Efficient Equivariant Transformer for Self-Driving Agent Modeling

Scott Xu, Dian Chen, Kelvin Wong, Chris Zhang, Kion Fallah, Raquel Urtasun

waabi · University of Toronto

CVPR JUNE 3-7, 2026 DENVER COLORADO

Symmetries in Traffic Modeling

Motivation: Accurately modeling agent behaviors is an important task with many symmetries, such as equivariance to the order of agents and objects in the scene or equivariance to arbitrary roto-translations of the entire scene as a whole; i.e., SE(2)-equivariance.

Goal: Learn an agent model that guarantees SE(2)-equivariance and efficient scalability to larger scenes and batch sizes.

Existing Approaches

1. Rely on Data Diversity

With enough training examples, hope the model learns an approximately equivariant function.

- ✘ sample and compute inefficient
- ✘ poor generalization as not truly equivariant

2. Encode Invariant Pairwise Features

E.g. transform the context around each agent into its respective coordinate frame and process each agent's context independently, or use pairwise relative positional encoding (RPE) to process all agents simultaneously.

- ✘ computationally expensive
- ✔ equivariant by construction

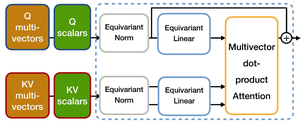
DriveGATr

Input: history of agent states $\mathcal{A}_t = \{a_1^{1:t}, \ldots, a_N^{1:t}\}$

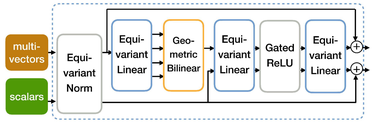
map tokens $\mathcal{M} = \{m_1, \ldots, m_M\}$

Output: an action for each agent

We use 3 kinds of factorized attention blocks: agent-map cross-attention, agent-agent self attention, causal time attention



Equivariant Attention Block



Equivariant MLP Block

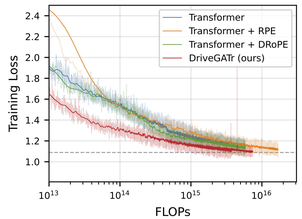
We define primitive equivariant layers for a transformer with multivector features:

- In the **equivariant norm** layer, we divide multivectors by their average norm across the hidden dimension
- In the **equivariant linear layer**, we take SE(2)-equivariant linear combinations of input multivector components
- In the **multivector attention** layer, we compute SE(2)-invariant attention logits by taking inner products between query and key multivectors; implemented via standard dot-product attention

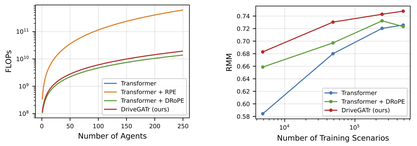
Training Curves

DriveGATr's training loss envelope is significantly lower than those of baselines:

- Transformer uses data augmentation to approximate equivariance
- Transformer + DRoPE uses rotary positional encodings to achieve translation equivariance
- Transformer + RPE uses RPE to achieve SE(2)-equivariance



DriveGATr also displays superior compute efficiency as the number of agents scale, and sample efficiency as the training dataset grows.



Realism

DriveGATr achieves a realism score higher than its baselines, and comparable to state-of-the-art on the Waymo Open Motion Dataset (2024 validation set).

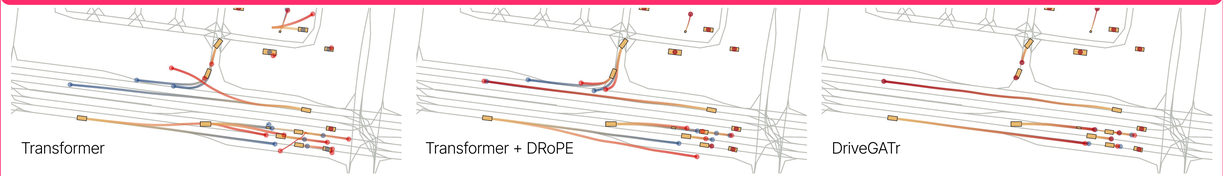
Method	# Params	RMM ↑	minADE ↓
BehaviorGPT	3M	0.7438	1.3804
SMART-7M	7M	0.7678	1.3532
SMART-7M + CAT-K	7M	0.7709	1.2953
Transformer	3M	0.7257	1.6720
Transformer + RPE	3M	0.7251	1.7486
Transformer + DRoPE	3M	0.7206	1.9193
DriveGATr-3M (ours)	3M	0.7620	1.4192
DriveGATr-30M (ours)	30M	0.7636	1.3682

Geometric Algebra Encodings

We present an SE(2)-equivariant transformer grounded in the framework of geometric algebra. Inspired by the Geometric Algebra Transformer (GATr), we encode poses into the 2D projective geometric algebra $\mathbb{R}^*_{2,0,1}$, which allows for a native representation of 2D objects and operators as 8-dimensional multivectors. Multivectors can be scaled, added, or multiplied together using the geometric product.

Geometry / transform	Representation
Line $ax + by + c = 0$	$ae_1 + be_2 + ce_0$
Point (x, y)	$xe_{20} + ye_{01} + e_{12}$
Translation by (a, b)	$1 - \frac{a}{2}e_{01} + \frac{b}{2}e_{20}$
Counterclockwise rotation of angle θ	$\cos\frac{\theta}{2} - \sin\frac{\theta}{2}e_{12}$

Robustness to Roto-translations



We overlay rollouts from the original coordinate frame (blue) vs one rotated by 90° and translated by 100m forward (red). DriveGATr produces consistent trajectories, demonstrating its robustness to roto-translations.