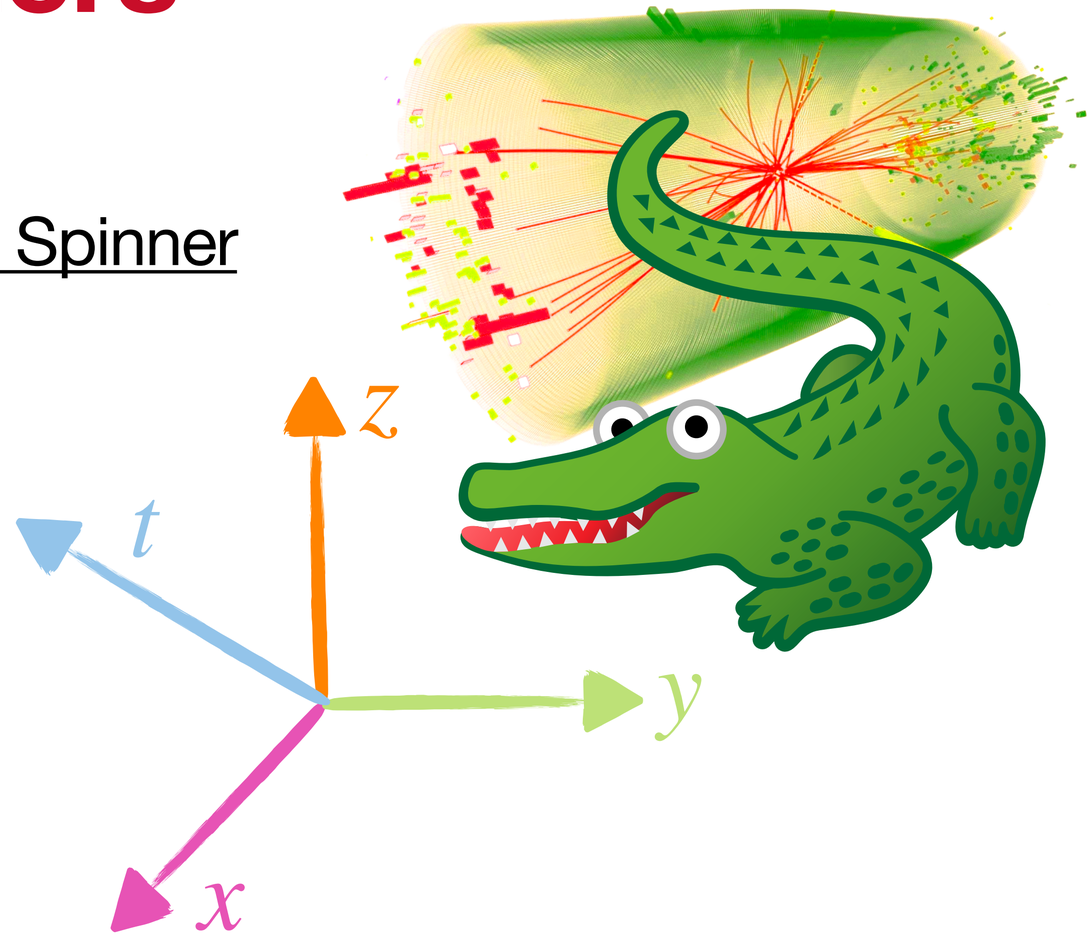
Virtues and Vices of Equivariant Transformers

Luigi Favaro, Tilman Plehn, Huilin Qu, Jonas Spinner

arXiv:2608.02735



ML4Jets 2026
18.09.2026

Classes of Lorentz-equivariant architectures

Invariants

$$m_{ij}^2 = \langle p_i, p_j \rangle$$

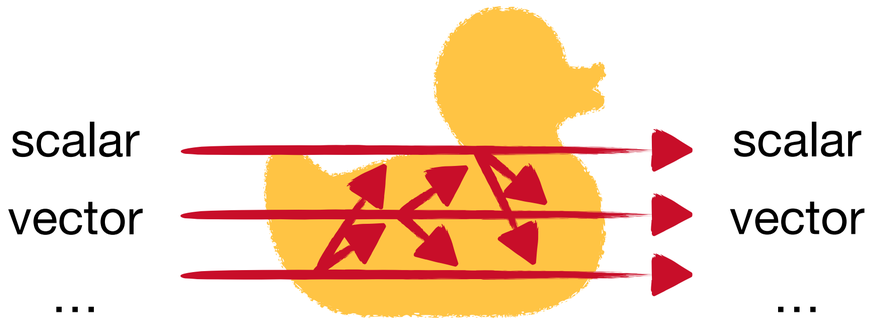
PELICAN

arXiv:2211.00454
arXiv:2307.16506

ParT

arXiv:2202.03772
(approximate)

Specialized layers

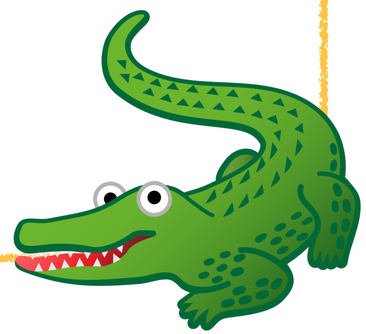


LorentzNet

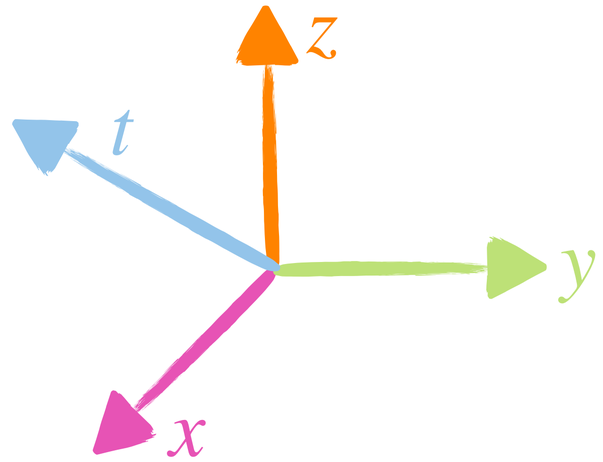
arXiv:2201.08187

L-GATr(-slim)

arXiv:2405.14806
arXiv:2411.00446
arXiv:2512.17011



Canonicalization



LLoCa-X

arXiv:2505.20280
arXiv:2508.14898

Graph Network
Transformer

L-GATr-slim

More in Antoine's talk (next)

arXiv:2405.14806
arXiv:2411.00446
arXiv:2512.17011

Extend standard network layers
to also operate on vectors

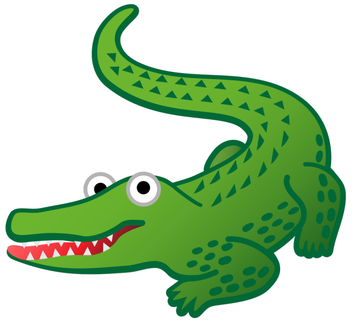
$$\mathrm{Linear}(x) = \mathrm{Linear}\begin{pmatrix} s \\ v \end{pmatrix} = \begin{pmatrix} \mathrm{Linear}_s(s) \\ \mathrm{Linear}_v(v) \end{pmatrix} = \begin{pmatrix} w_s \cdot s + b_s \\ w_v \cdot v \end{pmatrix}$$

$$\mathrm{GLU}\begin{pmatrix} s \\ v \end{pmatrix} = \begin{pmatrix} \mathrm{GELU}\left(\mathrm{Linear}_{s,1}(s)\right) \mathrm{Linear}_{s,2}(s) \\ \mathrm{GELU}\left(\langle \mathrm{Linear}_{v,1}(v), \mathrm{Linear}_{v,2}(v) \rangle\right) \mathrm{Linear}_{v,3}(v) \end{pmatrix}$$

$$\mathrm{RMSNorm}\begin{pmatrix} s \\ v \end{pmatrix} = \left(\frac{1}{c_v} \sum_{c=1}^{c_v} |\langle v_c, v_c \rangle|^2 + \frac{1}{c_v} \sum_{c=1}^{c_s} s_c^2 + \epsilon \right)^{-1/2} \begin{pmatrix} s \\ v \end{pmatrix}$$

$$\mathrm{Attention}\begin{pmatrix} q_s, k_s, v_s \\ q_v, k_v, v_v \end{pmatrix}_i = \sum_{j=1}^{n_t} \mathrm{Softmax}_j \left(\frac{\sum_{c=1}^{c_s} q_{s,ic} k_{s,jc} + \sum_{c=1}^{c_v} \langle q_{v,ic}, k_{v,jc} \rangle}{\sqrt{4n_v + n_s}} \right) \begin{pmatrix} v_s \\ v_v \end{pmatrix}_j$$

$$\begin{aligned}
\bar{x} &= \mathrm{RMSNorm}(x) \\
\mathrm{AttentionBlock}(x) &= \mathrm{Linear} \circ \mathrm{Attention}\left(\mathrm{Linear}(\bar{x}), \mathrm{Linear}(\bar{x}), \mathrm{Linear}(\bar{x})\right) + x \\
\mathrm{MLPBlock}(x) &= \mathrm{Linear} \circ \mathrm{GLU}(\bar{x}) + x \\
\mathrm{Block}(x) &= \mathrm{MLPBlock} \circ \mathrm{AttentionBlock}(x) \\
\text{L-GATr-slim}(x) &= \mathrm{Linear} \circ \mathrm{Block} \circ \mathrm{Block} \circ \cdots \circ \mathrm{Block} \circ \mathrm{Linear}(x)
\end{aligned}$$

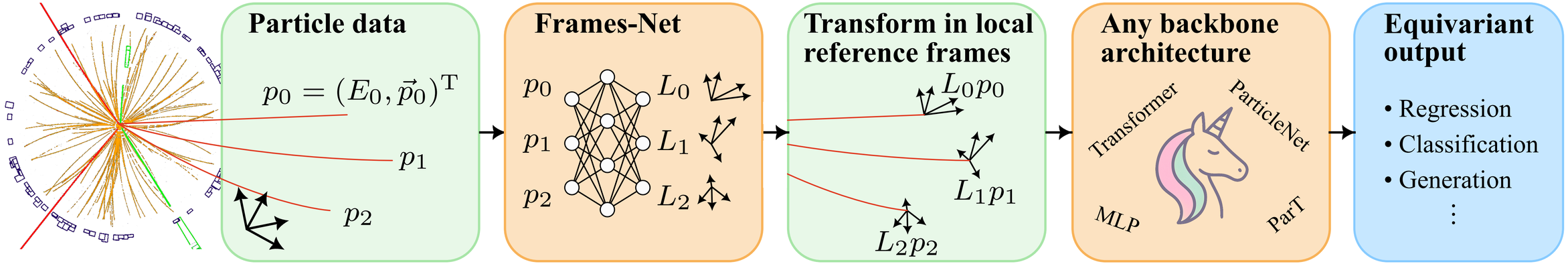


```
pip install lgatr
```

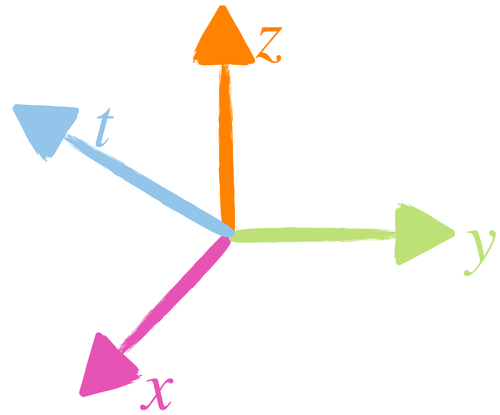
arXiv:2505.20280
arXiv:2508.14898

LLoCa-X

How to make any network Lorentz-equivariant:



