Lorentz-GATr

Lorentz-Equivariant Geometric Algebra Transformers for High-Energy Physics

Jonas Spinner*, Victor Breso*,
Pim de Haan, Tilman Plehn,
Jesse Thaler, Johann Brehmer

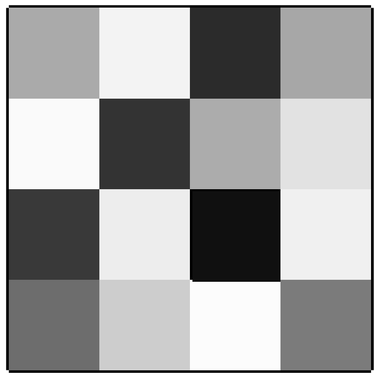
ML Journal Club
RWTH Aachen

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

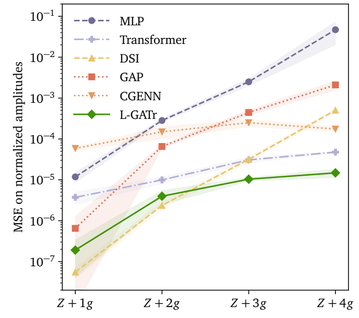
Lorentz symmetry is key in high-energy physics…

$$\mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu}$$
$$+ i \not{F} \not{D} \not{F} + h.c$$
$$+ \not{F}_i y_{ij} \not{F}_j \phi + h.c$$
$$+ |D_\mu \phi|^2 - V(\phi)$$

… so let’s build it into our neural networks



$$\begin{pmatrix} 1 \\ \gamma^\mu \\ \sigma^{\mu\nu} \\ \gamma^\mu\gamma_5 \\ \gamma_5 \end{pmatrix}$$



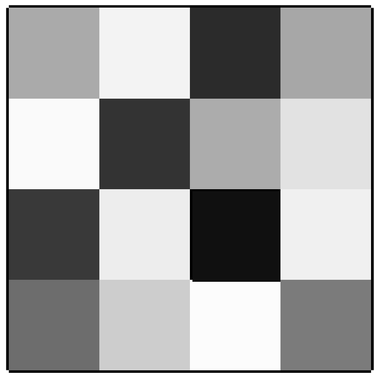
Transformer
architecture

Geometric algebra
representations

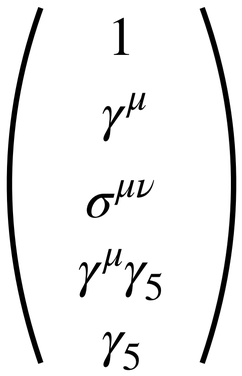
Lorentz-Equivariant
Geometric **A**lgebra
Transformer

Strong performance
on diverse problems

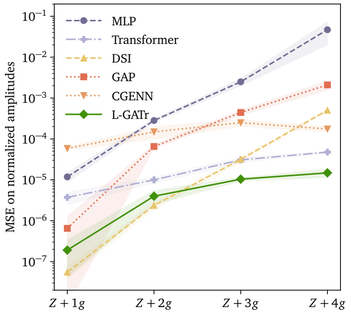
GATr was originally developed for E(3)
arXiv:2305.18415



+



=



Transformer
architecture

Geometric algebra
representations

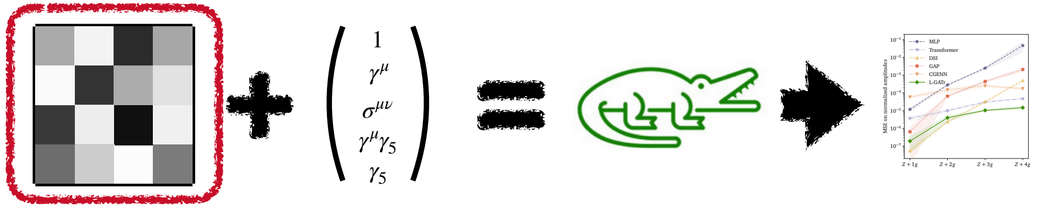
Lorentz-Equivariant
Geometric **A**lgebra
Transformer

Strong performance
on diverse problems

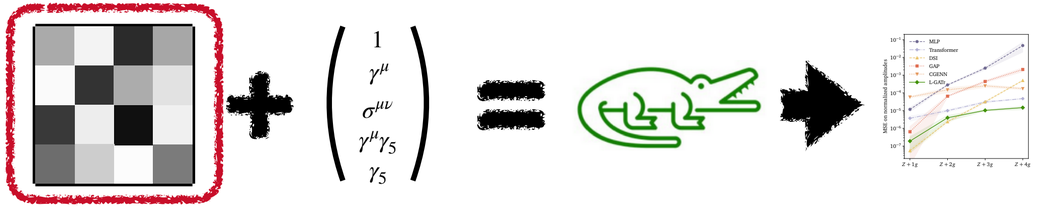
GATr was originally developed for E(3)
arXiv:2305.18415

Transformers

Point clouds



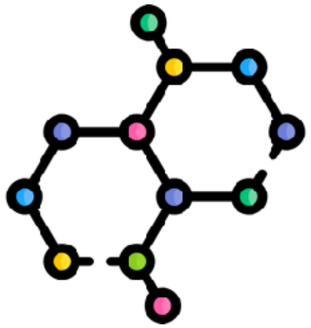
	UNSTRUCTURED	GRID	POINT CLOUD
Structure			
Network Architecture	MLP	Convolutional NN (CNN)	Graph NN (GNN)/ Transformer
Inductive Bias	-	Locality	Permutation symmetry Variable-size inputs Pairwise relations



Transformers

Point clouds

Molecules

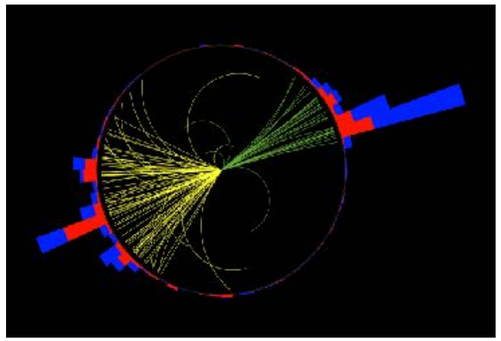


Calorimeter cells

Words

Attention is all you need .

Jets



Events

$$q\bar{q} \rightarrow e^+e^-\gamma$$

Patches in an image

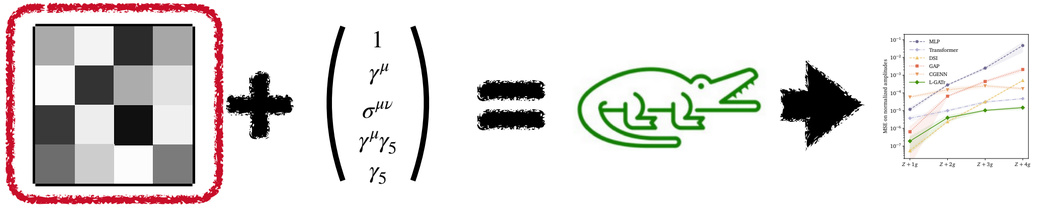
POINT CLOUD

Graph NN (GNN)/ Transformer

Permutation symmetry
Variable-size inputs
Pairwise relations

Transformers

Point cloud representations



x_i — embed into latent space → x_{ic}

‘Token’ index $i = 1 \ldots n$

(n can be different for every sample)

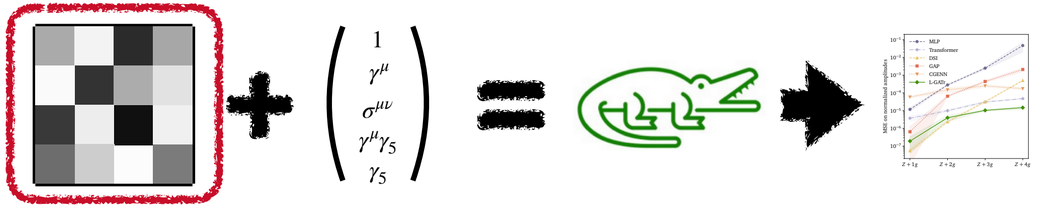
‘Latent’ index $\alpha = 1 \ldots d$

(d fixed)

Data has shape
(batchsize, #tokens, #latent dims)

Transformers

Attention

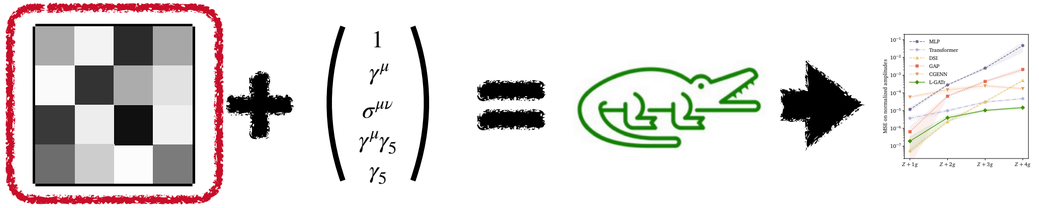


Want a simple update operation on point clouds $x'_{ic} = f_{ic}(x)$

Exchange information along ‘latent’ index c → Linear layer

$$x'_{ic} = W^V_{cc'} x_{ic'}$$

Transformers



Attention

Want a simple update operation on point clouds $x'_{ic} = f_{ic}(x)$

Exchange information along 'token' index i (without breaking permutation symmetry) → Attention matrix

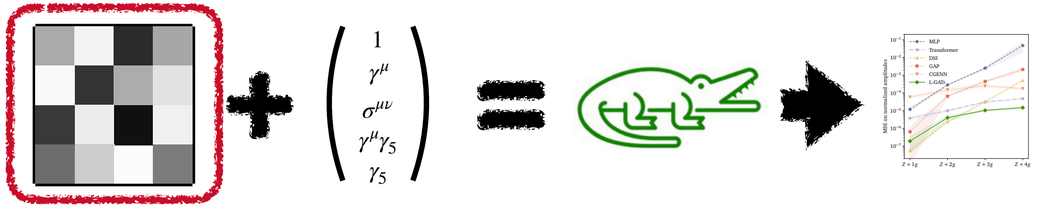
$$x'_{ic} = A_{ij}(x) W^V_{cc'} x_{jc'}$$

Exchange information along 'latent' index c → Linear layer

$$A_{ij}(x) = \mathrm{Softmax}_j\left(\frac{x_{ic} W^Q_{cc'} W^K_{c'c''} x_{jc''}}{\sqrt{d}}\right)$$

More sophisticated update operations: Message-Passing Graph Networks

Transformers

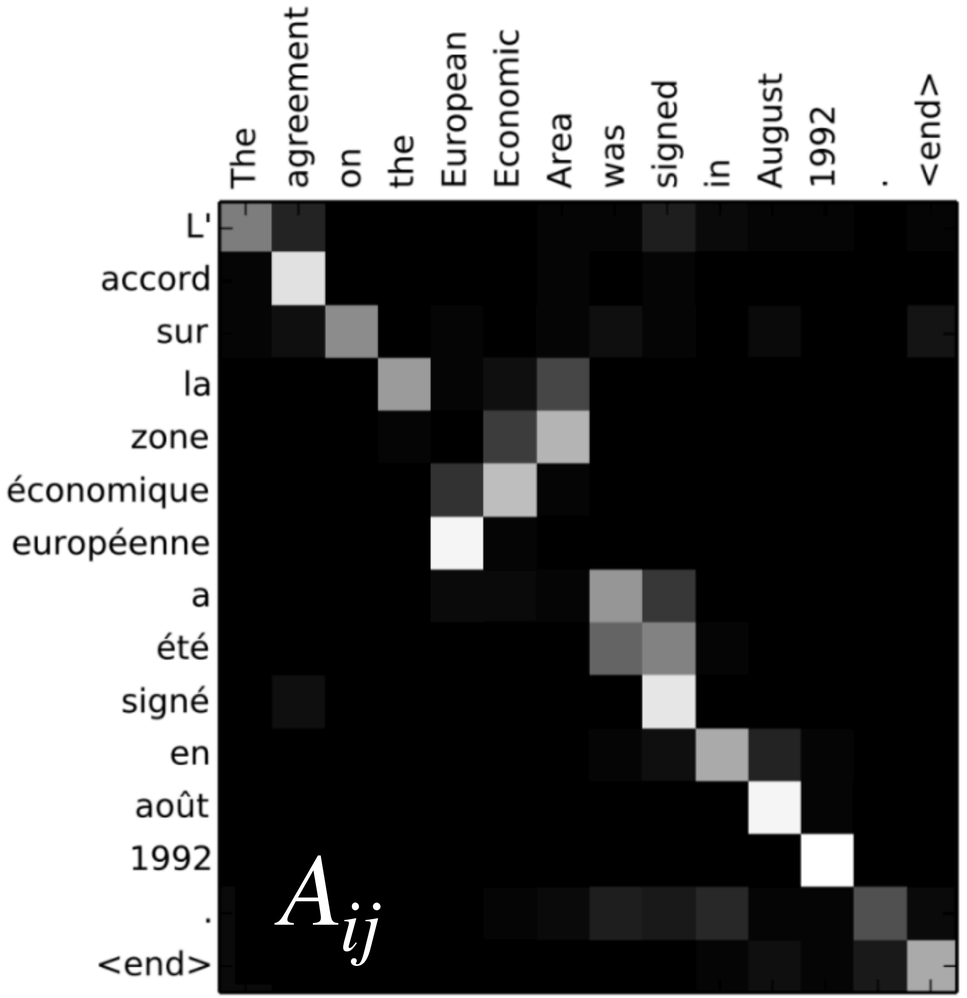


Attention matrix A_{ij} - visualised

Self-attention

	Hello	I	love	you
Hello	0.8	0.1	0.05	0.05
I	0.1	0.6	0.2	0.1
love	0.05	0.2	0.65	0.1
you	0.2	0.1	0.1	0.6

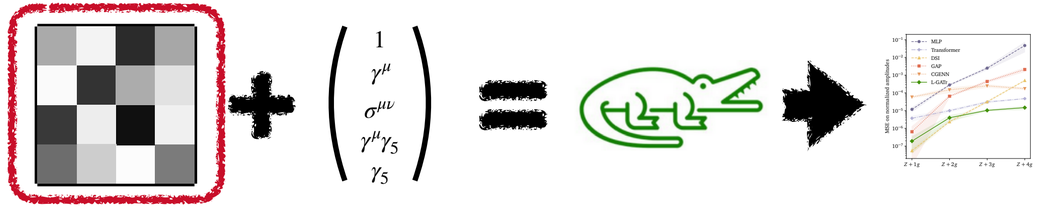
Cross-attention



Transformers

Attention is all you need

- Transformer = Network architecture based solely on attention mechanisms
- Proposed 2017 in ‘Attention is all you need’ (already 130k citations)
- Established as the most scalable point cloud architecture



