

Geometric Algebra Transformer

Johann Brehmer*, Pim de Haan*, Sönke Behrends, Taco Cohen

*equal contribution

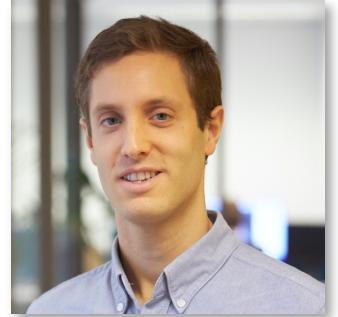
NeurIPS 2023, [arXiv:2305.18415](https://arxiv.org/abs/2305.18415)



Pim de Haan



Sönke Behrends



Taco Cohen

Euclidean, Projective, Conformal:

Choosing a Geometric Algebra for your Equivariant Transformer

Pim de Haan, Taco Cohen, Johann Brehmer

AISTATS 2024, [arXiv:2311.04744](https://arxiv.org/abs/2311.04744)

A Guided Tour to the Plane-Based Geometric Algebra PGA

Leo Dorst, Steven De Keninck

bivector.net/PGA4CS.pdf

Clifford group equivariant neural networks

David Ruhe, Johannes Brandstetter, Patrick Forré

NeurIPS 2023, [arXiv:2305.11141](https://arxiv.org/abs/2305.11141)

Mesh neural networks for SE(3)-equivariant hemodynamics estimation on the artery wall

Julian Suk, Pim de Haan, Phillip Lippe, Christoph Brune, Jelmer M. Wolterink

STACOM workshop 2022, [arXiv:2212.05023](https://arxiv.org/abs/2212.05023)

Planning with Diffusion for Flexible Behavior Synthesis

Michael Janner, Yilun Du, Joshua Tenenbaum, Sergey Levine

ICML 2022, [arXiv:2205.09991](https://arxiv.org/abs/2205.09991)

EDGI: Equivariant Diffusion for Planning with Embodied Agents

Johann Brehmer, Joey Bose, Pim de Haan, Taco Cohen

NeurIPS 2023, [arXiv:2303.12410](https://arxiv.org/abs/2303.12410)

Thank you



Follow us on: [in](#) [Twitter](#) [Instagram](#) [YouTube](#) [Facebook](#)

For more information, visit us at:

qualcomm.com & qualcomm.com/blog

Nothing in these materials is an offer to sell any of the components or devices referenced herein.

©2018-2023 Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved.

Qualcomm is a trademark or registered trademark of Qualcomm Incorporated. Other products and brand names may be trademarks or registered trademarks of their respective owners.

References in this presentation to "Qualcomm" may mean Qualcomm Incorporated, Qualcomm Technologies, Inc., and/or other subsidiaries or business units within the Qualcomm corporate structure, as applicable. Qualcomm Incorporated includes our licensing business, QTL, and the vast majority of our patent portfolio. Qualcomm Technologies, Inc., a subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of our engineering, research and development functions, and substantially all of our products and services businesses, including our QCT semiconductor business. Snapdragon and Qualcomm branded products are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm patented technologies are licensed by Qualcomm Incorporated.