Lorentz Local Canonicalization

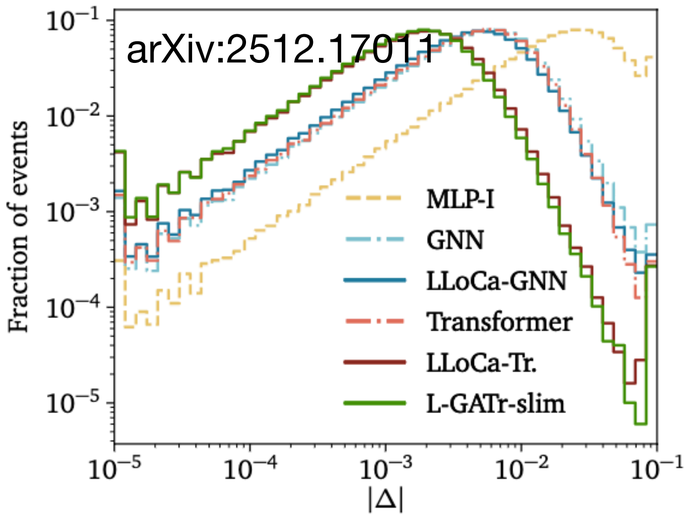


LoCa was originally developed for SO(3)
arXiv:2405.15389

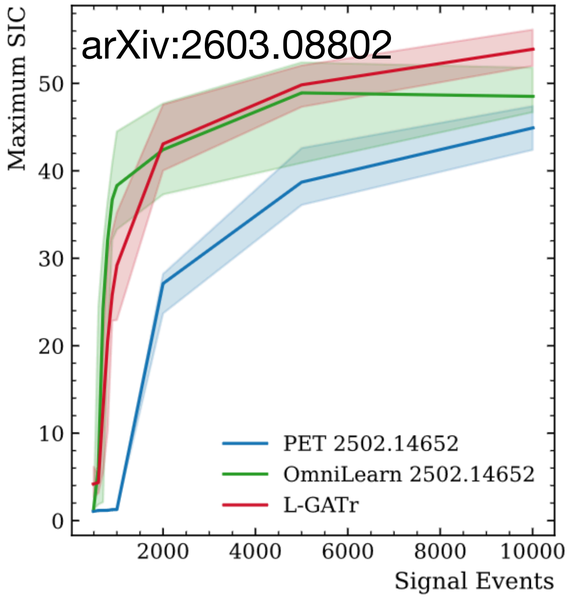
```
pip install lloca
```

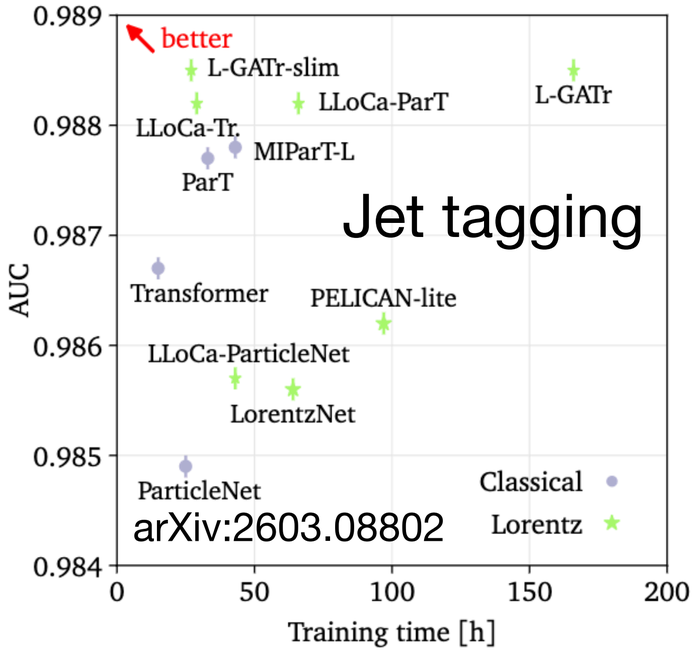
Lorentz equivariance is universal

Amplitude regression

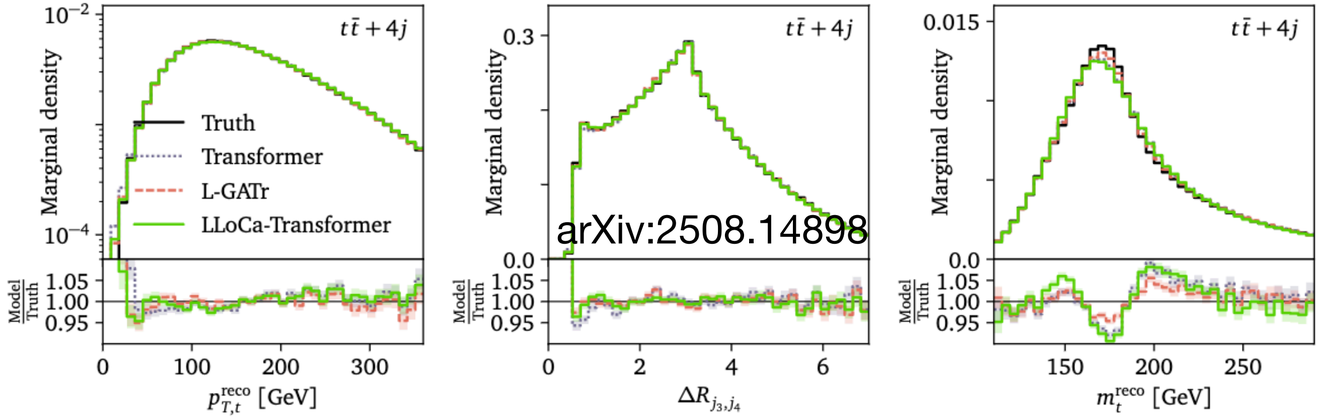


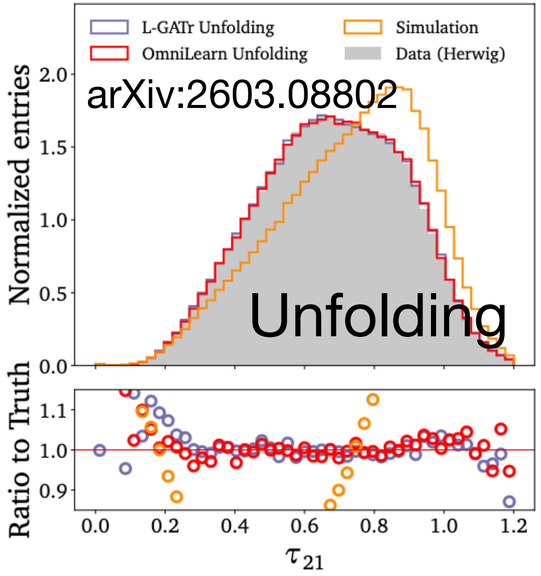
Anomaly detection





Event generation





Is Lorentz equivariance worth it?

Higher precision at
fixed dataset/network size…

Extra compute cost
(FLOPs, time, memory…)

More robust because
network knows symmetries

Extra design choices
(symmetry breaking…)

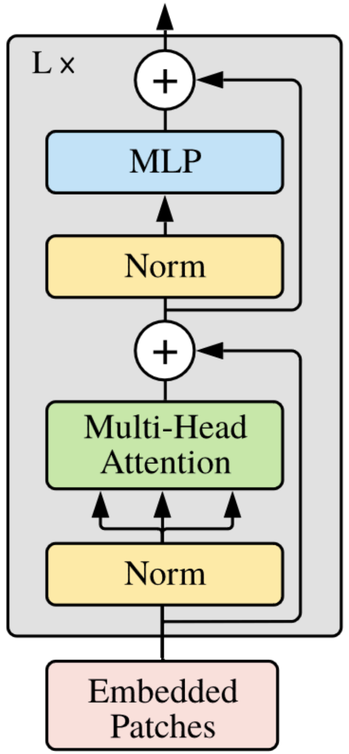
We are physicists!

Someone has to implement it

Who wins at fixed compute cost?

Transformer architectures

Transformer



ParT

$$m_{ij}^2 = \langle p_i, p_j \rangle$$

+ invariant edge features

arXiv:2202.03772

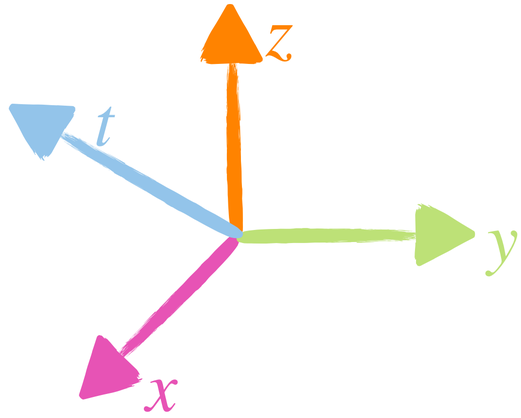
L-GATr(-slim)

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

+ (multi-)vector representations

arXiv:2405.14806
arXiv:2411.00446
arXiv:2512.17011

LLoCa



+ local frames

arXiv:2505.20280
arXiv:2508.14898

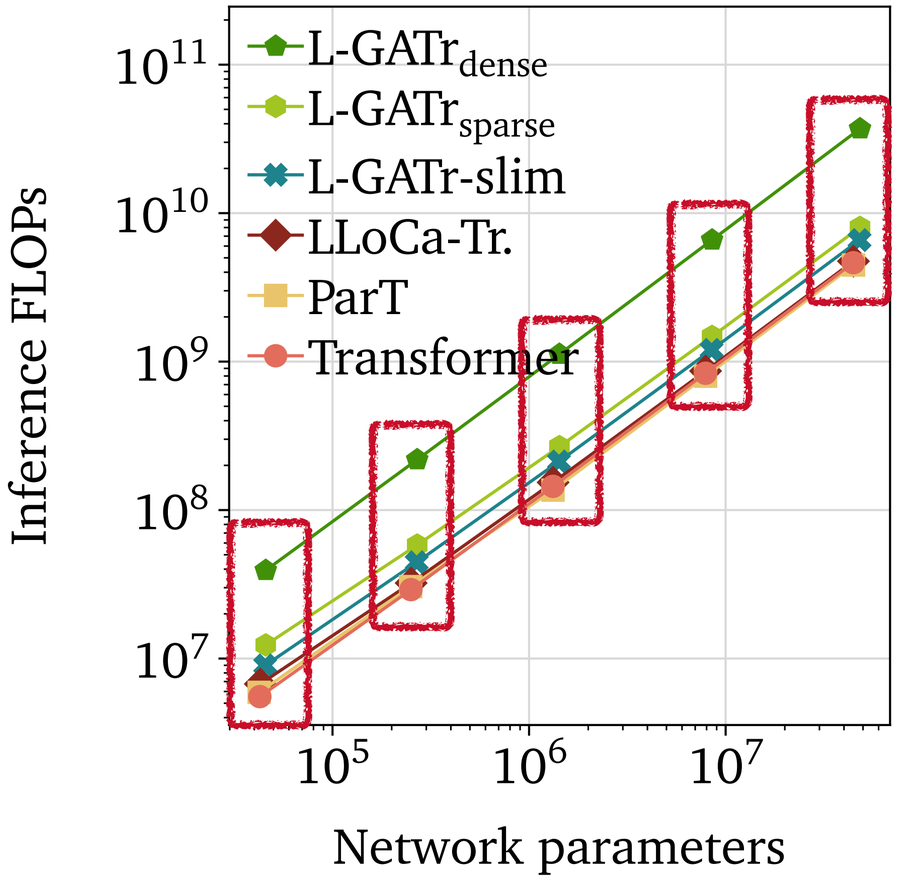
Same design principles for all architectures