1706.03762

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

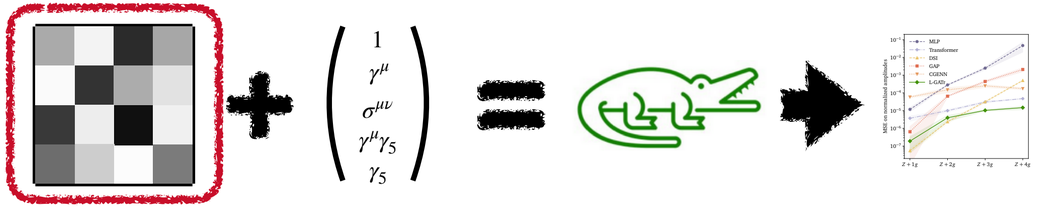
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

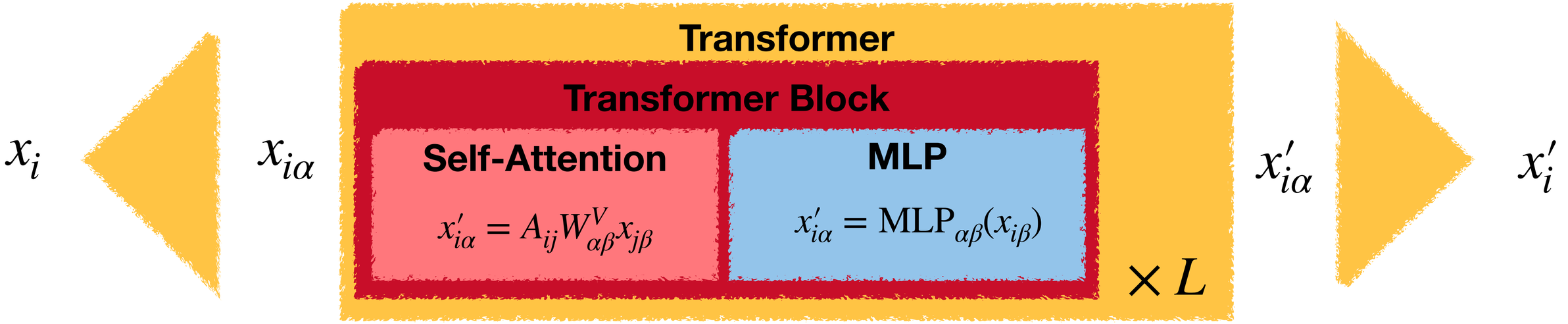
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to

Transformers

Architecture overview





Embed into latent space

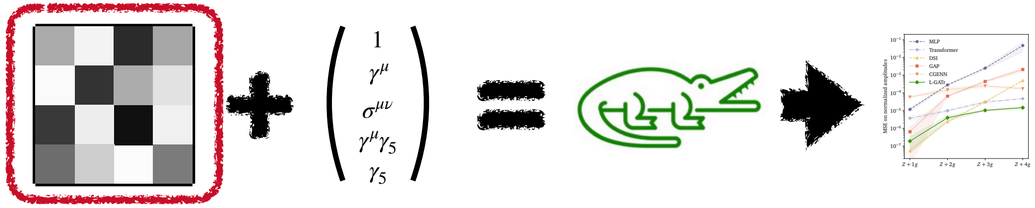
‘Exchange information between tokens’

Nonlinearity + ‘Clean up latent space’

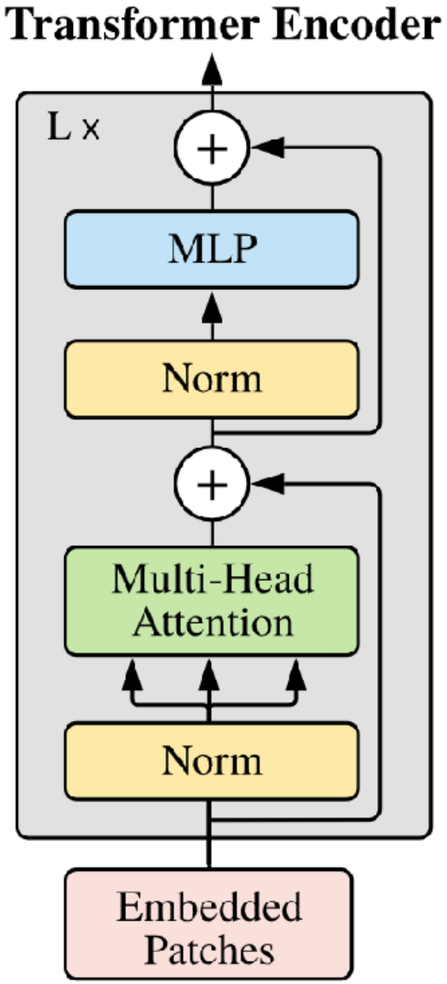
Extract from latent space

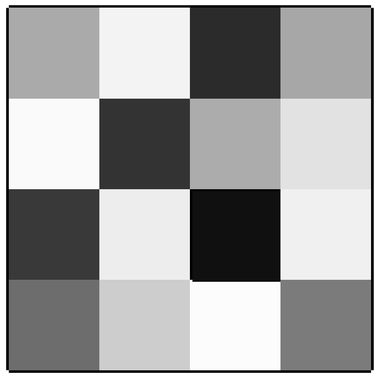
Transformers

Implementation

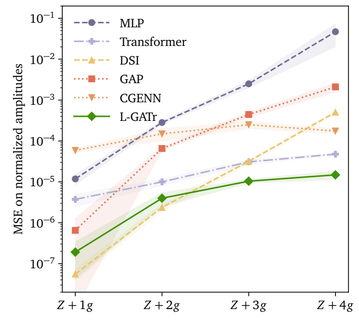


- **Residual connections** ⇒ Helps for deep networks
- **LayerNorm** ⇒ Numerical stability
- **Dropout** before residual connections ⇒ Regularisation
- Off-the-shelf implementation:
  `torch.nn.TransformerEncoderLayer`





$$\begin{pmatrix} 1 \\ \gamma^\mu \\ \sigma^{\mu\nu} \\ \gamma^\mu\gamma_5 \\ \gamma_5 \end{pmatrix}$$



Transformer
architecture

Geometric algebra
representations

Lorentz-Equivariant
Geometric **A**lgebra
Transformer

Strong performance
on diverse problems

Geometric algebra

Geometric algebra = Clifford algebra

Geometric algebra = Vector space + geometric product $xy = \frac{\{x, y\}}{2} + \frac{[x, y]}{2}$

- Symmetric part $\{x, y\}$: scalar/inner product
- Antisymmetric part $[x, y]$: outer product (yields higher-order objects)

Spacetime algebra: Geometric algebra over vector space $\mathbb{R}^4$

- Basis elements γ^μ are orthonormal: $\{\gamma^\mu, \gamma^\nu\} = 2g^{\mu\nu}$
- The Dirac algebra is the same up to $\mathbb{R} \to \mathbb{C}$

Geometric algebra

Building multivectors

- Scalar and vectors $1,\ \gamma^\mu$ (1+4 objects)
- Product of two vectors: $\gamma^\mu\gamma^\nu = \frac{\{\gamma^\mu,\gamma^\nu\}}{2} + \frac{[\gamma^\mu,\gamma^\nu]}{2} = g^{\mu\nu} + \sigma^{\mu\nu}$ (6 new objects)
- Axial vector: $\epsilon_{\mu\nu\rho\sigma}\gamma^\nu\gamma^\rho\gamma^\sigma$ (4 new objects)
- Pseudoscalar: $\gamma^5 = \gamma^0\gamma^1\gamma^2\gamma^3 = \frac{1}{4!}\epsilon_{\mu\nu\rho\sigma}\gamma^\mu\gamma^\nu\gamma^\rho\gamma^\sigma$ (1 new object)

Multivector: $x = x^S 1 + x^V_\mu\gamma^\mu + x^B_{\mu\nu}\sigma^{\mu\nu} + x^A_\mu\gamma^\mu\gamma^5 + x^P\gamma^5$ with $(x^S, x^V_\mu, x^B_{\mu\nu}, x^A_\mu, x^P) \in \mathbb{R}^{16}$

Geometric algebra

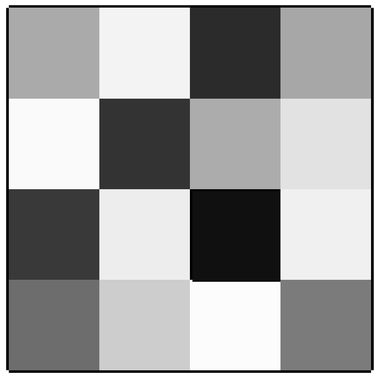
Physics with multivectors

- Particle: $x = p_\mu \gamma^\mu$
- Boost in z direction: $v = e^{\omega\sigma^{03}} = 1\cosh\omega + \sigma^{03}\sinh\omega$
- Boost applied to particle moving in z direction $x = E\gamma^0 + p_z\gamma^3$:

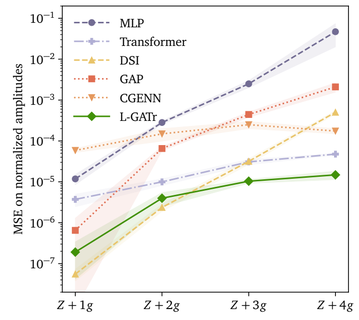
$$vxv^{-1} = (1\cosh\omega + \sigma^{03}\sinh\omega)(E\gamma^0 + p_z\gamma^3)(1\cosh\omega - \sigma^{03}\sinh\omega)$$

$$= (E\cosh\omega - p_z\sinh\omega)\gamma^0 + (p_z\cosh\omega - E\sinh\omega)\gamma^3$$

$$= E^{\text{boosted}}\gamma^0 + p_z^{\text{boosted}}\gamma^3$$



$$\begin{pmatrix} 1 \\ \gamma^\mu \\ \sigma^{\mu\nu} \\ \gamma^\mu\gamma_5 \\ \gamma_5 \end{pmatrix}$$



Transformer
architecture

Geometric algebra
representations

Lorentz-Equivariant
Geometric **A**lgebra
Transformer

Strong performance
on diverse problems

L-GATr

Geometric algebra representations

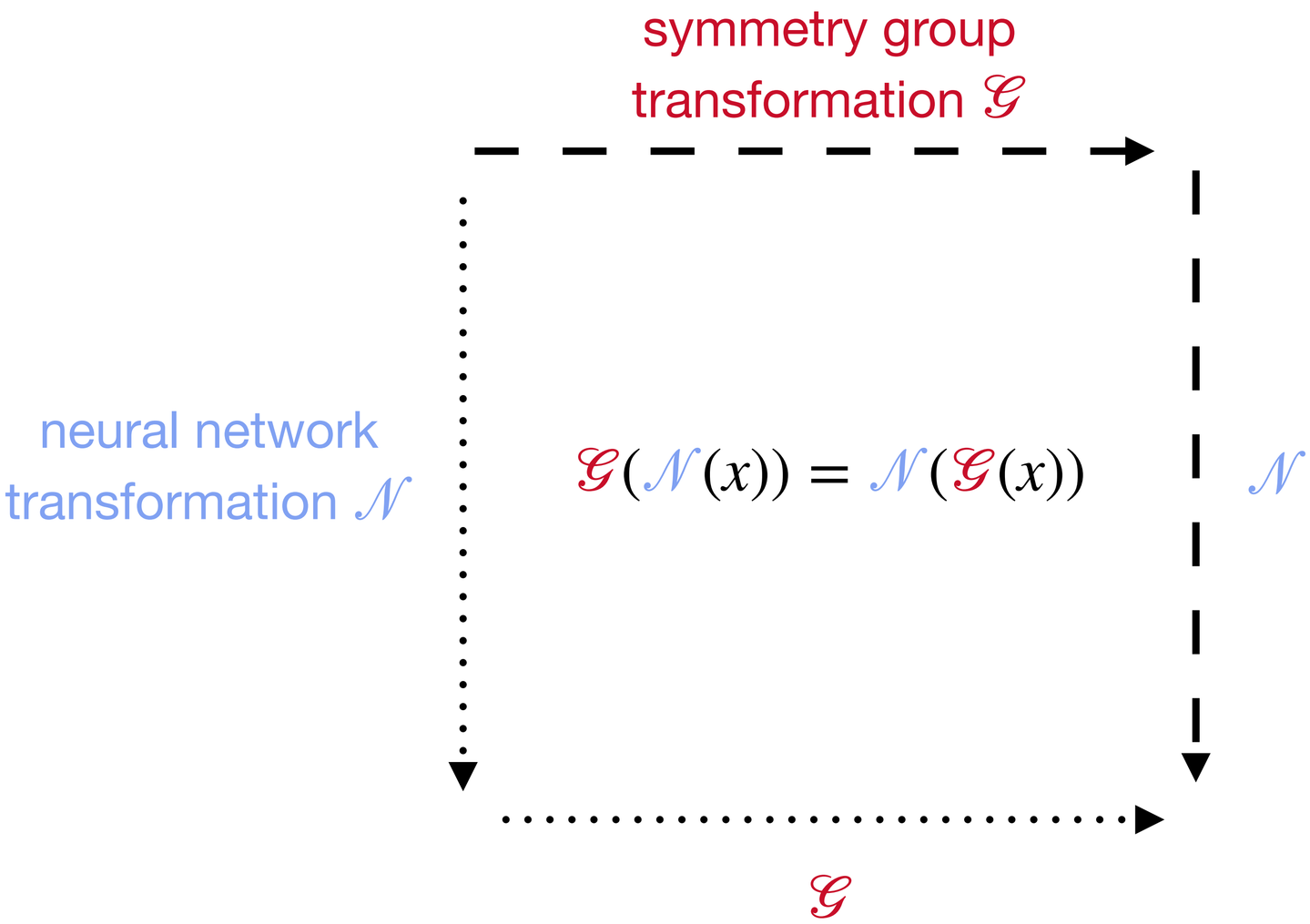
$$x^S \quad \Rightarrow \quad \begin{pmatrix} x^S \\ x^V_\mu \\ x^B_{\mu\nu} \\ x^A_\mu \\ x^P \end{pmatrix} \in \mathbb{R}^{16}$$

scalars

multivectors
(+ extra scalars)