`pip install weaver-core`

`pip install lgatr`

`pip install lloca`

Unified scaling recipe

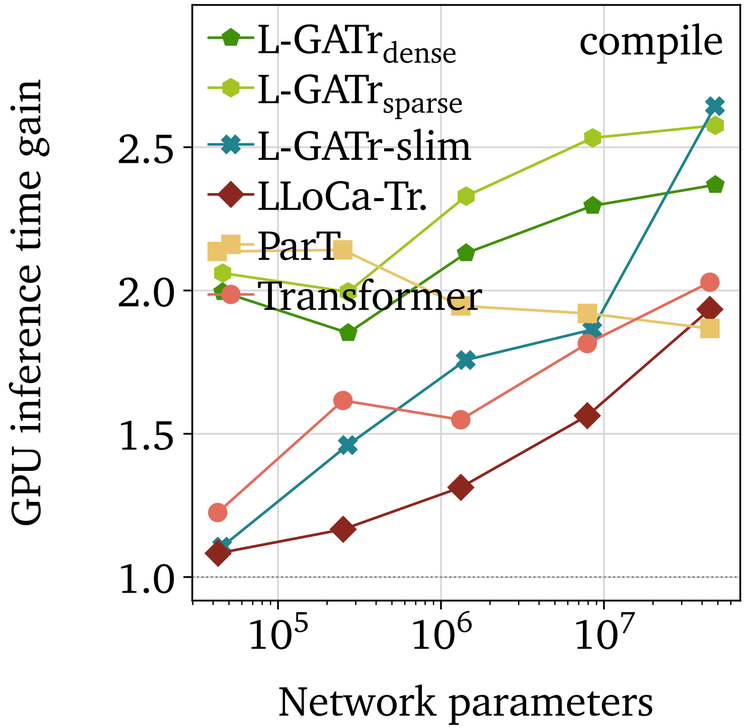


	XS $s=-2$	S $s=-1$	M $s=0$	L $s=1$	XL $s=2$	s
Channels	32	64	128	256	512	2^{7+s}
Blocks	4	6	8	12	17	$3 \cdot 2^{\lfloor (s+3)/2 \rfloor}$
Heads	4	4	8	8	16	$2^{3+\lfloor s/2 \rfloor}$
L-GATr-slim vectors	8	16	32	64	128	2^{5+s}
L-GATr multivectors	4	8	15	32	64	2^{4+s}
L-GATr scalars	24	48	96	192	384	$3 \cdot 2^{5+s}$
LLoCa Frames-Net channels	8	16	32	64	128	2^{5+s}
ParT edge features	16	32	64	128	256	2^{6+s}
ParT class-att'n blocks	1	1	2	2	2	2
Learning rate	10^{-3}	$5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	$1.25 \cdot 10^{-4}$	$6.25 \cdot 10^{-5}$	$2.5 \cdot 10^{-4} \cdot 2^{-s}$
Number of parameters (±5%)	$4.2 \cdot 10^{4}$	$2.5 \cdot 10^{5}$	$1.4 \cdot 10^{6}$	$7.9 \cdot 10^{6}$	$4.2 \cdot 10^{7}$	$1.4 \cdot 10^{6} \cdot 2^{3/2 \cdot s}$
FLOPs (±5%)	$6 \cdot 10^{6}$	$3 \cdot 10^{7}$	$2 \cdot 10^{8}$	$8 \cdot 10^{8}$	$4 \cdot 10^{9}$	$1.4 \cdot 10^{8} \cdot 2^{3/2 \cdot s}$

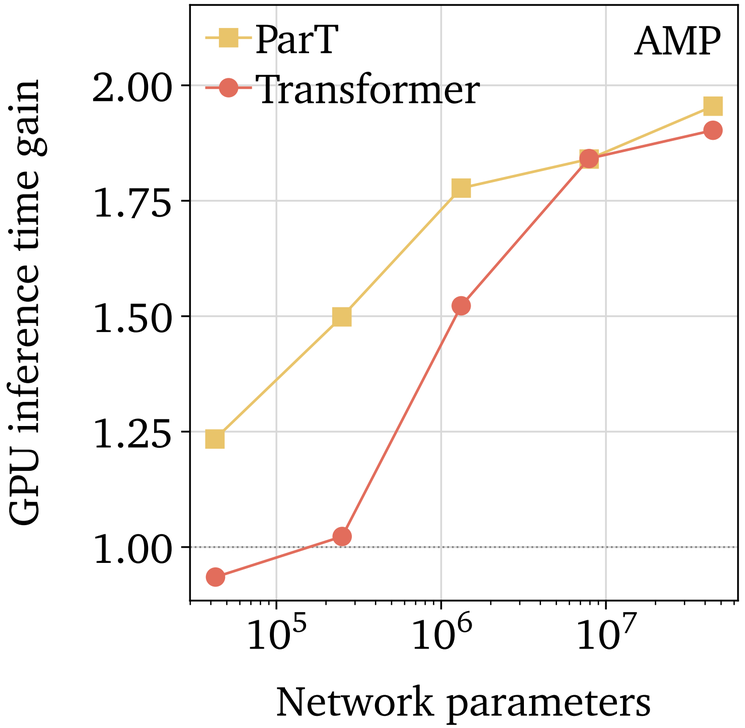
Only <10% extra parameters from equivariance

Efficient implementation

✓ `torch.compile`

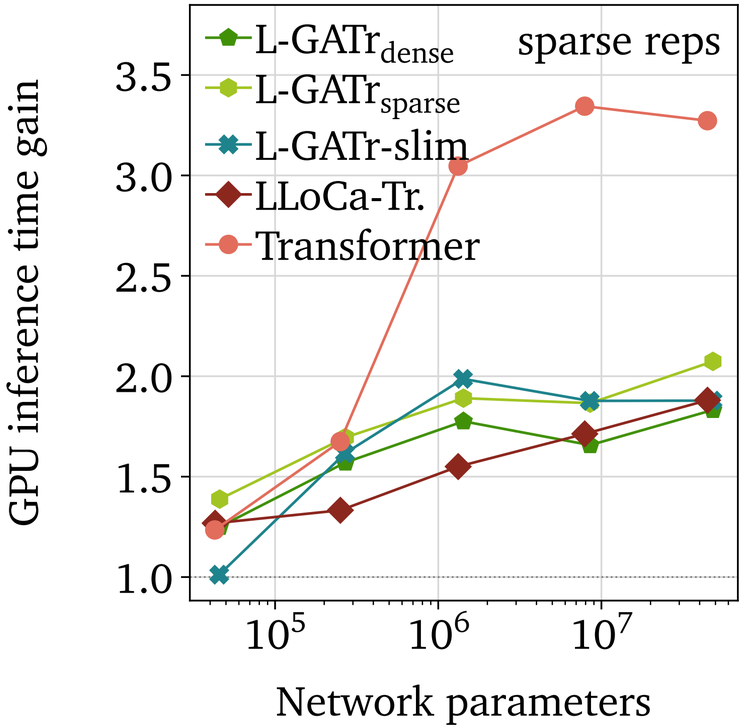


✓ mixed precision



*soon also for the equivariant family

✓ sparse jet reps



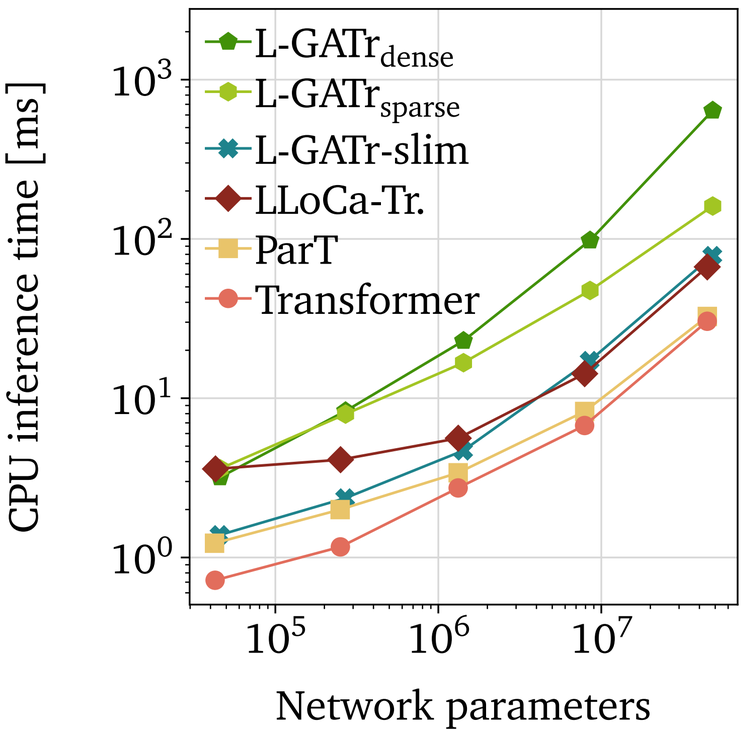
Efficient implementation

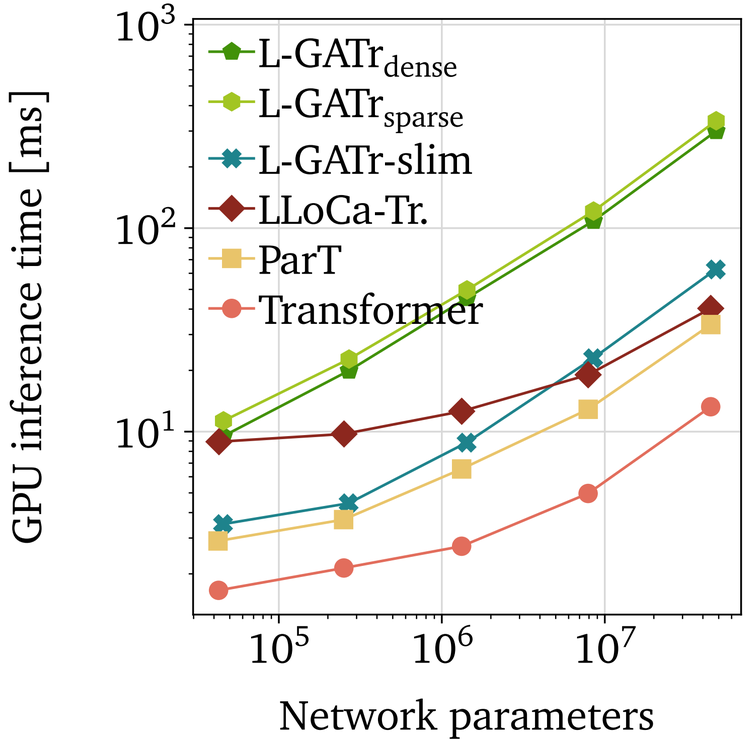
JetClass, 5 epochs with batch size 512, H100:

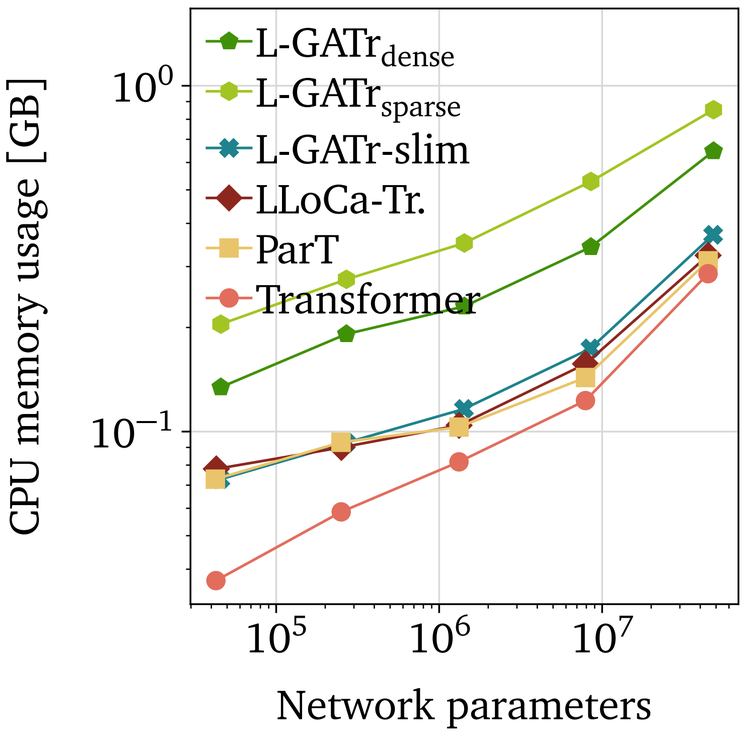
Architecture	Accuracy	AUC	Time	FLOPs	Memory
Baseline transformer [21]	0.855	0.9867	15h → 9h	210M	2.3G
ParT [11]	0.861	0.9878	33h → 19h	211M	13.3G → 7.2G
L-GATr$_{\text{sparse}}$ [19]	0.865	0.9885	166h → 63h	2060M → 352M	19.0G → 16.8G
L-GATr$_{\text{dense}}$ [19]	0.865	0.9885	166h → 57h	2060M → 1999M	19.0G → 14.3G
L-GATr-slim [20]	0.866	0.9885	27h → 16h	329M	8.1G
LLoCa transformer [21]	0.864	0.9882	28h → 12h	219M	4.1G