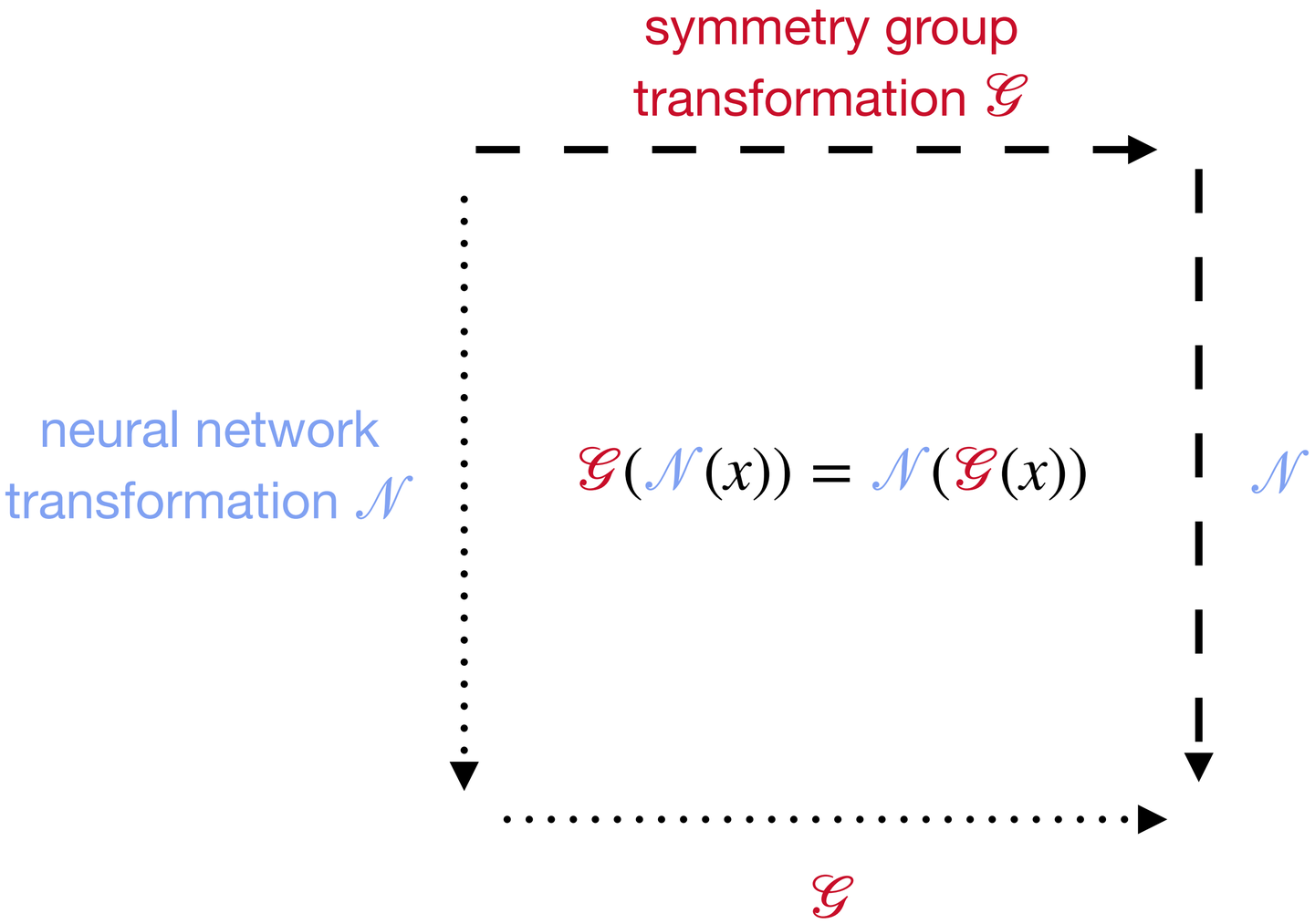
L-GATr

Equivariance



L-GATr

Equivariance



Equivariance on geometric algebra representations:

Multivector grades form subresentations of the Lorentz group → All components within the same grade have to transform equally

$$\begin{pmatrix} x^S \\ x^V_\mu \\ x^B_{\mu\nu} \\ x^A_\mu \\ x^P \end{pmatrix} \xrightarrow{\mathcal{G}} \begin{pmatrix} w^S x^S \\ w^V x^V_\mu \\ w^B x^B_{\mu\nu} \\ w^A x^A_\mu \\ w^P x^P \end{pmatrix}$$

L-GATr

GATr-ing all transformer layers

Layer type	Transformer	L-GATr
$\mathrm{Linear}(x)$	$vx + w$	
$\mathrm{Attention}(q,k,v)_{ic}$	$\sum_{j,c'} \mathrm{Softmax}_j\left(\frac{q_{ic'}k_{jc'}}{\sqrt{n_c}}\right)v_{jc}$	
$\mathrm{LayerNorm}(x)$	$x\left[\frac{1}{n_c}\sum_{c=1}^{n_c} x_c^2 + \epsilon\right]^{-1/2}$	
$\mathrm{Activation}(x)$	$\mathrm{GELU}(x)$	

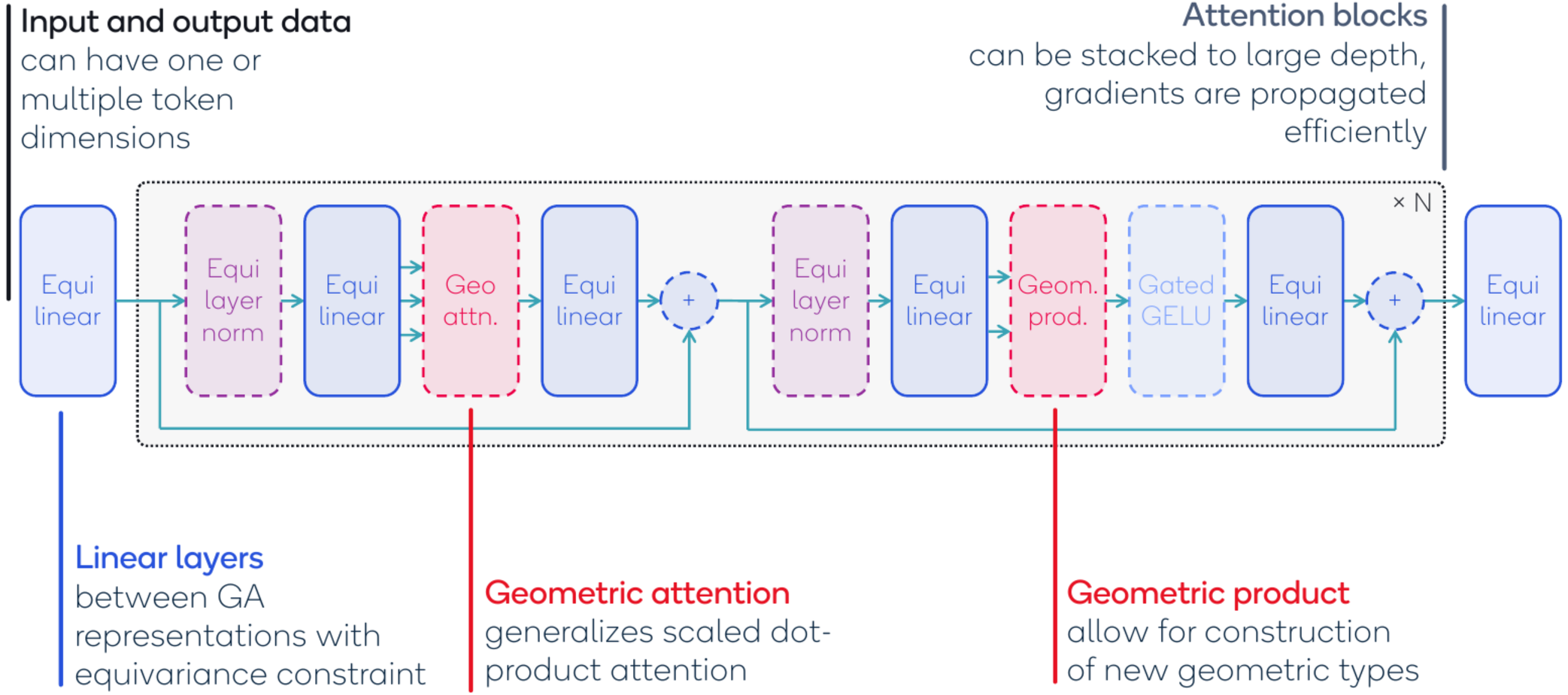
L-GATr

GATr-ing all transformer layers

Layer type	Transformer	L-GATr
Linear(x)	$vx + w$	$\sum_{k=0}^{4} v_k \langle x \rangle_k + \sum_{k=0}^{4} w_k \gamma^5 \langle x \rangle_k$
Attention$(q,k,v)_{ic}$	$\sum_{j,c'} \mathrm{Softmax}_j \left(\frac{q_{ic'} k_{jc'}}{\sqrt{n_c}} \right) v_{jc}$	$\sum_{j,c'} \mathrm{Softmax}_j \left(\frac{\langle q_{ic'}, k_{jc'} \rangle}{\sqrt{16 n_c}} \right) v_{jc}$
LayerNorm(x)	$x \left[\frac{1}{n_c} \sum_{c=1}^{n_c} x_c^2 + \epsilon \right]^{-1/2}$	$\mathrm{x} \left[\frac{1}{n_c} \sum_{c=1}^{n_c} \sum_{k=0}^{4} \left\lvert \left\langle \langle x_c \rangle_k, \langle x_c \rangle_k \right\rangle \right\rvert + \epsilon \right]^{-1/2}$
Activation(x)	GELU(x)	GELU$(\langle x \rangle_0) x$
GP(x,y)	—	xy

L-GATr

Full architecture



Credits to Johann Brehmer

L-GATr

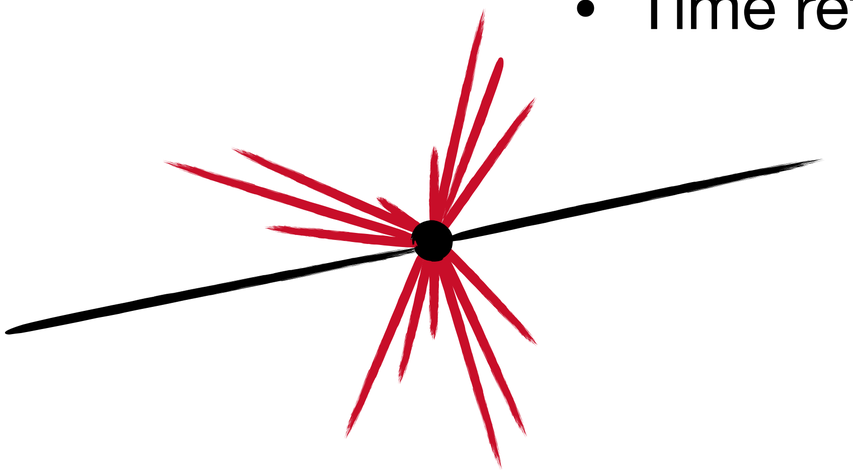
Symmetry breaking with spurions

Lorentz symmetry is rarely exact

- Beam direction in collider
- Detector effects
- …?

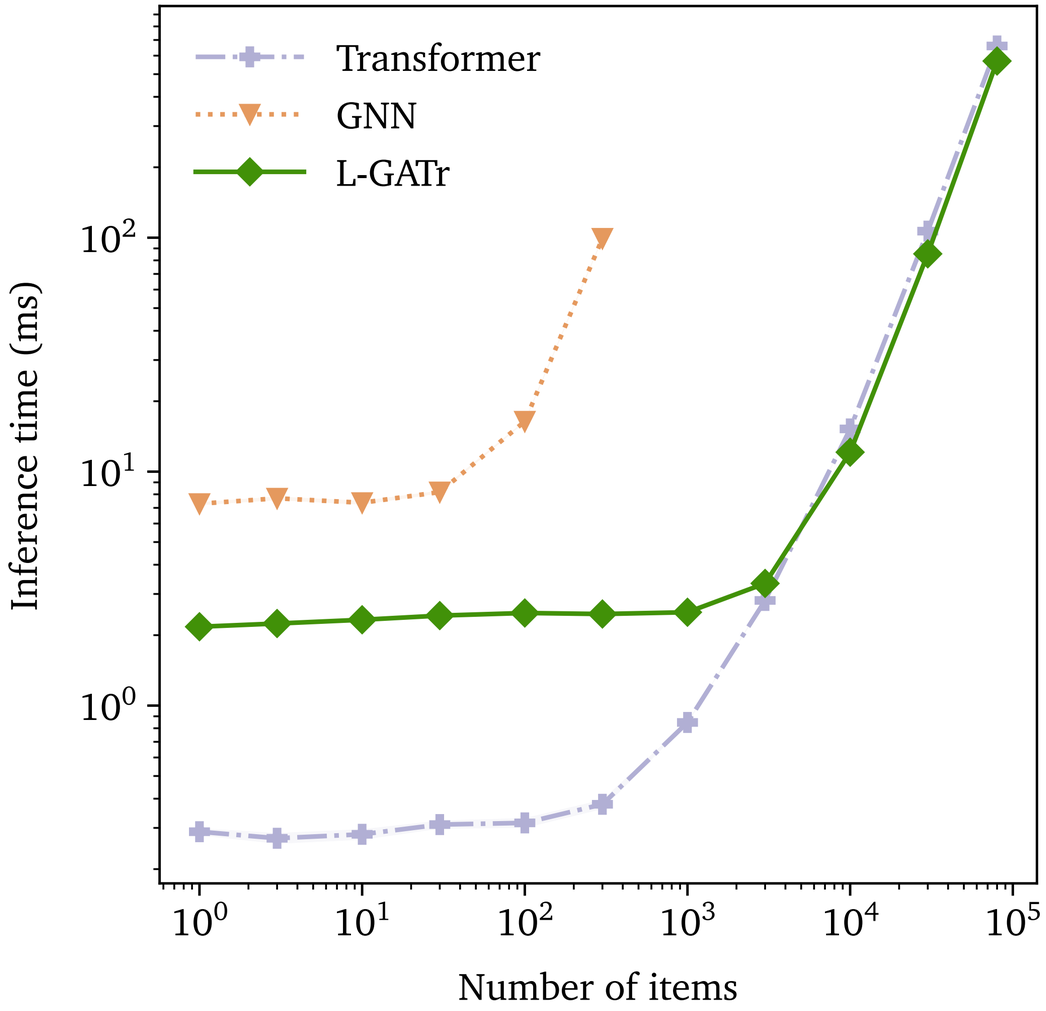
Add a spurion to the particle list (either as token or channel)

- Beam reference: $p^\mu = (0,0,0,\pm 1)$
- Time reference: $p^\mu = (1,0,0,0)$

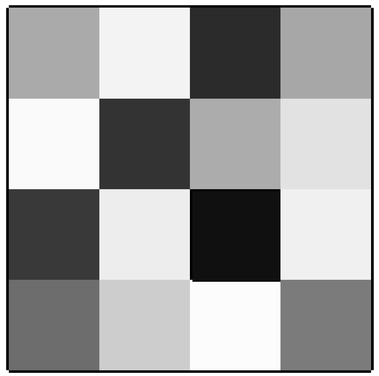


L-GATr

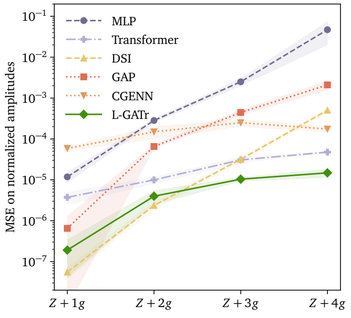
Processing thousands of particles



Transformers scale
better than graph networks



$$\begin{pmatrix} 1 \\ \gamma^\mu \\ \sigma^{\mu\nu} \\ \gamma^\mu\gamma_5 \\ \gamma_5 \end{pmatrix}$$



Transformer
architecture

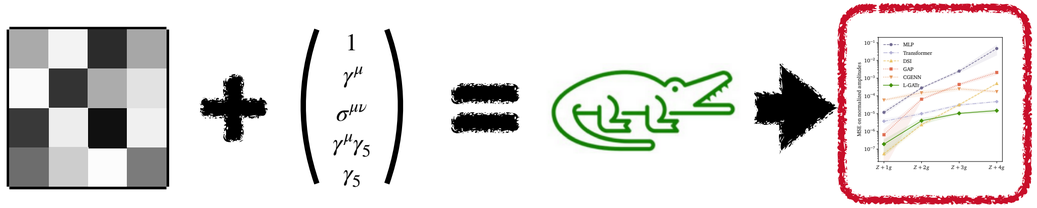
Geometric algebra
representations

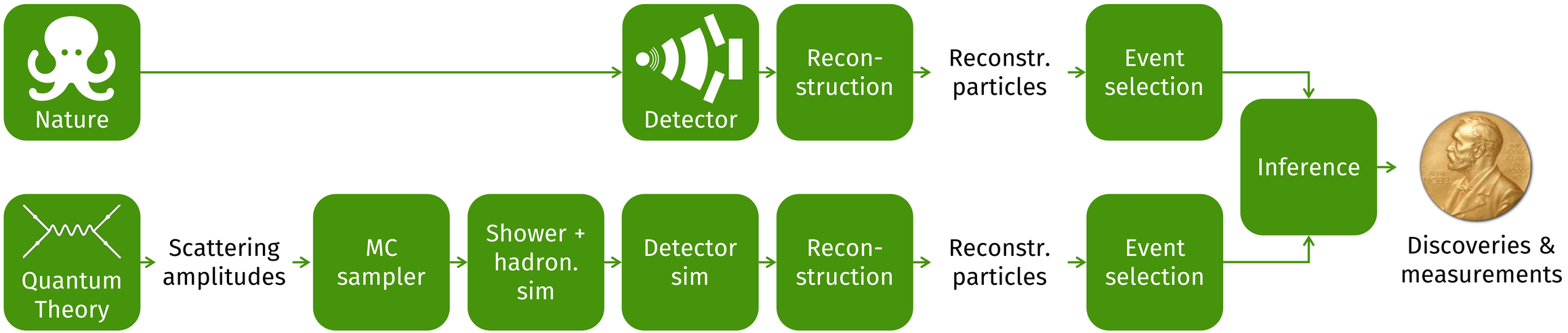
Lorentz-Equivariant
Geometric **A**lgebra
Transformer

Strong performance
on diverse problems

Experiments

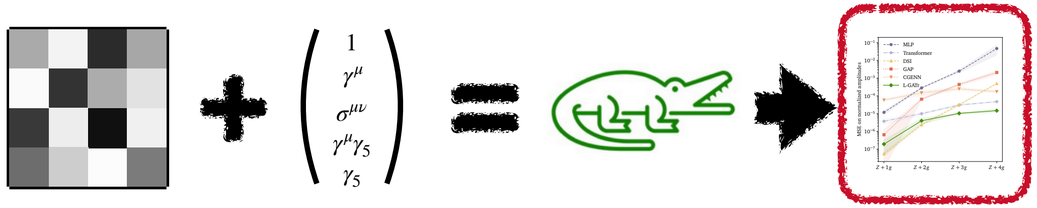
LHC simulation chain





Experiments

LHC simulation chain meets ML



Calibration, clustering etc

Simulation-based inference: MadMiner, MEM etc

Nature

Detector

Recon-struction

Reconstr. particles

Event selection

Inference

Discoveries & measurements

Quantum Theory

Scattering amplitudes

MC sampler

Shower + hadron. sim

Detector sim

Recon-struction

Reconstr. particles

Event selection

Amplitude regression

Neural importance sampling

Generative networks

Jet tagging

More applications

Anomaly detection

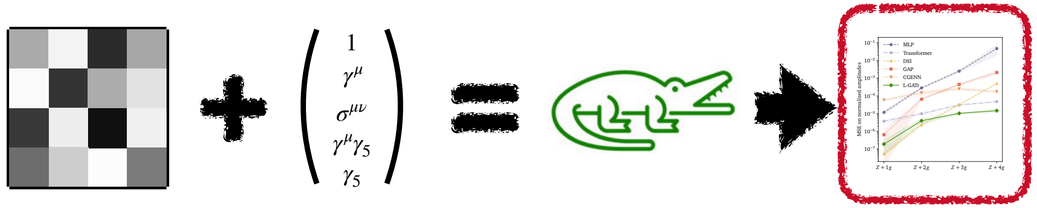
Unfolding

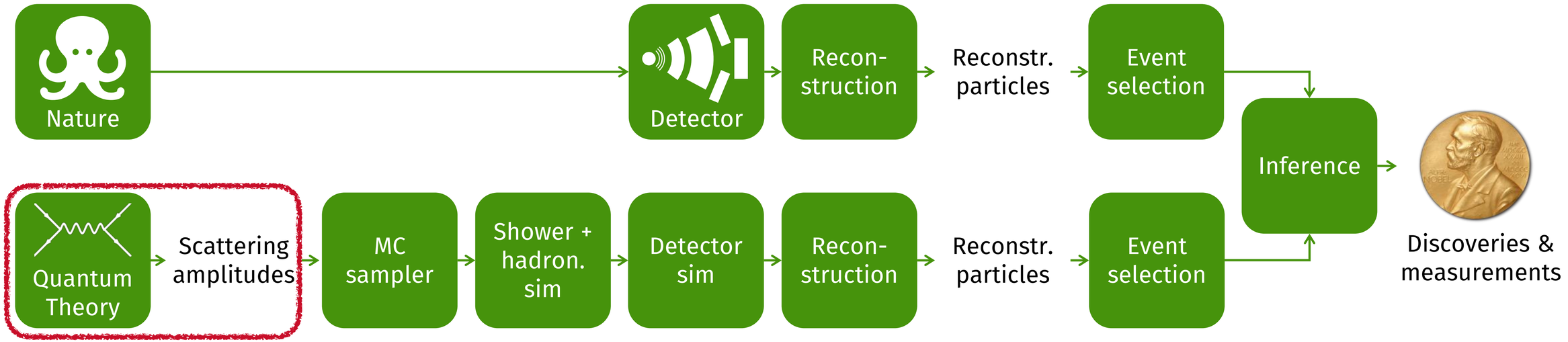
…

Credits to Theo Heimel

Experiments

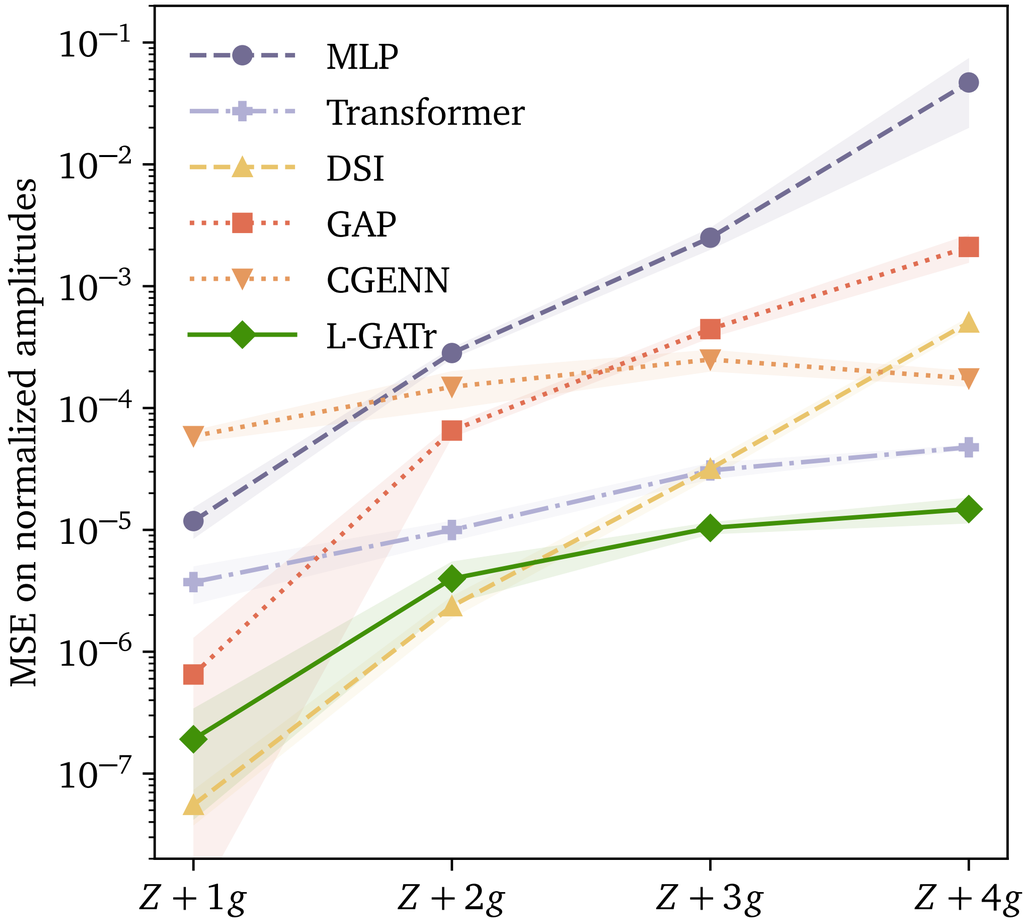
Amplitude regression





Experiments

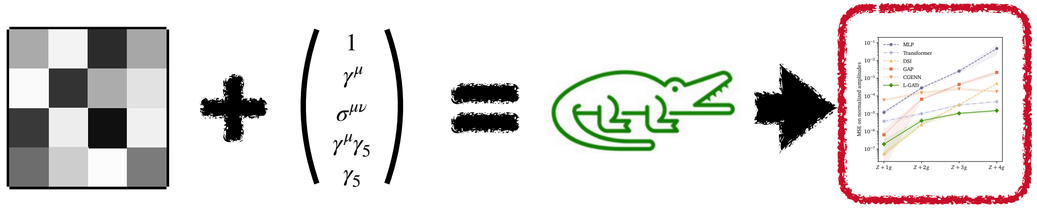
Amplitude regression

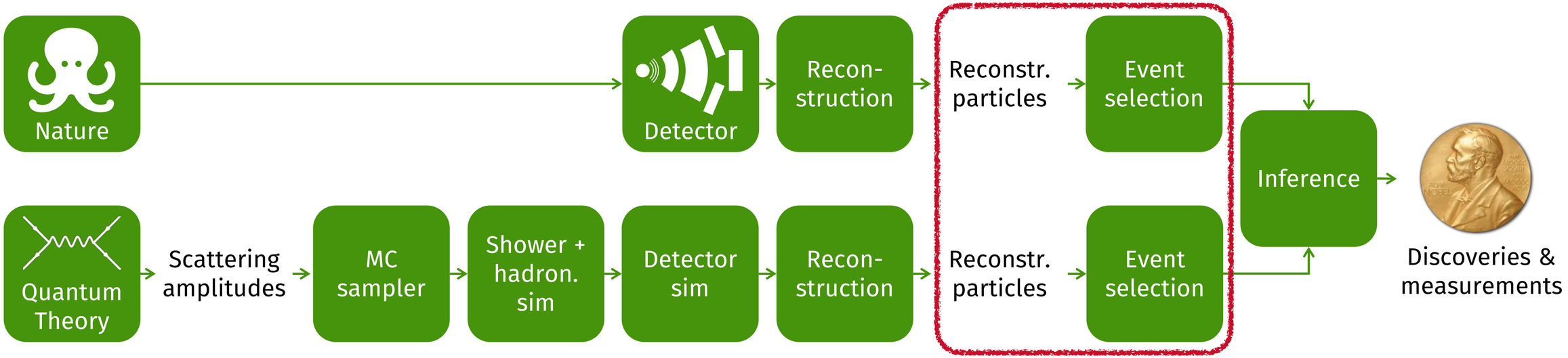


L-GATr scales best to high multiplicity,
where amplitude surrogates are most useful