$2\times$ to $3\times$ speedup
compared to 2025 implementation

Efficient implementation



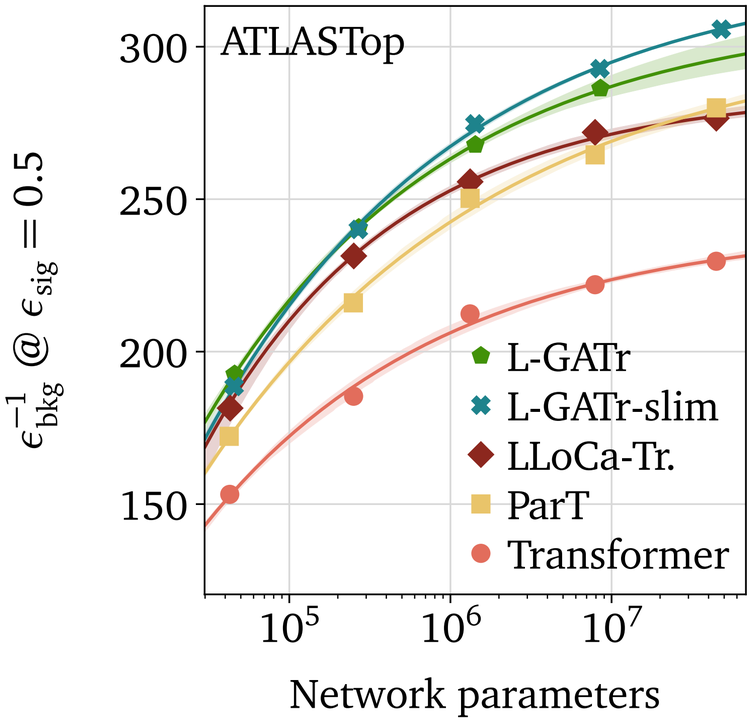




Lorentz equivariance and ParT similarly expensive

Transformer $2\times$ cheaper

Large-R jet tagging performance

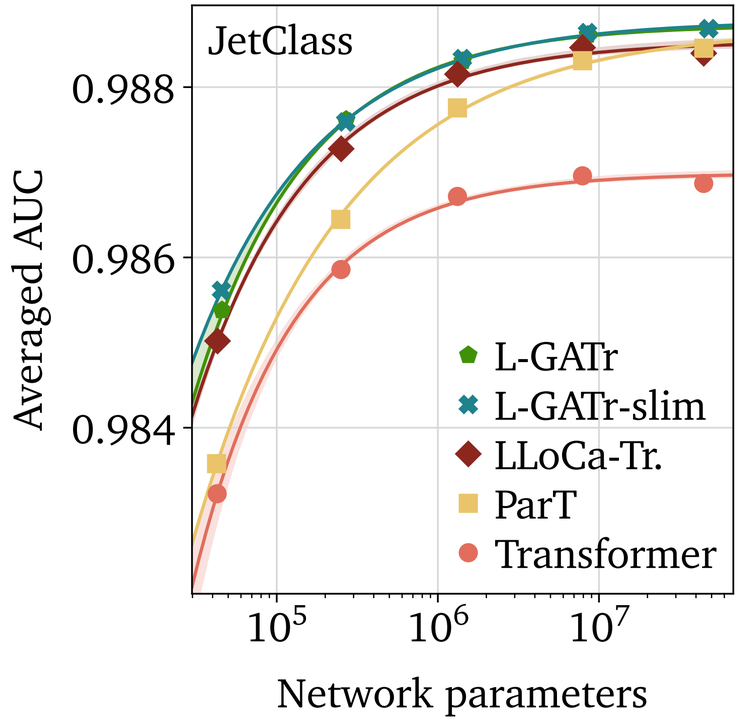


100M jets, full ATLAS simulation

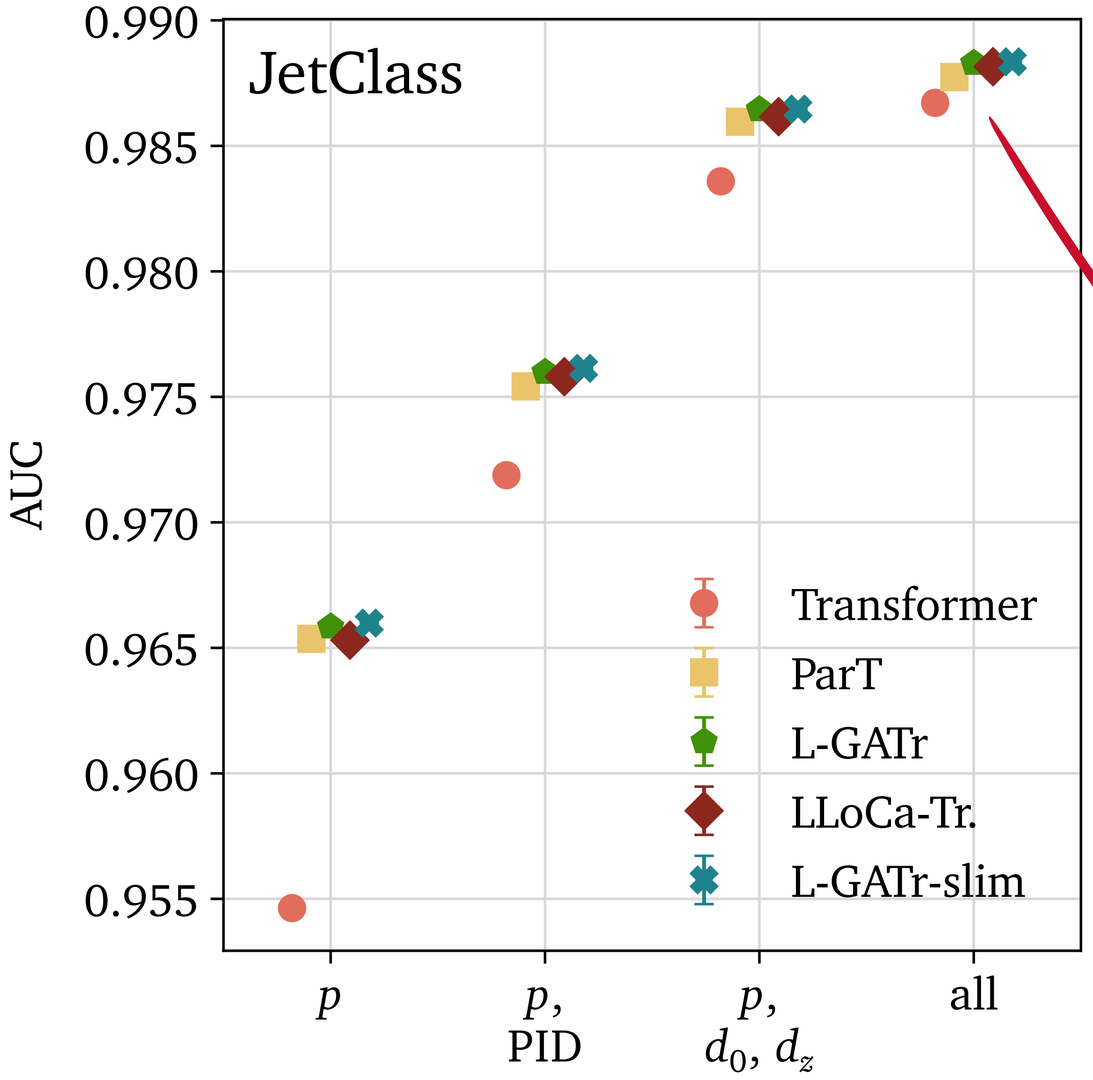
Only kinematic information

100M jets, Delphes

Kinematic information + PIDs
+ trajectory displacement information

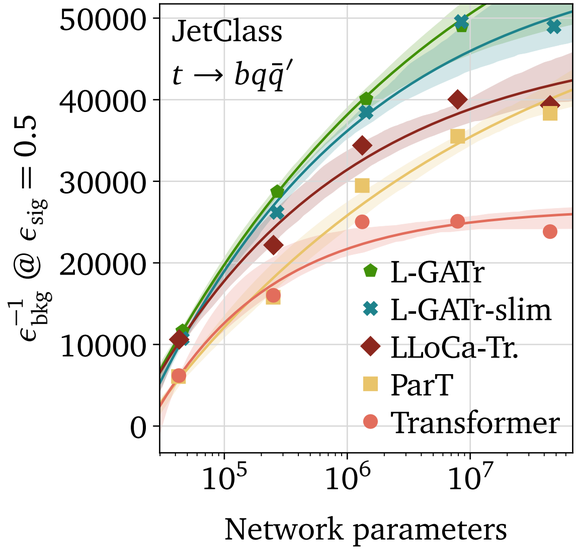


Feature relevance on JetClass

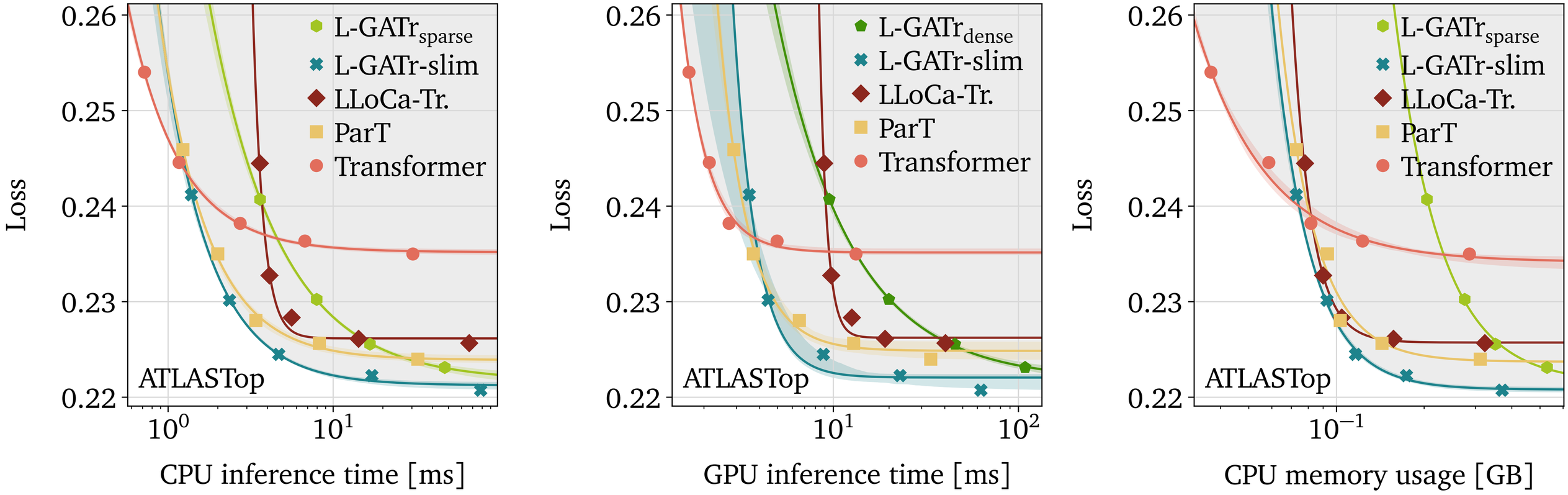


Largest equivariance gain
when training only on kinematics

With extra scalar features the gain
shrinks but remains significant

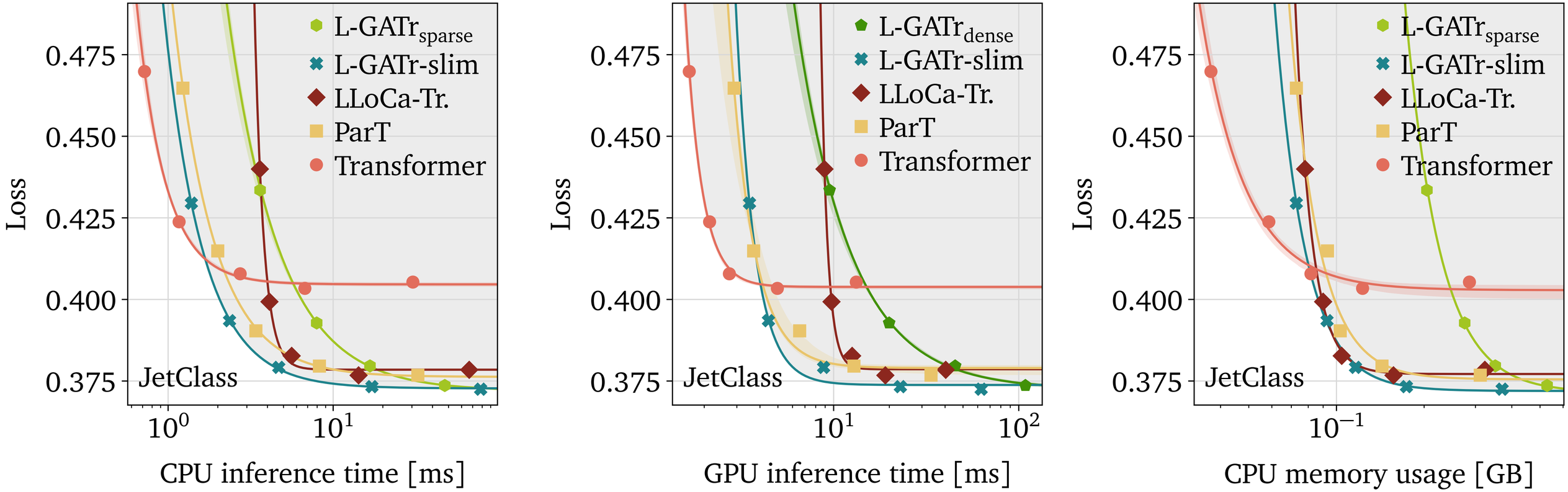


Fixed evaluation cost on ATLASTop



Lorentz equivariance wins at fixed cost!

Fixed evaluation cost on JetClass



Lorentz equivariance wins at fixed cost!

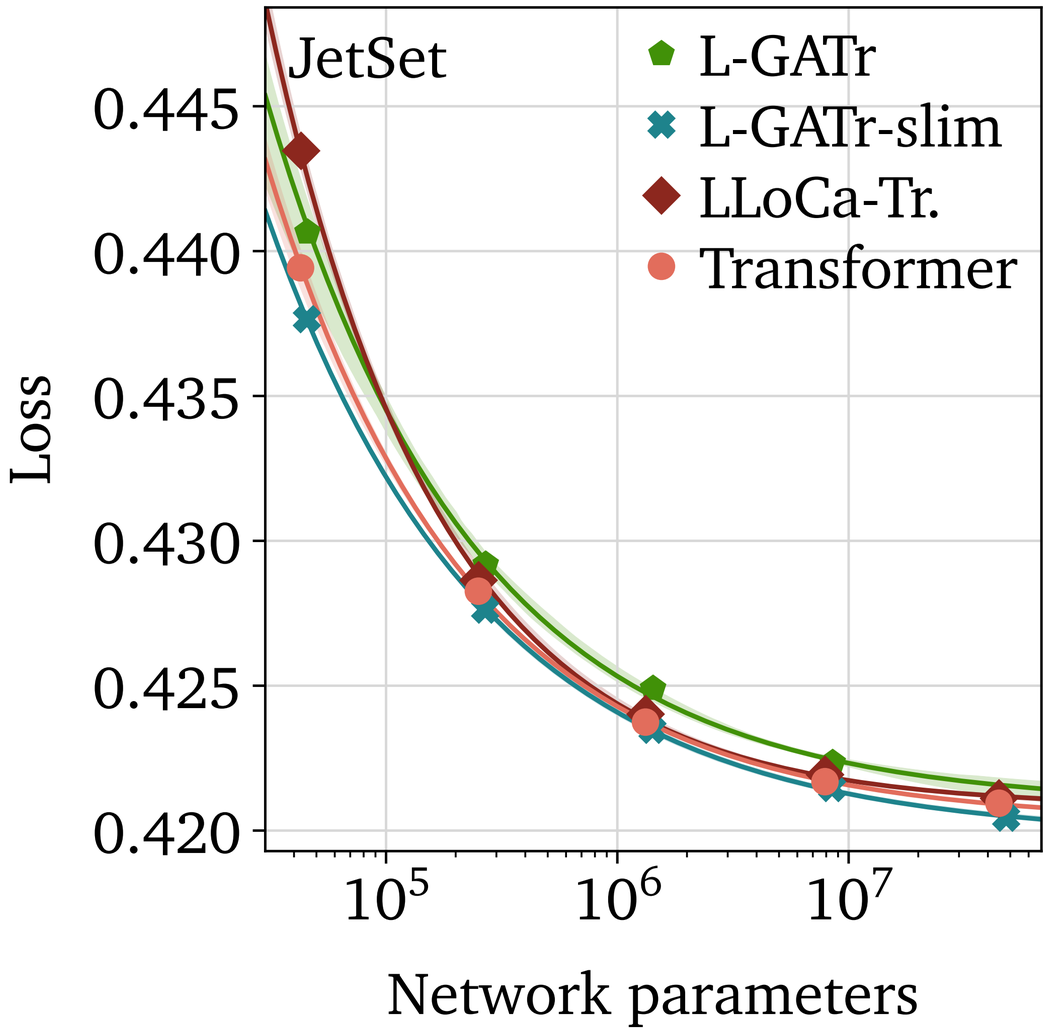
Flavour tagging — JetSet

200M jets from $t\bar{t}$ events,
full ATLAS simulation

Rich feature set beyond kinematics:
impact parameters, uncertainties,
hit counts, …

No auxiliary tasks!

No significant gain from
Lorentz-equivariance!