Experiments

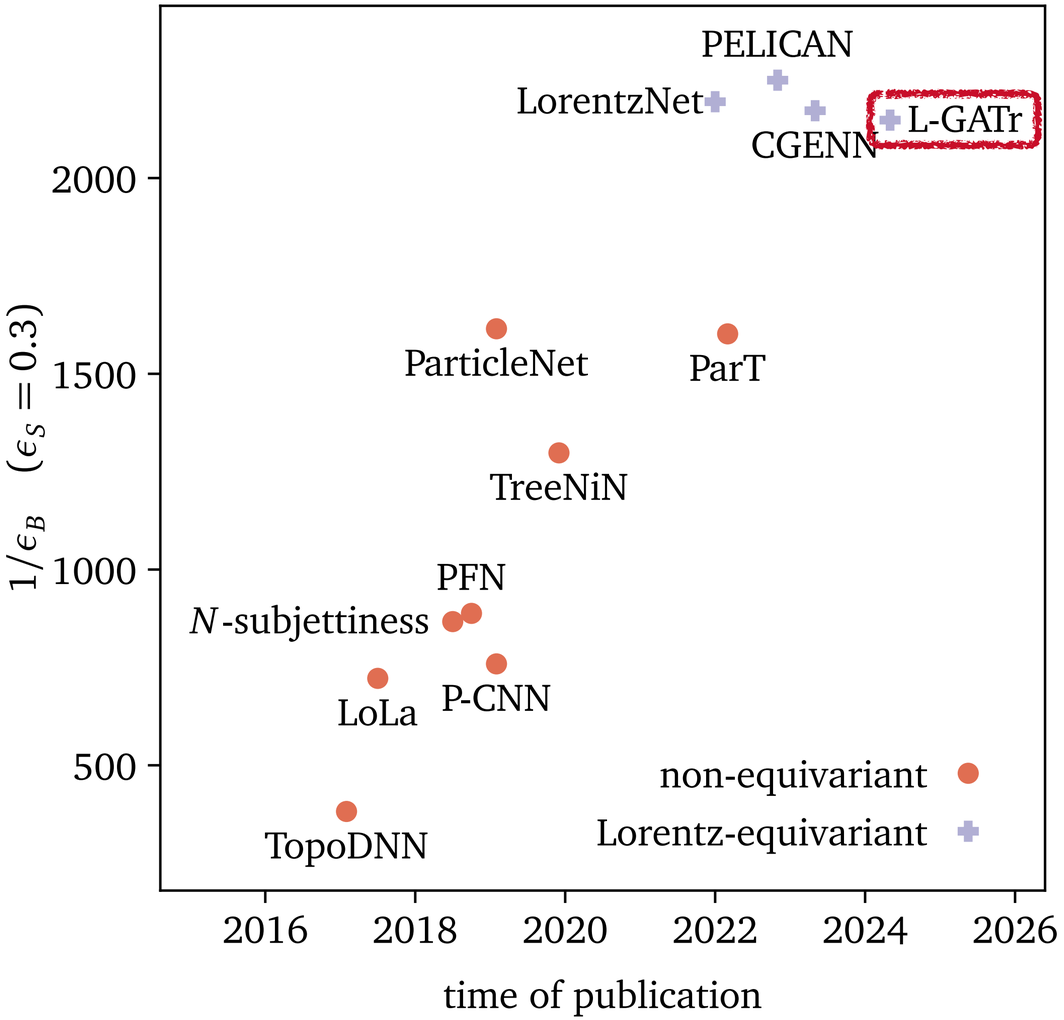
Top tagging





Experiments

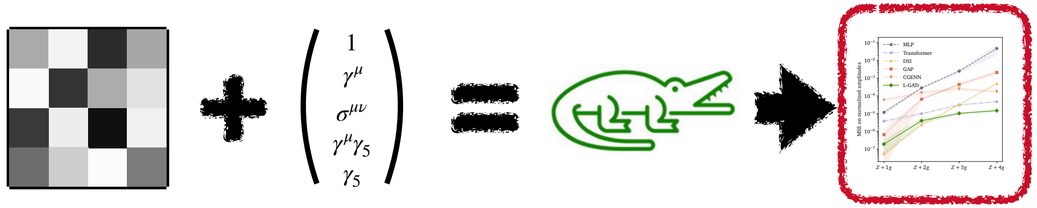
Top tagging

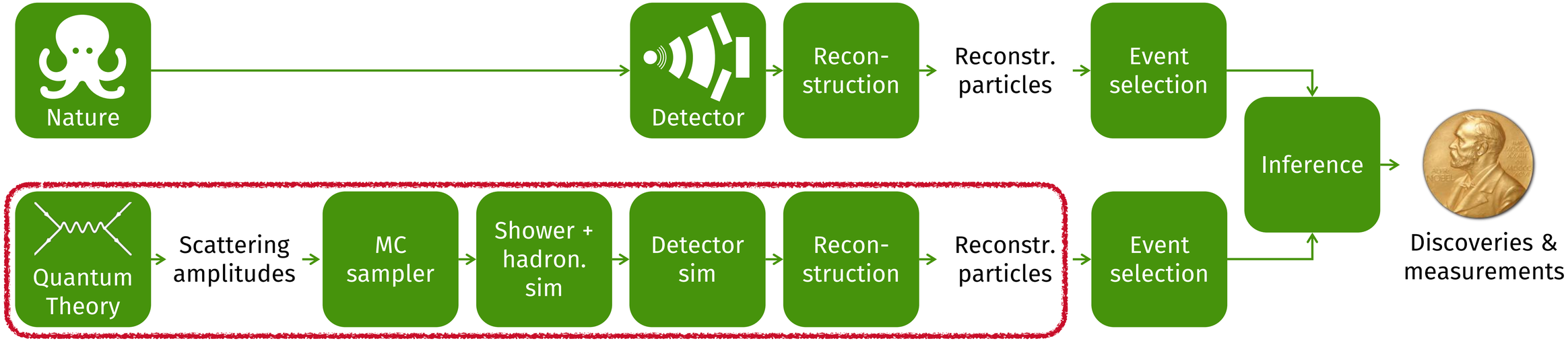


L-GATr is on par with the best equivariant (*) baselines

Experiments

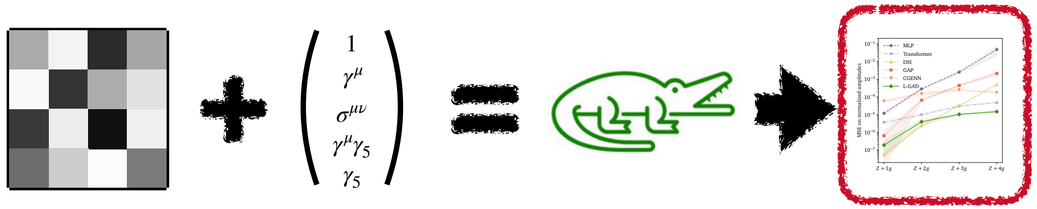
Event generation





Experiments

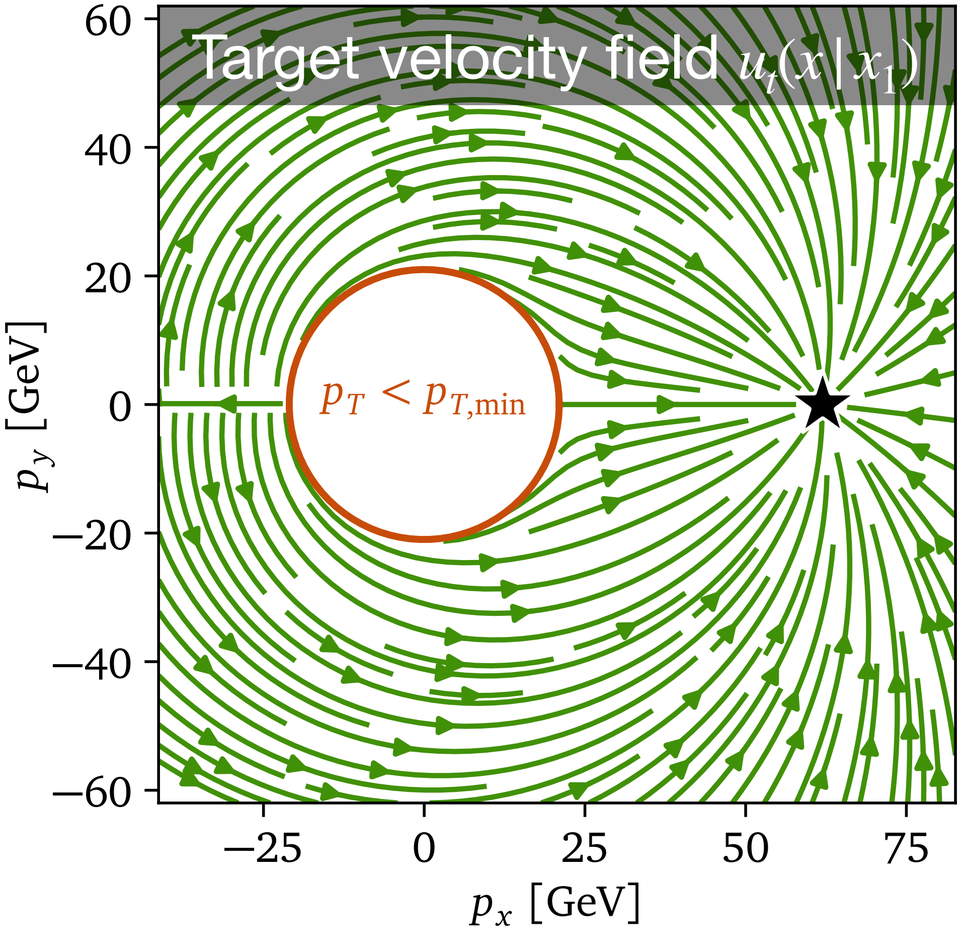
Event generation



Continuous normalising flows (CNF)
connect a simple base density
to a complex target density
through a neural differential equation

$$\frac{d}{dt}x = v_t(x)$$

Conditional flow matching (CFM)
is a simple way to train CNFs
by comparing the learned velocity $v_t(x)$
to a conditional target velocity $u_t(x \mid x_1)$

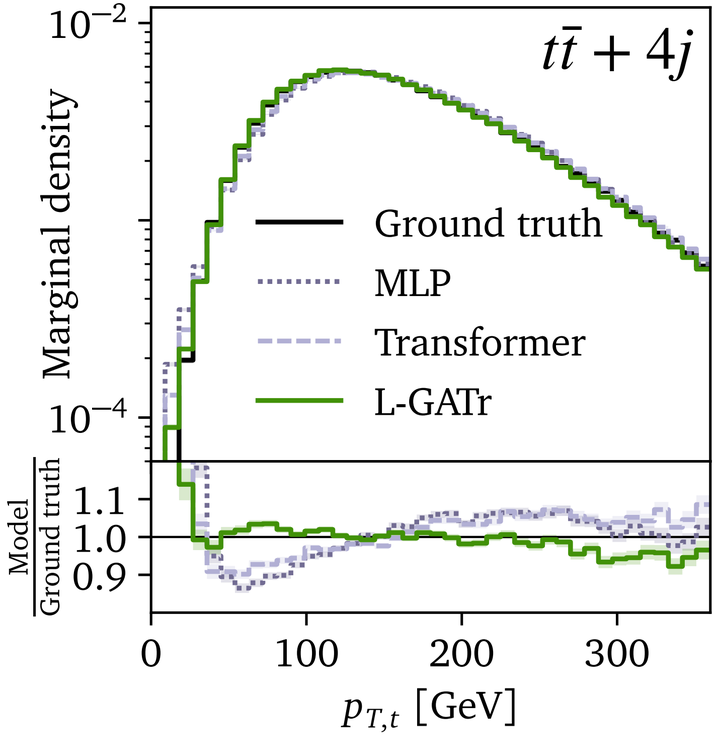


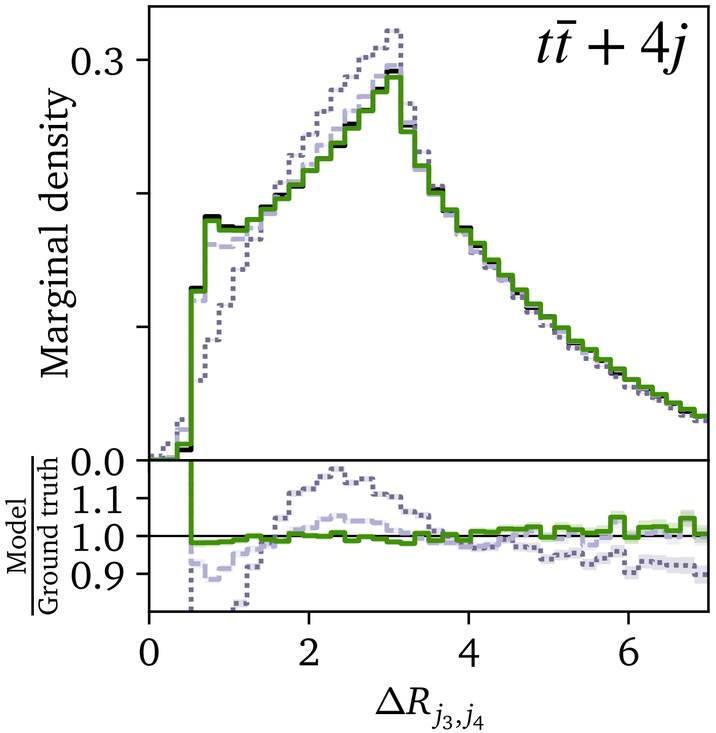
Continuous normalising flows
arXiv:1806.07366

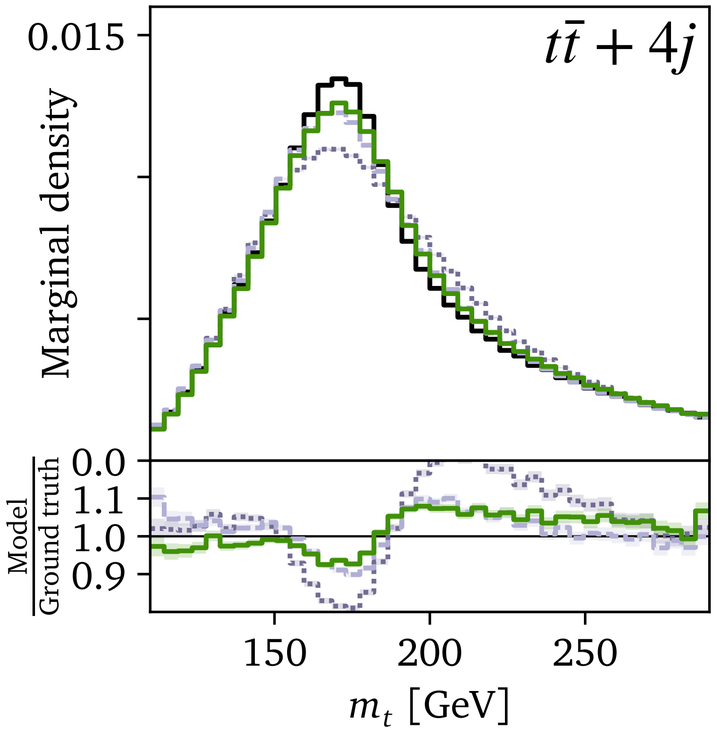
Conditional flow matching
arXiv:2210.02747

Experiments

Event generation



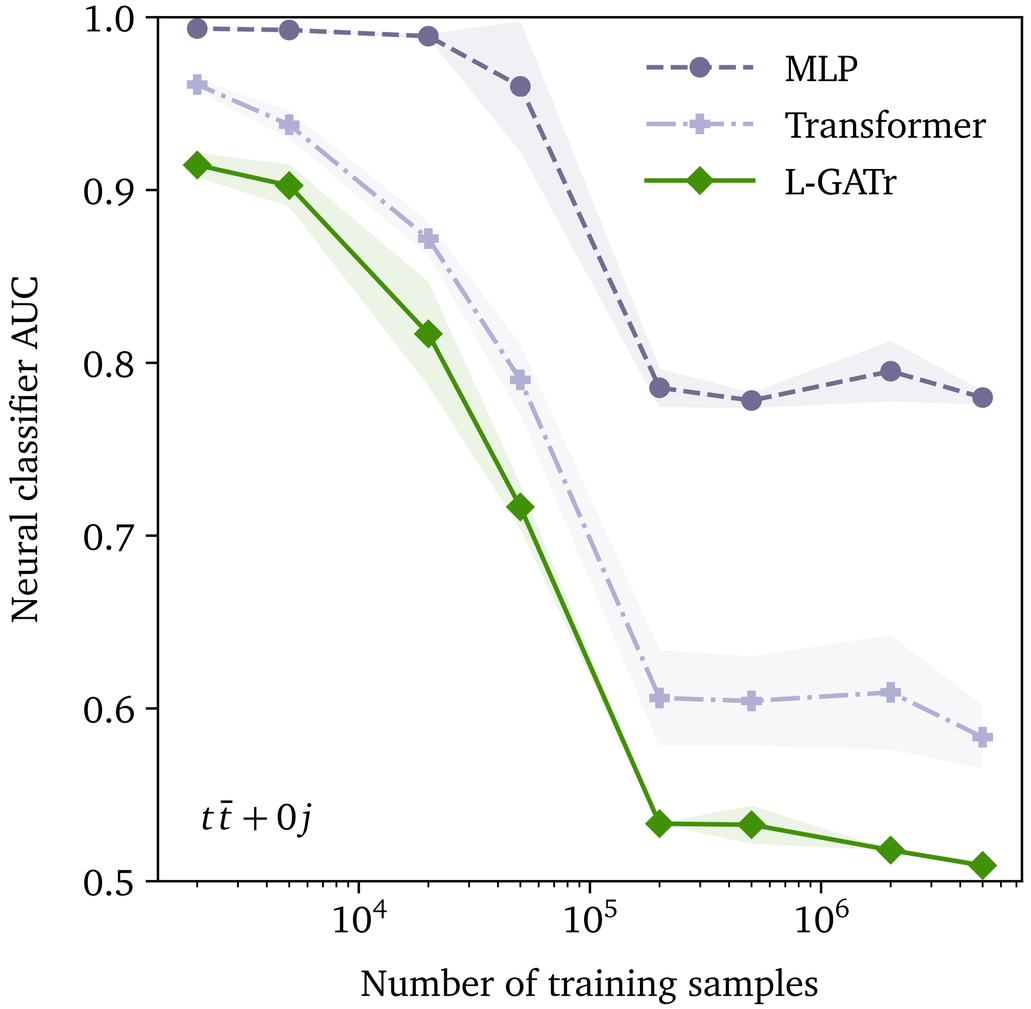


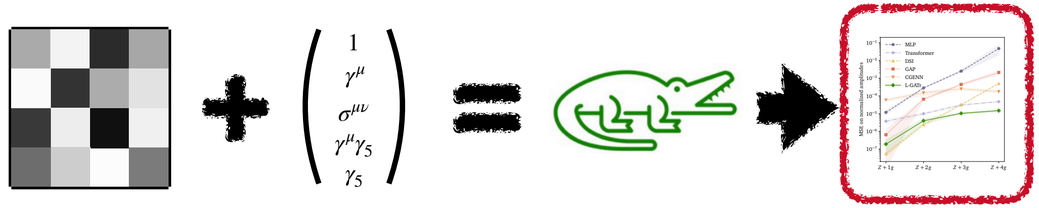


L-GATr helps with tricky kinematic features

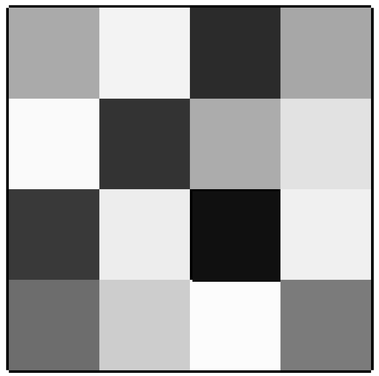
Experiments

Event generation

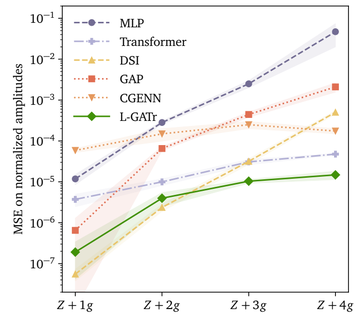




L-GATr generates samples that a classifier can almost not distinguish from the ground truth



$$\begin{pmatrix} 1 \\ \gamma^\mu \\ \sigma^{\mu\nu} \\ \gamma^\mu\gamma_5 \\ \gamma_5 \end{pmatrix}$$



Transformer
architecture

Geometric algebra
representations

Lorentz-Equivariant
Geometric **A**lgebra
Transformer

Strong performance
on diverse problems

L-GATr combines **equivariance** and **scalability**

Victor Bresó

Pim de Haan

Tilman Plehn

Jesse Thaler

Johann Brehmer

Geometric Algebra Transformer

E(3)-equivariant version
Johann Brehmer*, Pim de Haan*, Sönke Behrends, Taco Cohen
NeurIPS 2023, arXiv:2305.18415

E(3)-GATr paper

E(3)-GATr code

Lorentz-Equivariant Geometric Algebra Transformer for High-Energy Physics

Jonas Spinner*, Victor Breso*, Pim de Haan, Tilman Plehn, Jesse Thaler, Johann Brehmer
NeurIPS 2024, arXiv:2405.14806

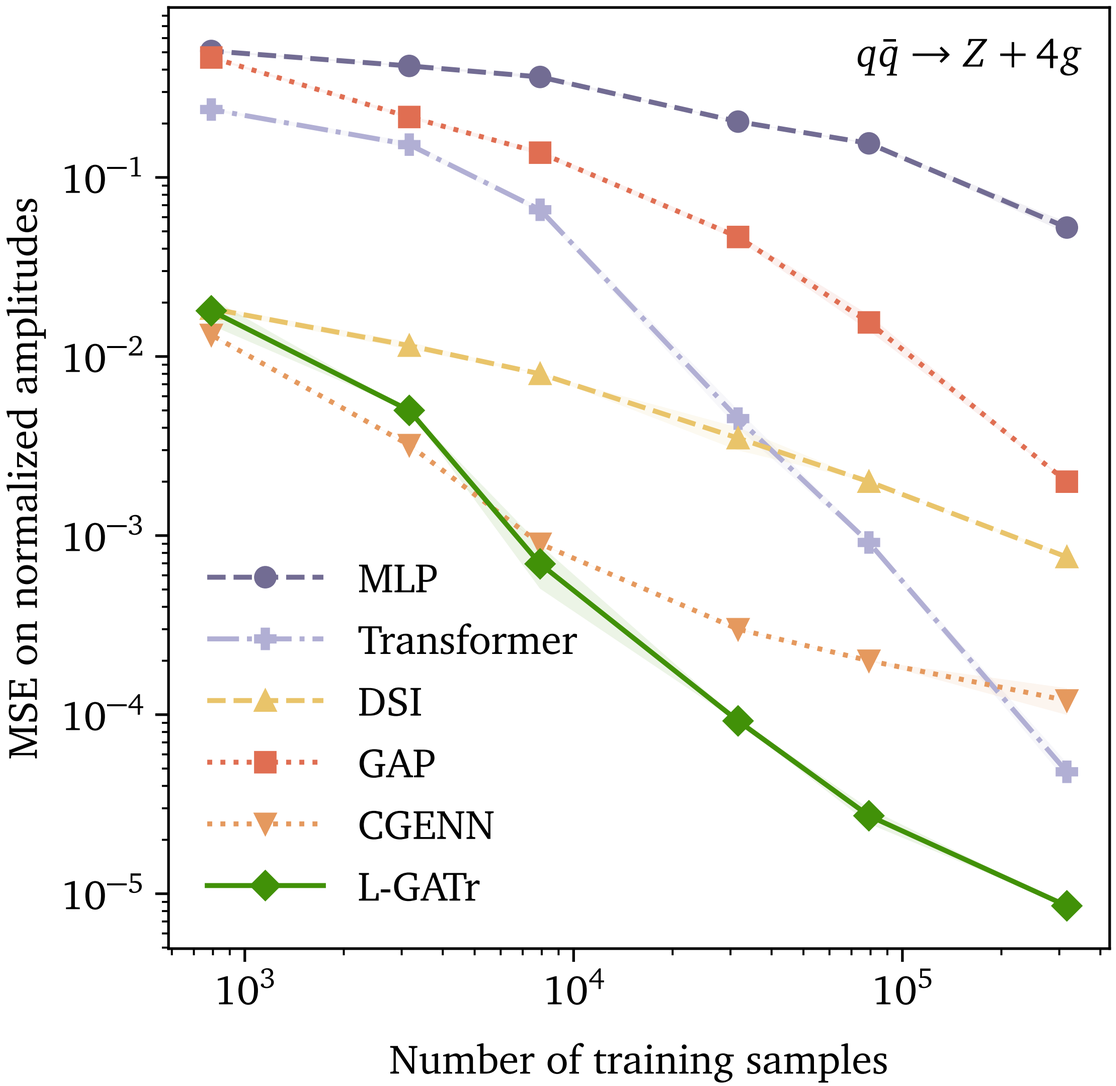
L-GATr paper

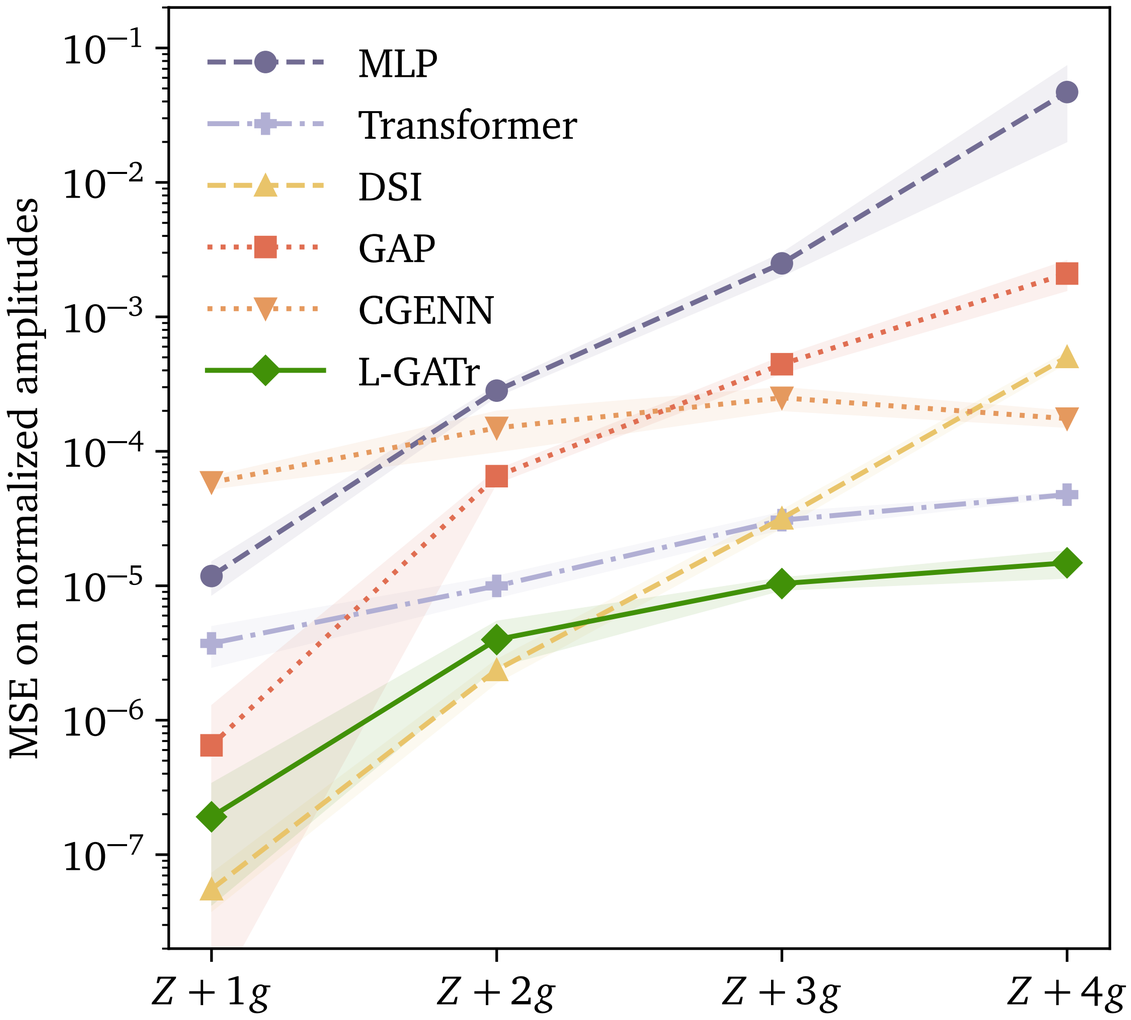
L-GATr code

What would **you** use L-GATr for?