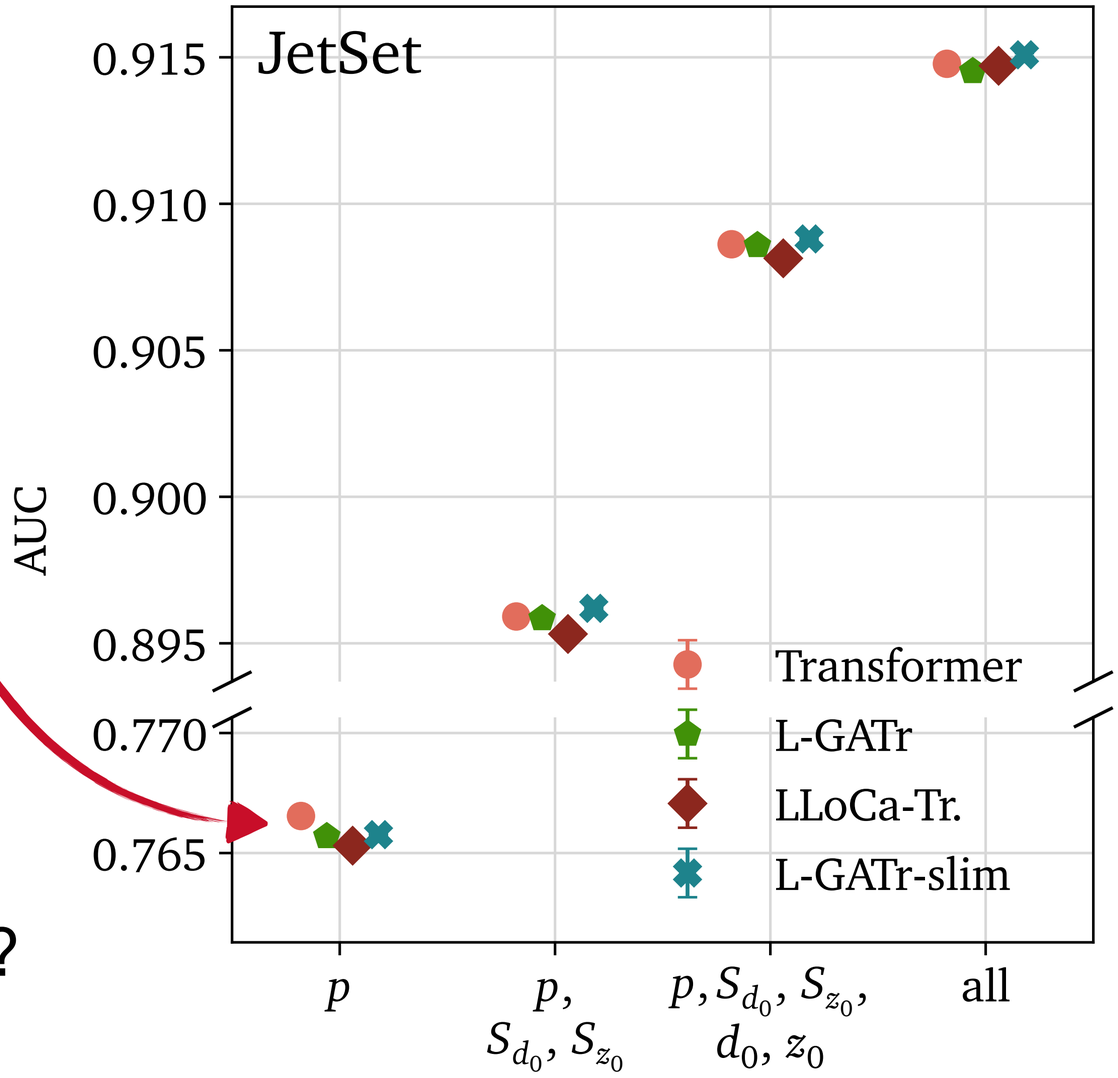


Feature relevance on JetSet

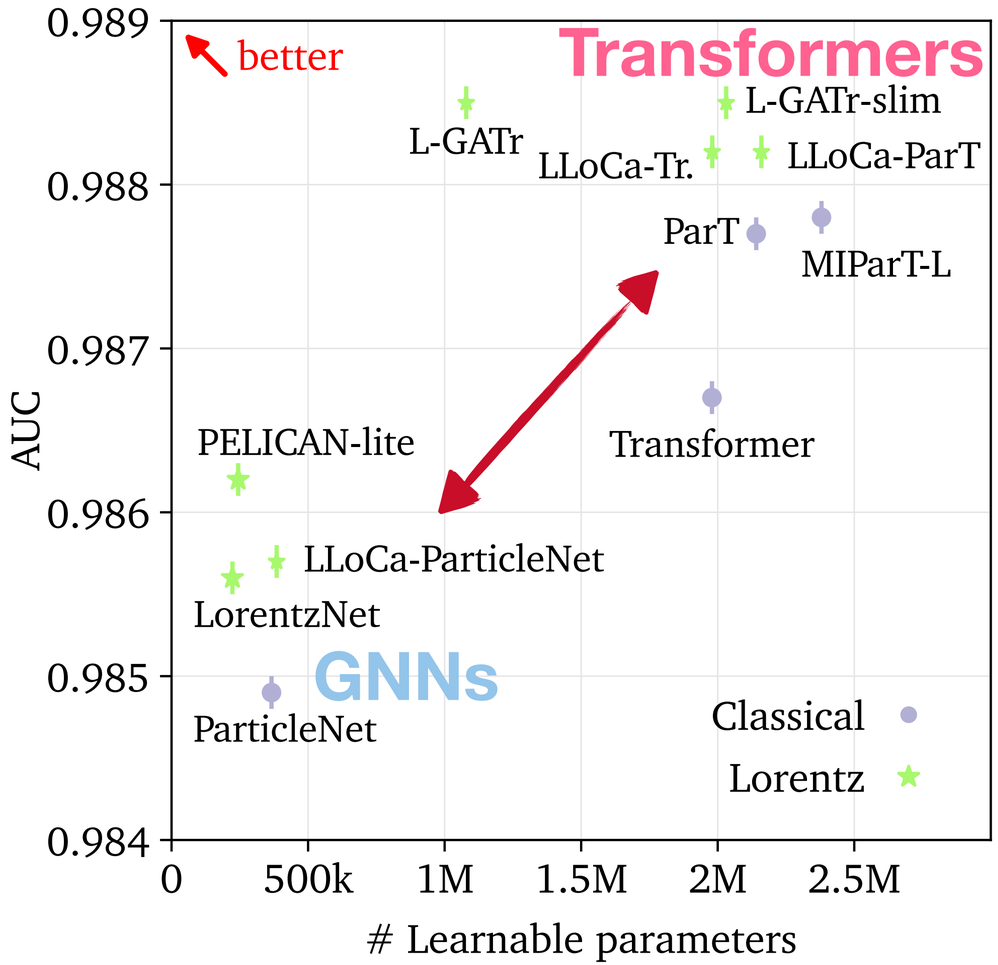
Extra features improve performance

Kinematic features are not useful

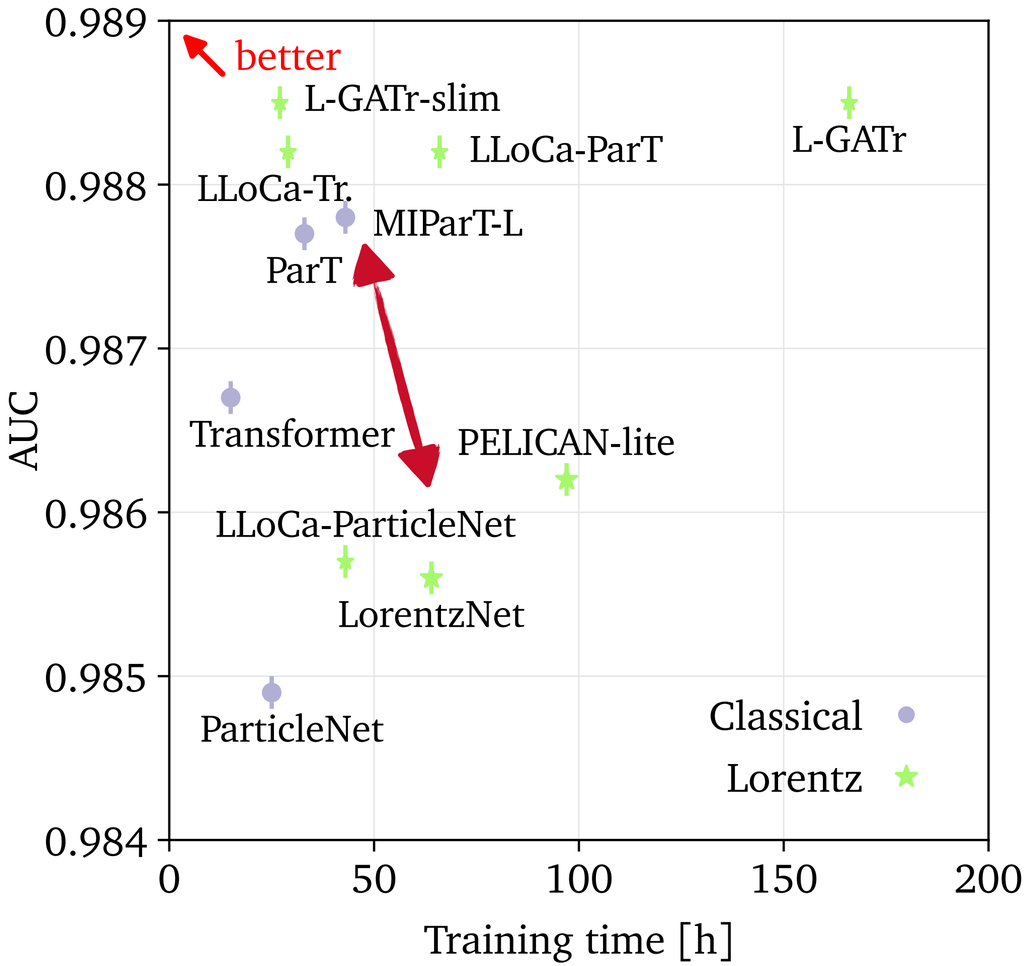
Trajectory information is important, how can we embed them geometrically?



What about graph networks?



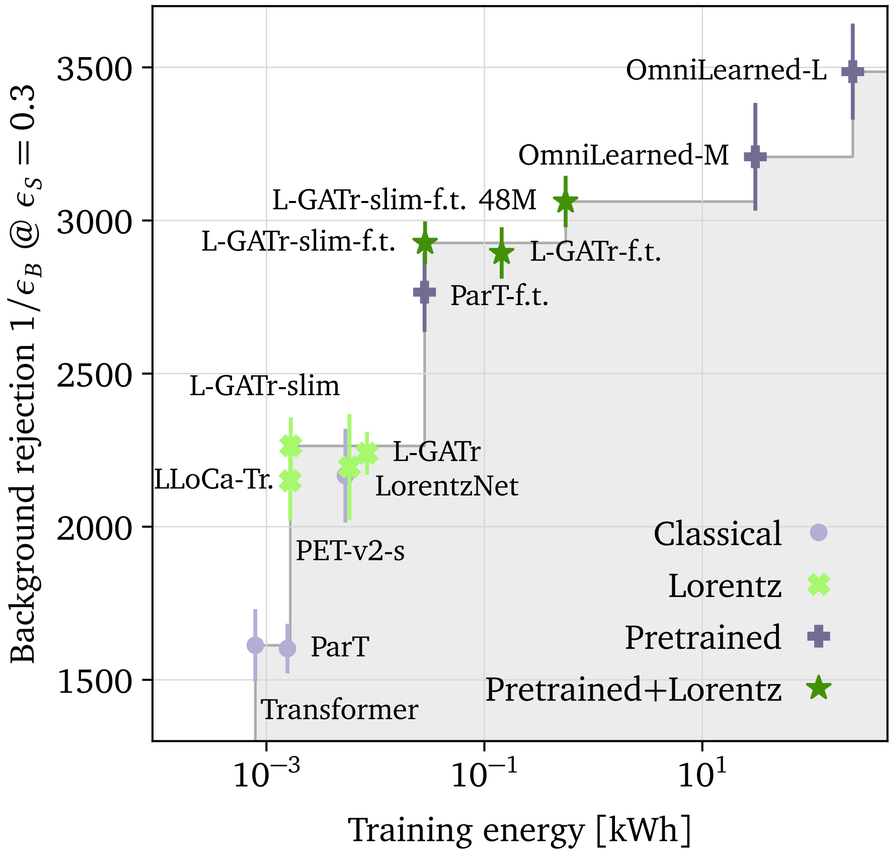
For fixed parameter count there is no clear winner…

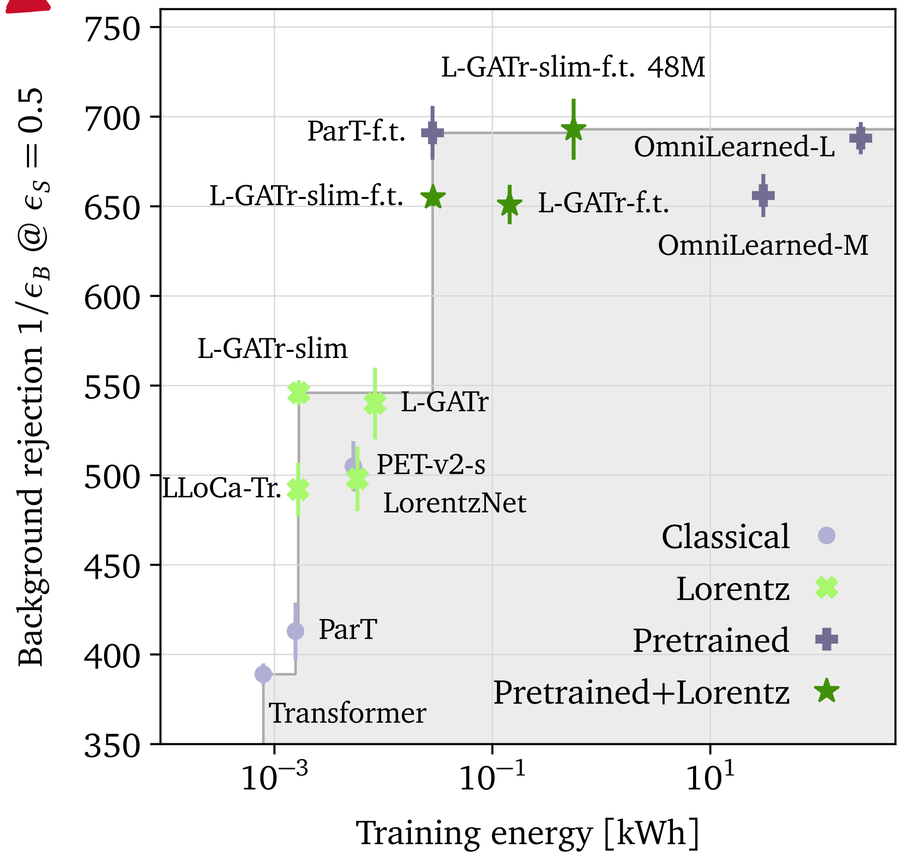


…but GNNs are clearly worse at fixed compute cost

Transfer learning at fixed cost

different evaluation metric

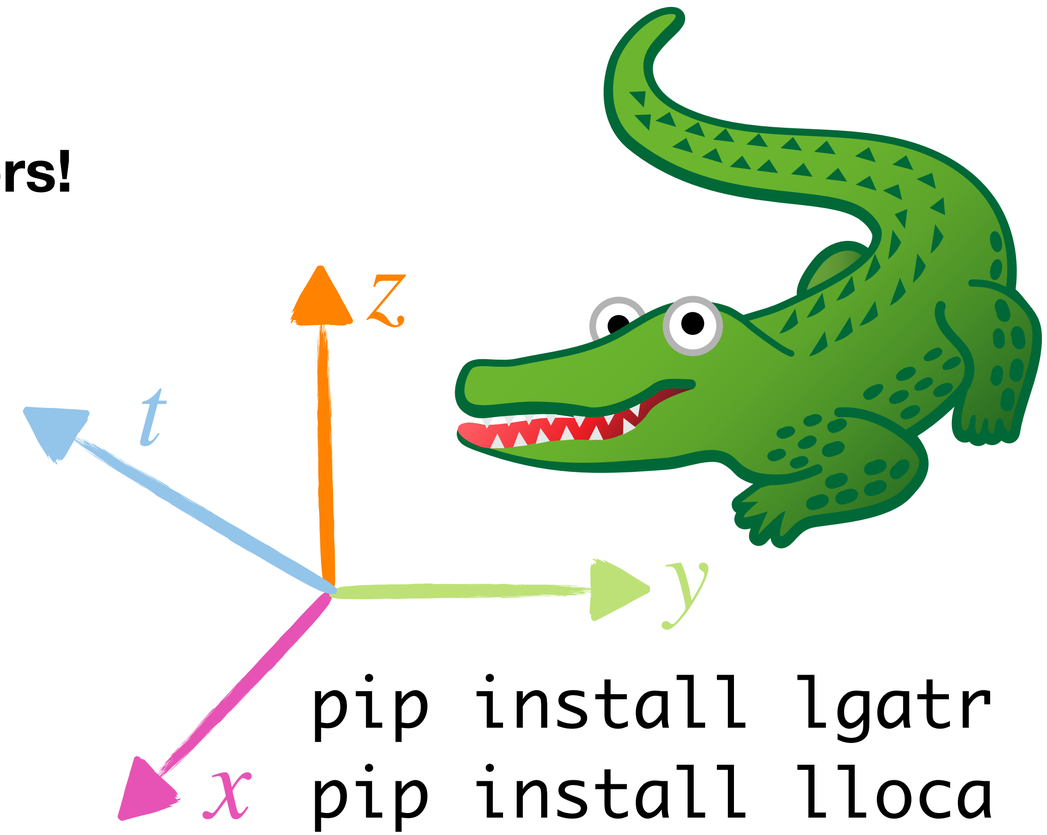




The ‘old’ top tagging dataset is saturated!

Lorentz-equivariant transformers…

- … have efficient implementations and documentation
- … win at fixed cost for large-R jet tagging
- … do not (yet?) win on flavour tagging
- **… are ready for your XYZ ML task on four-vectors!**



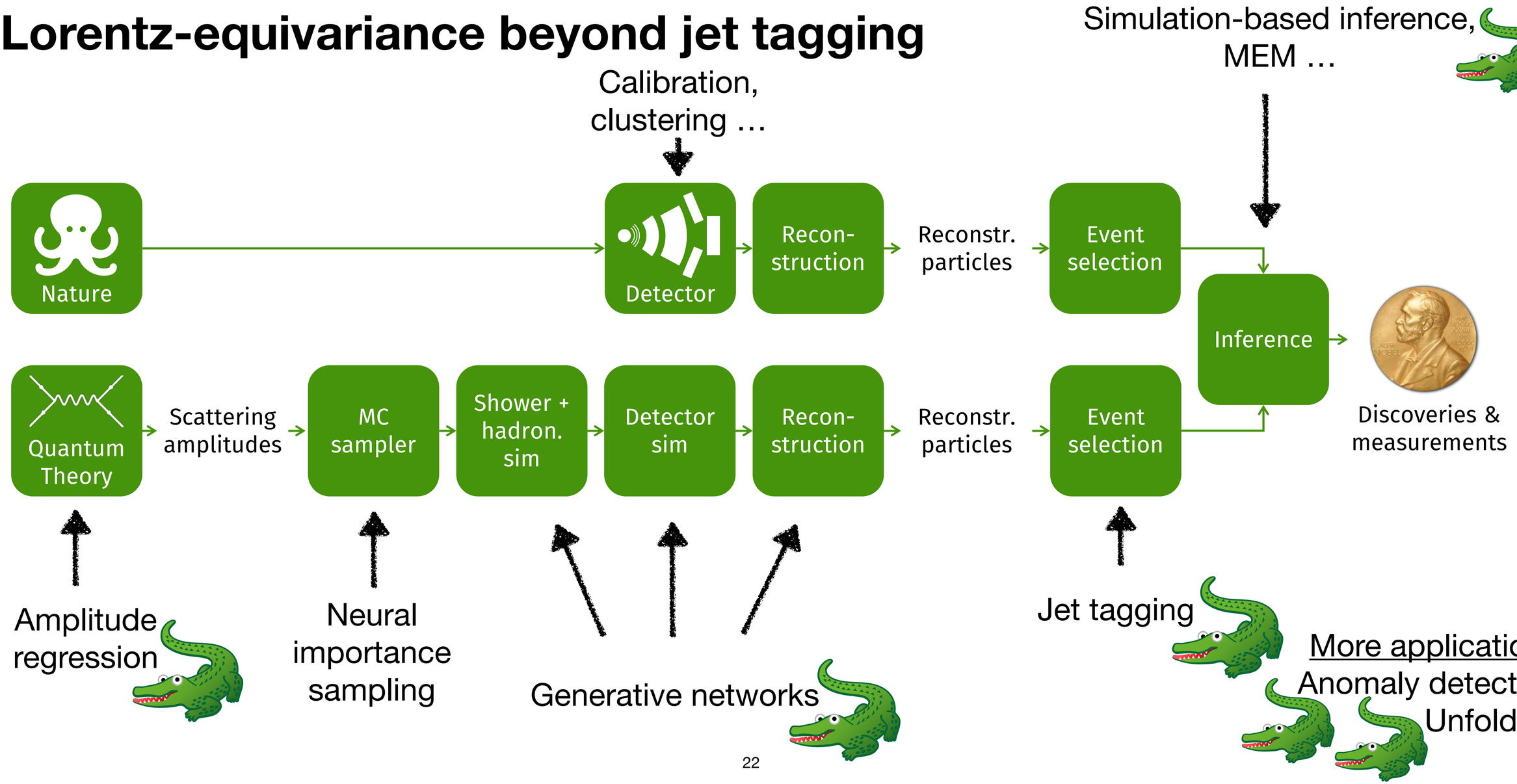
https://heidelberg-hepml.github.io/lgatr/
https://heidelberg-hepml.github.io/lloca/

```
pip install lgatr
pip install lloca
```