Bonus material

Amplitude regression





Experiments

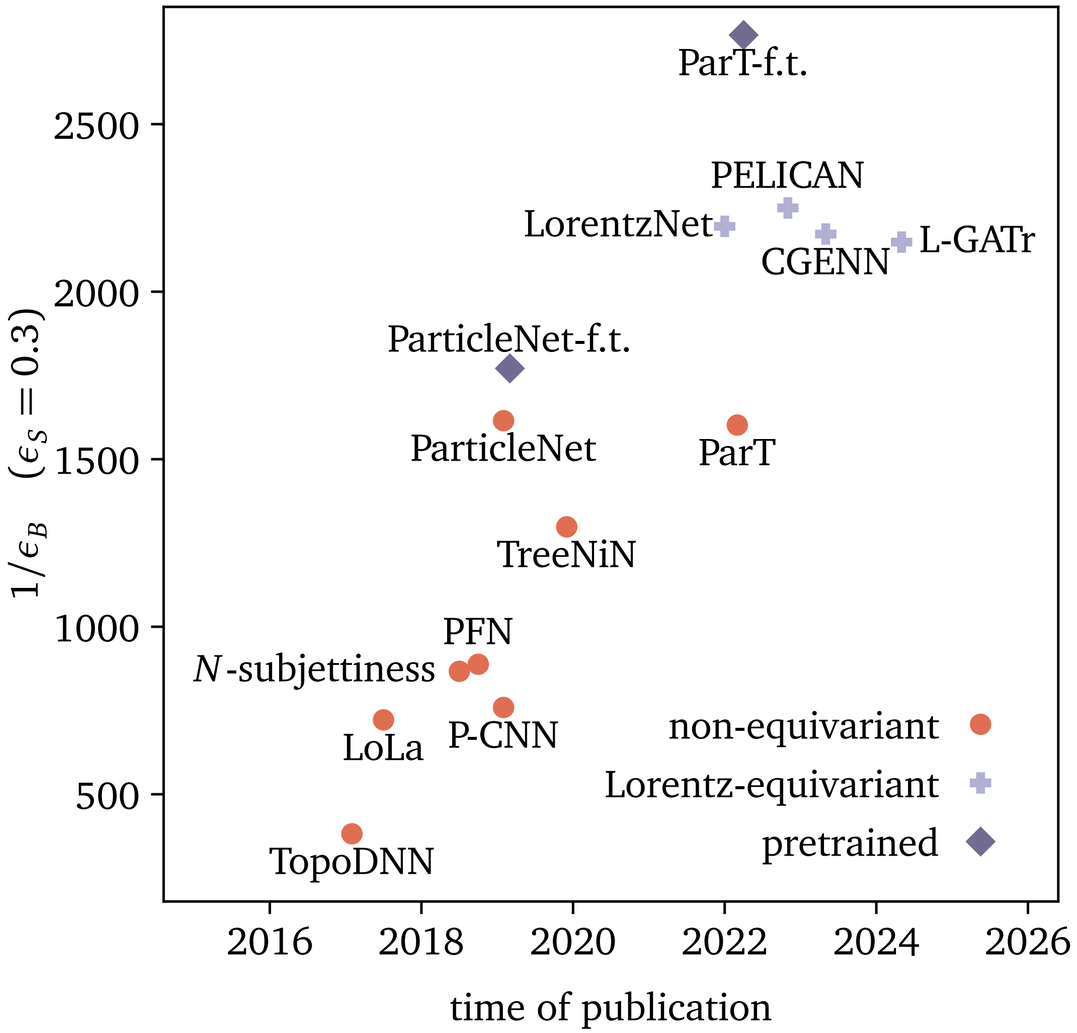
Top tagging

Model	Accuracy	AUC	$1/\epsilon_B$ ($\epsilon_S = 0.5$)	$1/\epsilon_B$ ($\epsilon_S = 0.3$)
TopoDNN [48]	0.916	0.972	–	295 ± 5
LoLa [15]	0.929	0.980	–	722 ± 17
P-CNN [1]	0.930	0.9803	201 ± 4	759 ± 24
N-subjettiness [61]	0.929	0.981	–	867 ± 15
PFN [50]	0.932	0.9819	247 ± 3	888 ± 17
TreeNiN [57]	0.933	0.982	–	1025 ± 11
ParticleNet [63]	0.940	0.9858	397 ± 7	1615 ± 93
ParT [64]	0.940	0.9858	413 ± 16	1602 ± 81
LorentzNet* [41]	0.942	0.9868	498 ± 18	2195 ± 173
CGENN* [67]	0.942	0.9869	500	2172
PELICAN* [9]	**0.9426** ± 0.0002	**0.9870** ± 0.0001	–	**2250** ± 75
L-GATr (ours)*	0.9417 ± 0.0002	0.9868 ± 0.0001	**548** ± 26	2148 ± 106

Experiments

Top tagging

- New paradigm: **Transfer learning**
  Pretrain model on large dataset, then fine-tune on target dataset
- Transformers transfer better than graph networks



Experiments

Conditional Flow Matching

Continuous normalising flows (CNF)
connect a simple base density
to a complex target density
through a neural differential equation

$$\frac{d}{dt}x = v_t(x)$$

Conditional flow matching (CFM)
is a simple way to train CNFs
by comparing the learned velocity $v_t(x)$
to a conditional target velocity $u_t(x \mid x_1)$

$$\mathscr{L} = \mathbb{E}_{t,x,x_1}\|v_t(x) - u_t(x \mid x_1)\|^2$$

Continuous normalising flows
arXiv:1806.07366

Conditional flow matching
arXiv:2210.02747

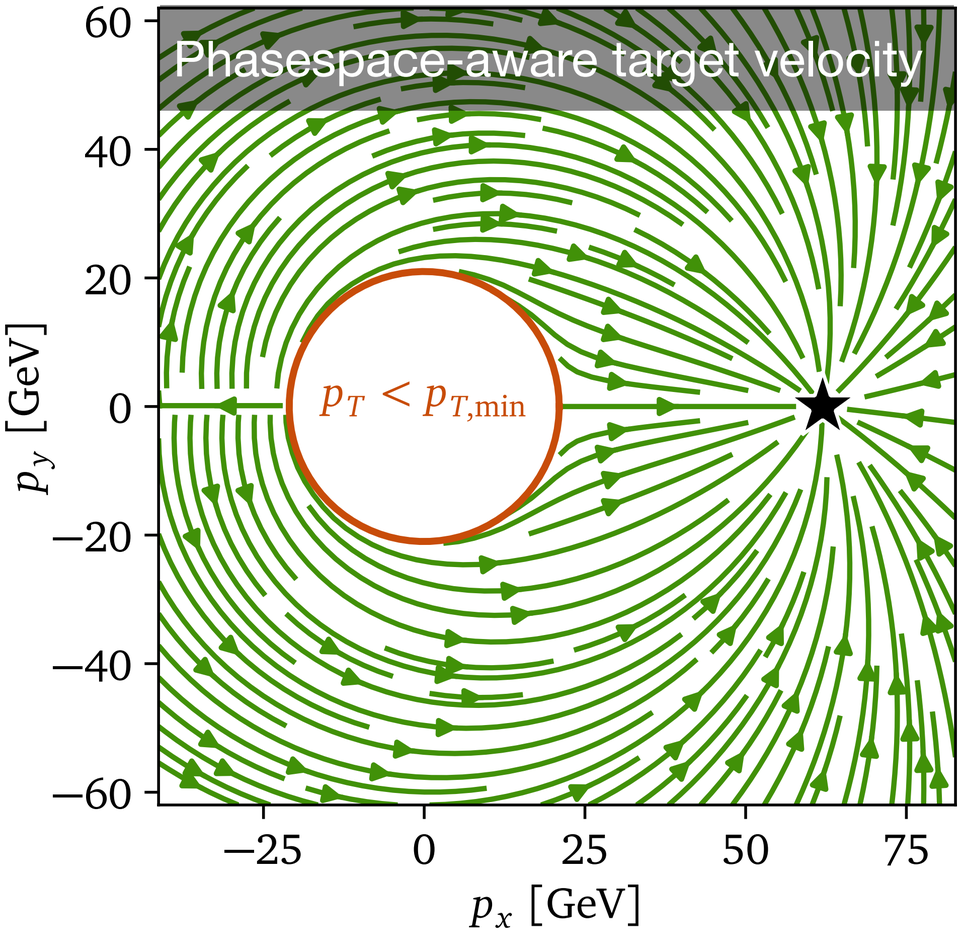
Experiments

Target velocities for CFM

In conditional flow matching (CFM), the choice of target velocity can be more important than the architecture

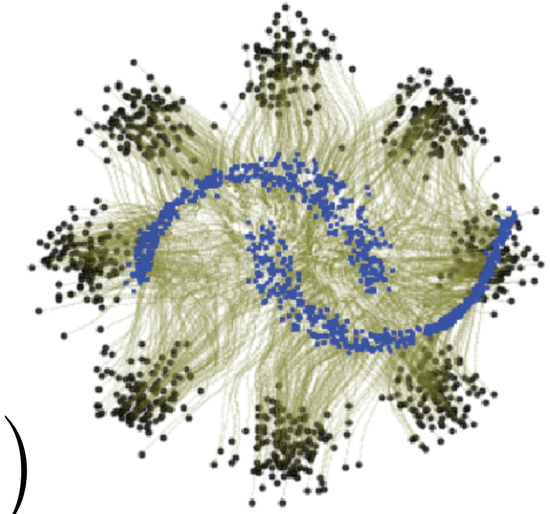
Target velocity	Architecture	AUC
Euclidean	L-GATr	0.99
Phasespace-aware	MLP	0.78
Phasespace-aware	L-GATr	**0.51**

Riemannian Flow Matching
arXiv:2302.03660



Event generation

Target velocities for CFM



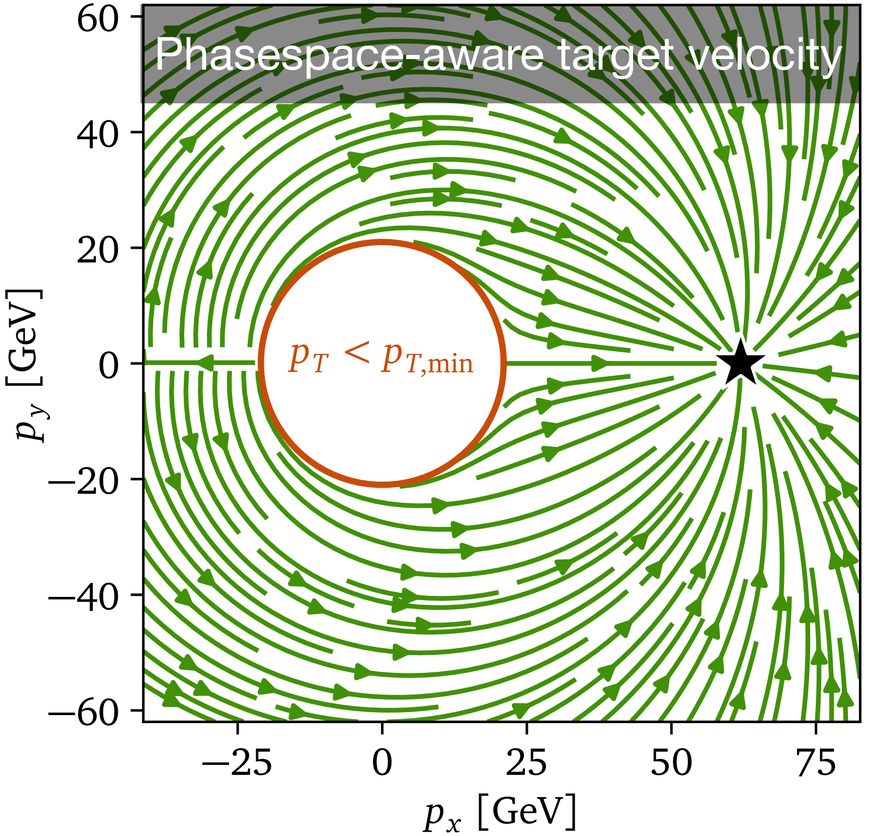
$$p = (E, p_x, p_y, p_z) = f(y) = \left(\sqrt{m^2 + p_T^2 \cosh^2 \eta}, p_T \cos\phi, p_T \sin\phi, p_T \sinh\eta\right)$$

$$y = (y_m, y_p, \phi, \eta), \qquad m^2 = \exp(y_m), \quad p_T = p_{T,\min} + \exp(y_p)$$

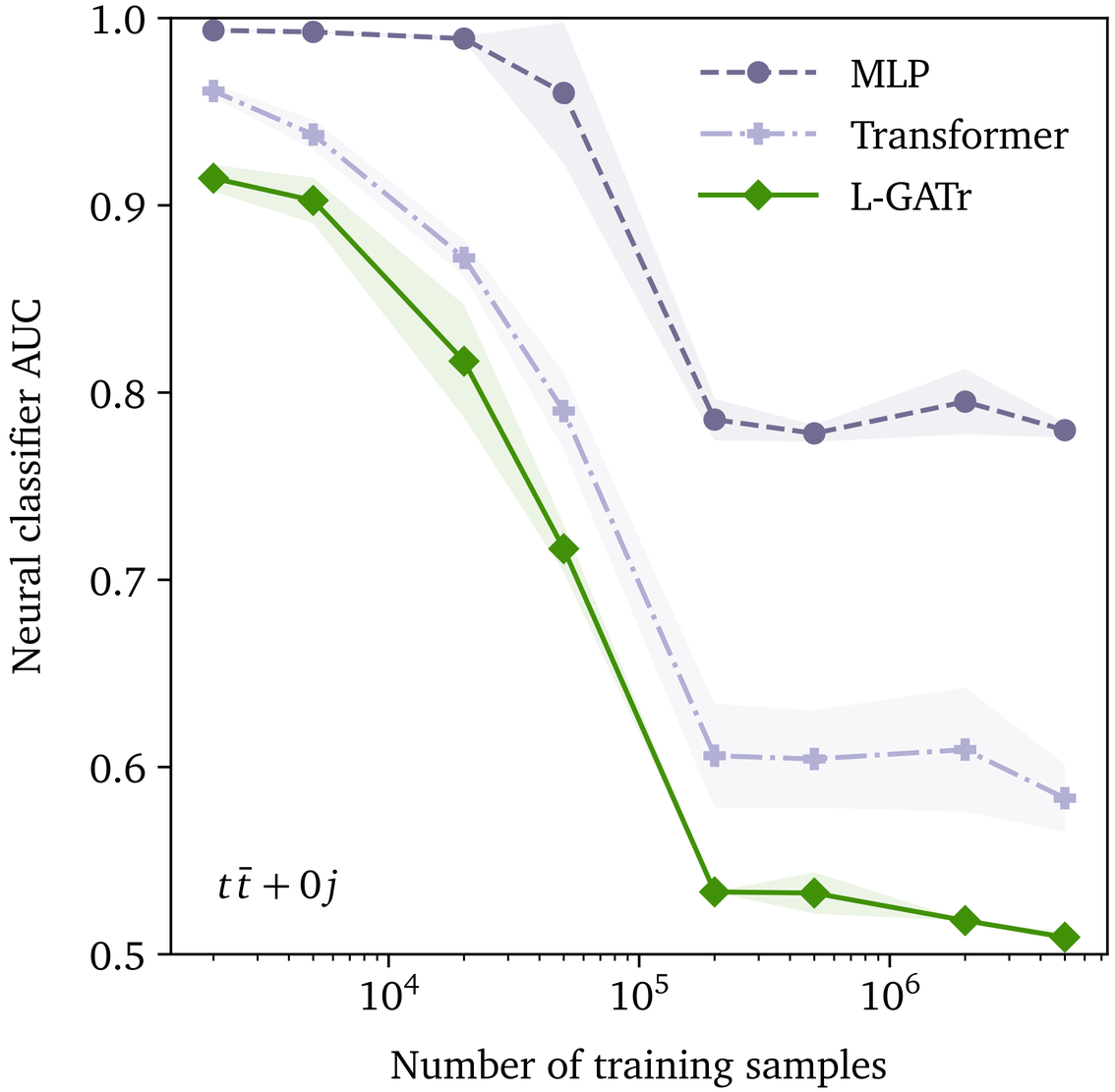
Target velocities can be

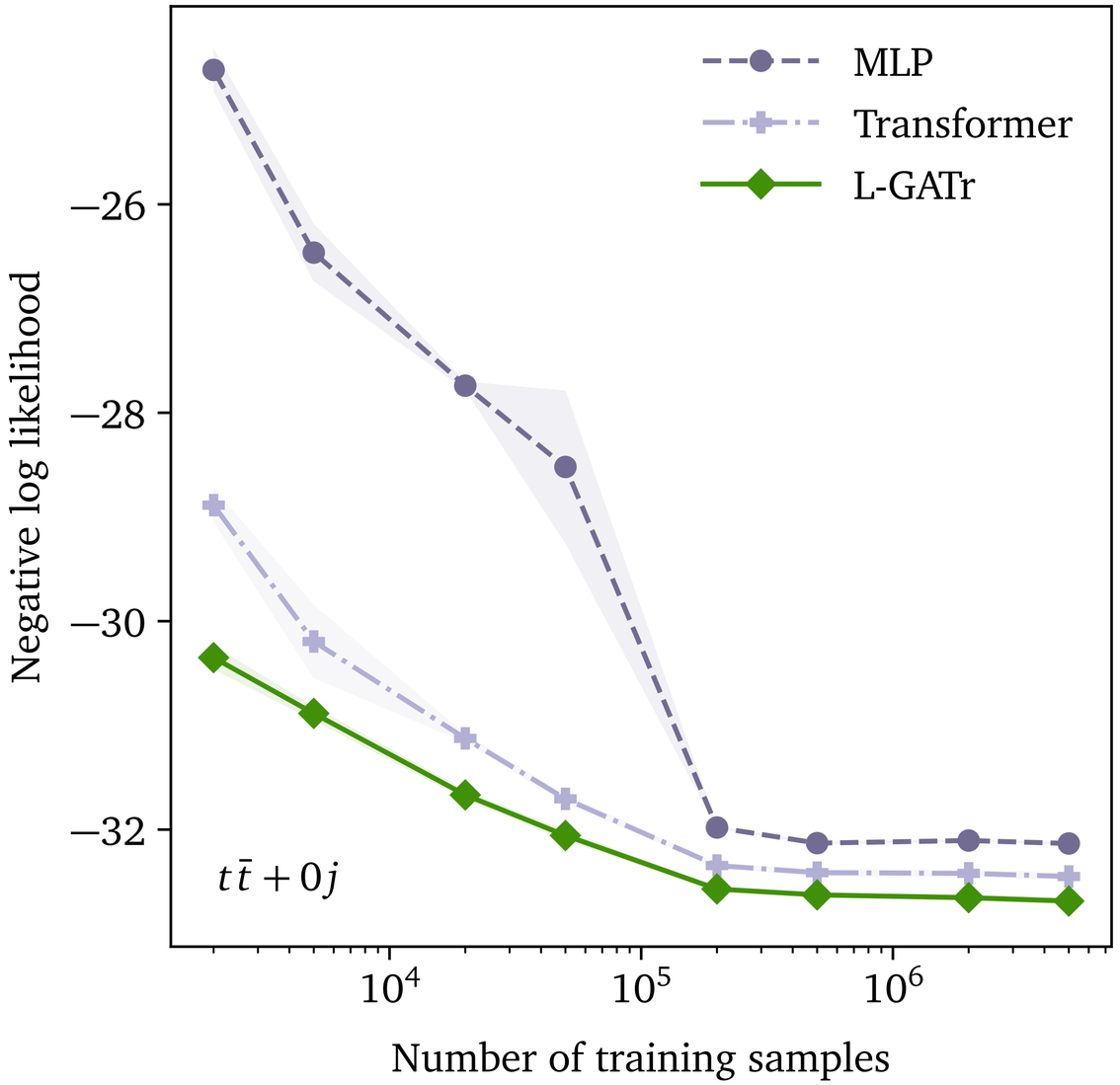
constant in $p = (E, p_x, p_y, p_z)$ ('euclidean')

constant in $y = (y_m, y_p, \phi, \eta)$ ('phasespace-aware')

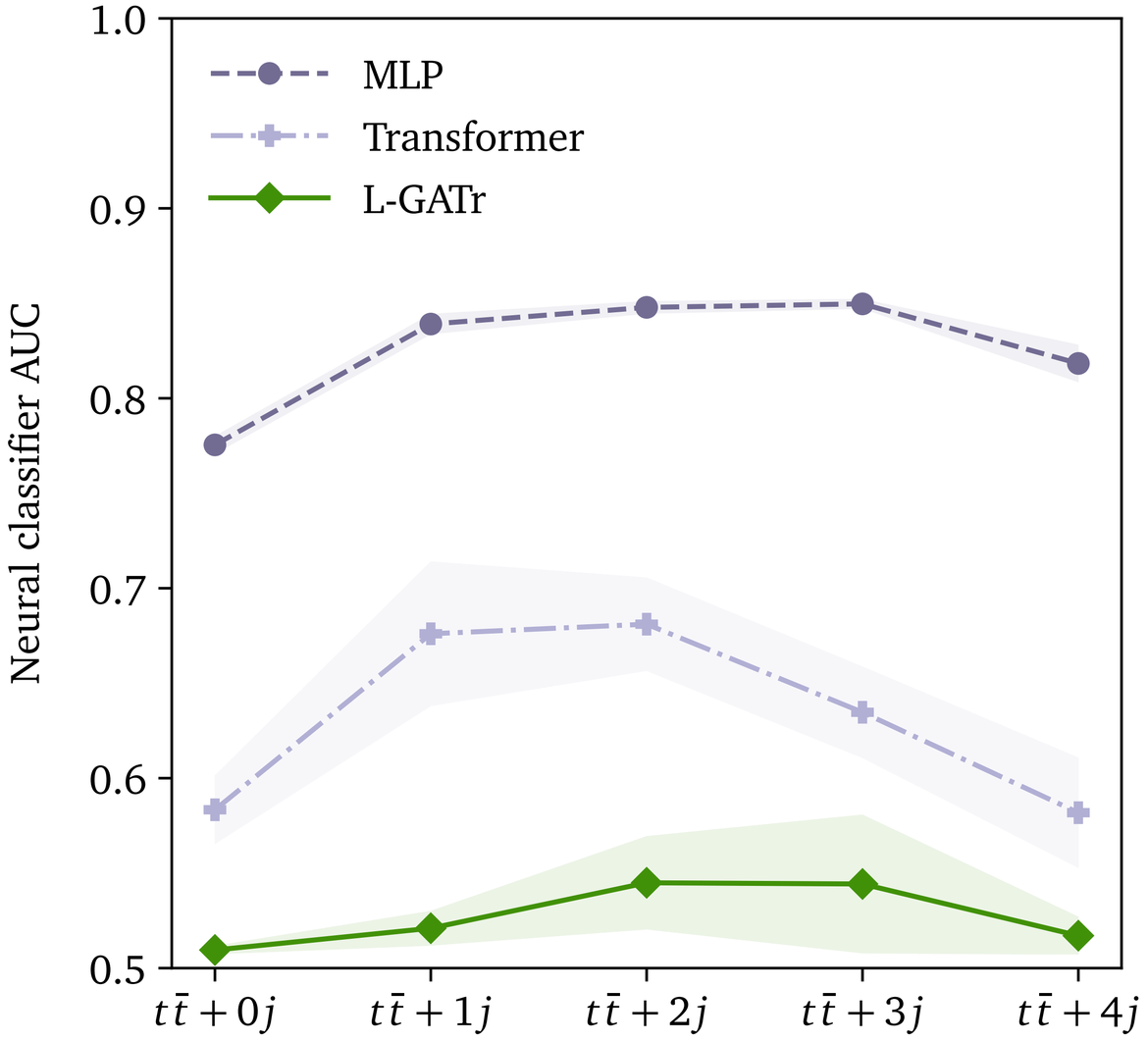


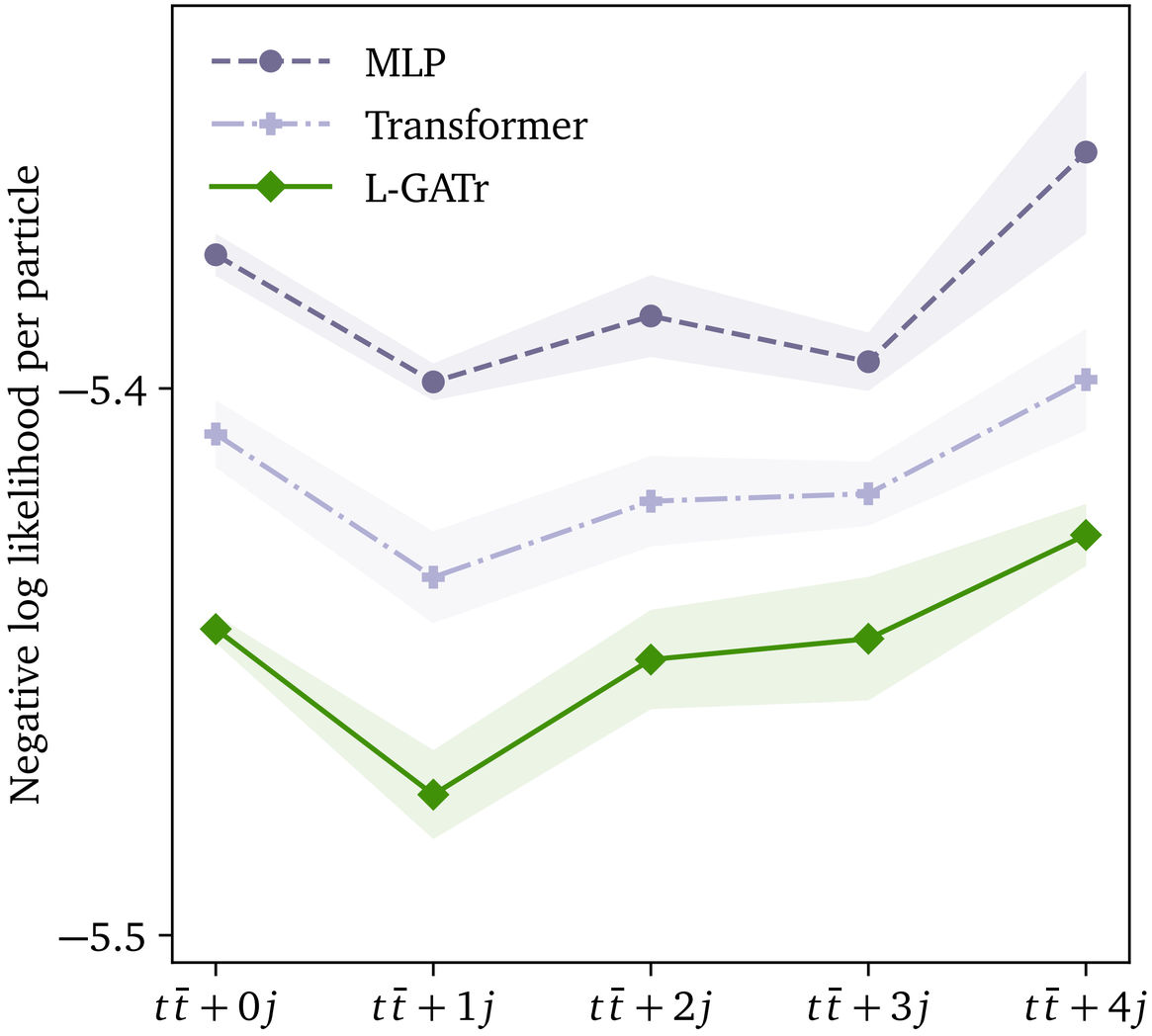
Event generation





Event generation





Symmetry breaking with spurious

Sources of symmetry breaking

- Real world: Beam direction, detector geometry…
  Symmetry-breaking object: Beam direction spurion

- Generation: Have to break $SO(1,3) \to SO(3)$ because generative networks can only be defined on compact groups
  Symmetry-breaking object: Time direction spurn

We break the symmetry by adding the spurious as extra token or as extra channel for each token