Backup slides

#1 Universality

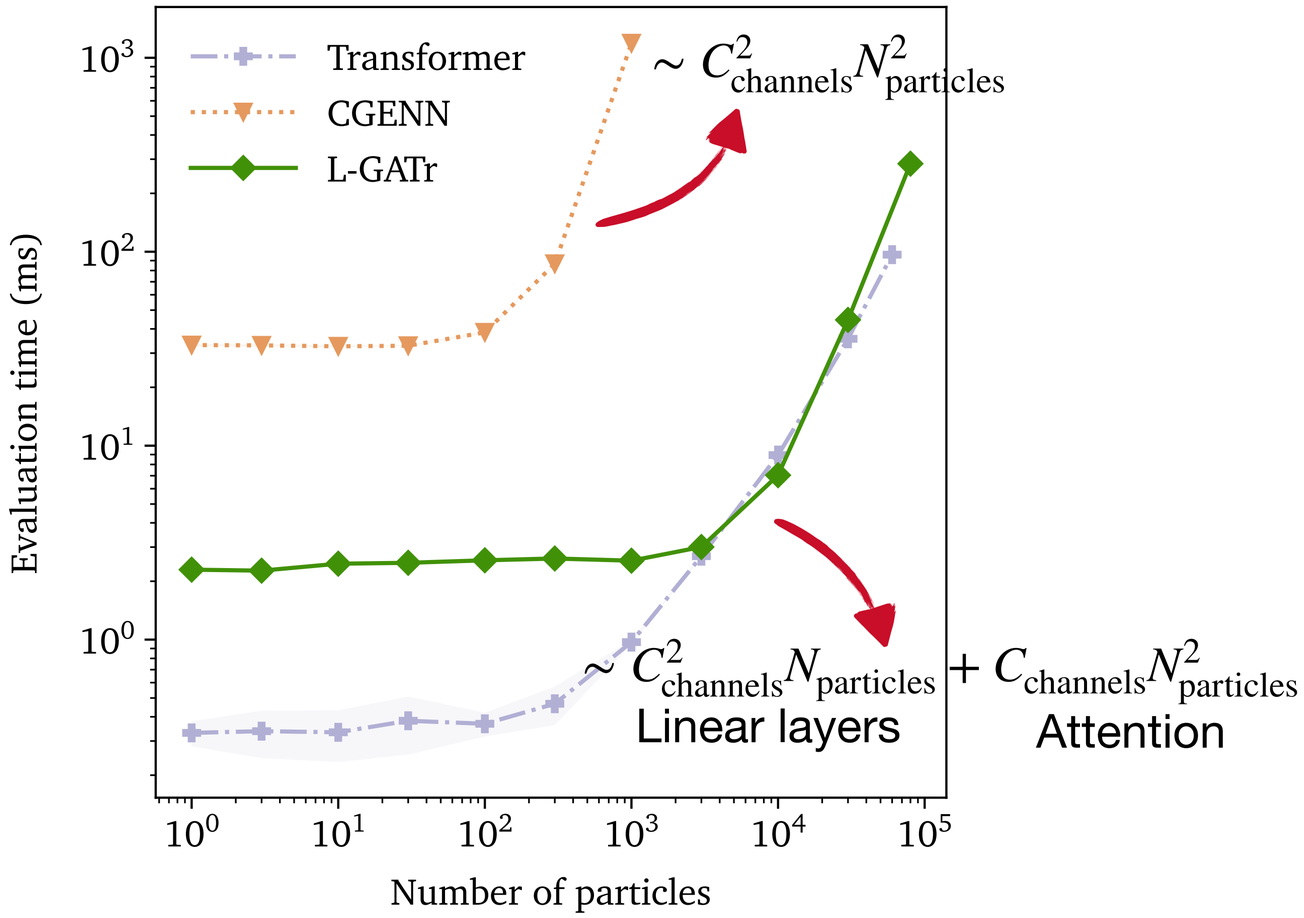
Lorentz-equivariance beyond jet tagging



arXiv:2411.00446
arXiv:2512.17011

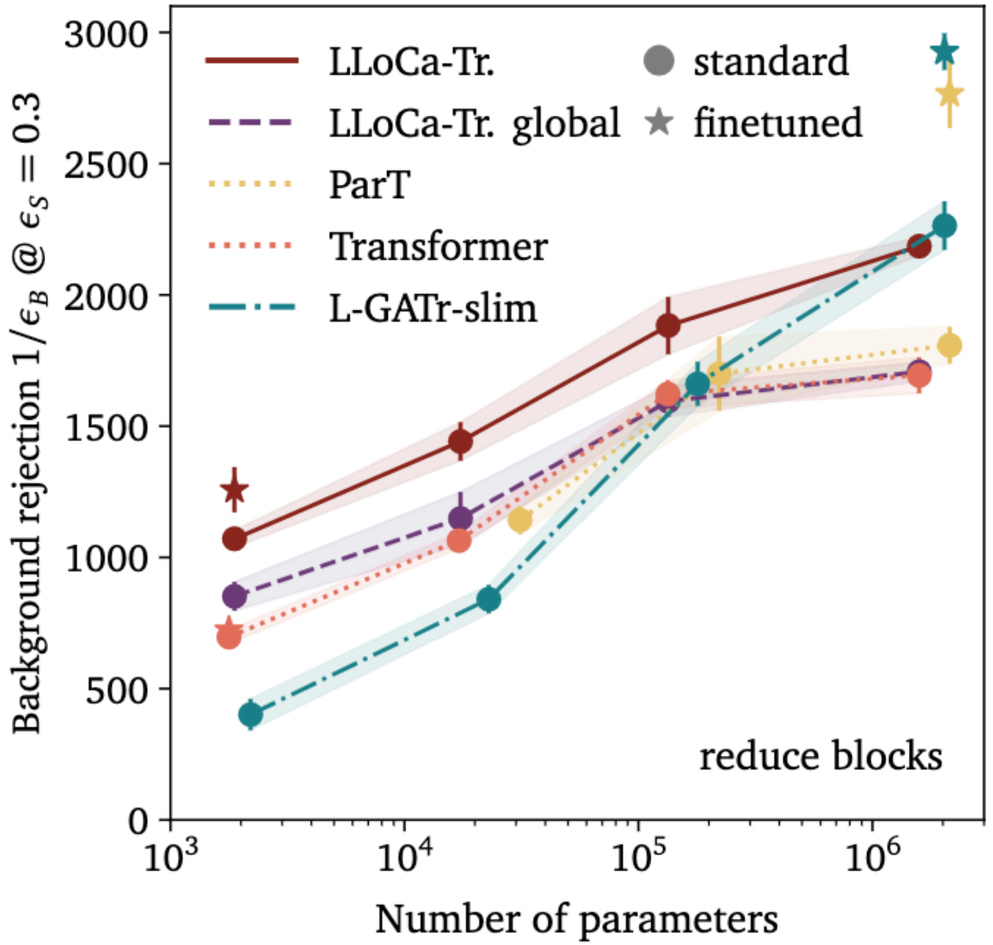
#2 Transformers vs GNNs

Transformers win on fully connected graphs



GNNs don’t have to be fully connected

Transformers don’t have to be large

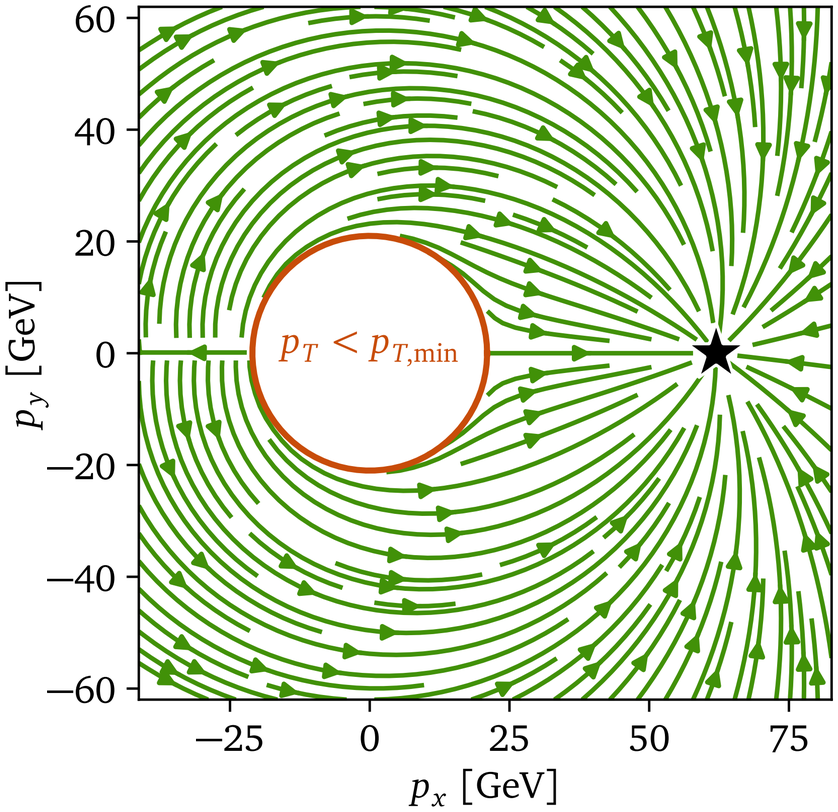


arXiv:2411.00446

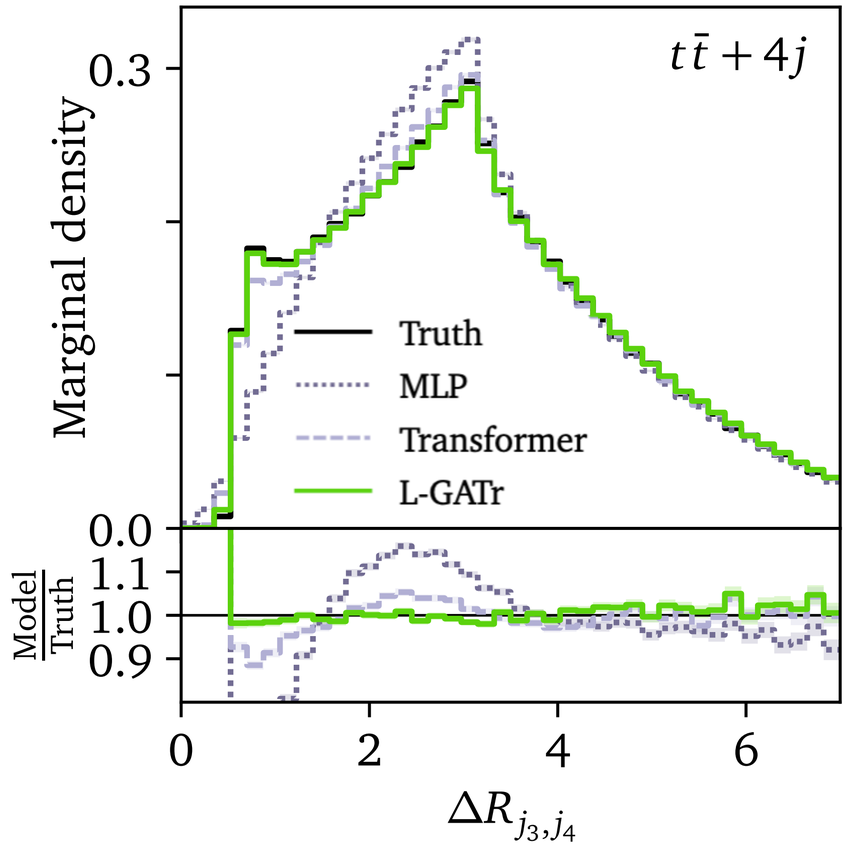
#3 Equivariance > invariance

Applications with vector targets

CFM: Event generation as ODE problem $\partial_t x = v_t(x)$



Lorentz-equivariant network $v_t(x)$



Also: Vector regression

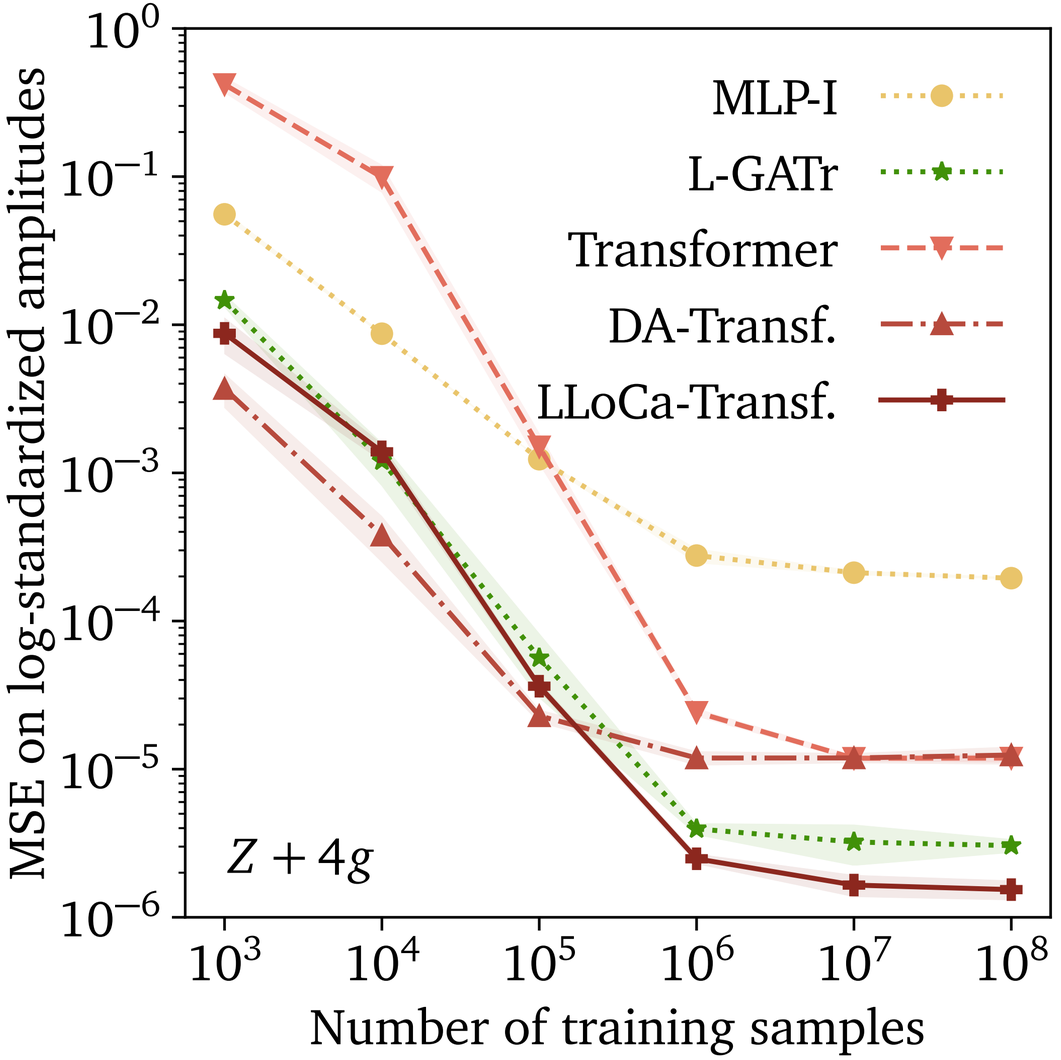
arXiv:2508.14898

#4 Equivariance vs data augmentation

Scaling with training data

Data augmentation wins for little data

QFT amplitude regression



Equivariant networks win at scale

arXiv:2508.14898

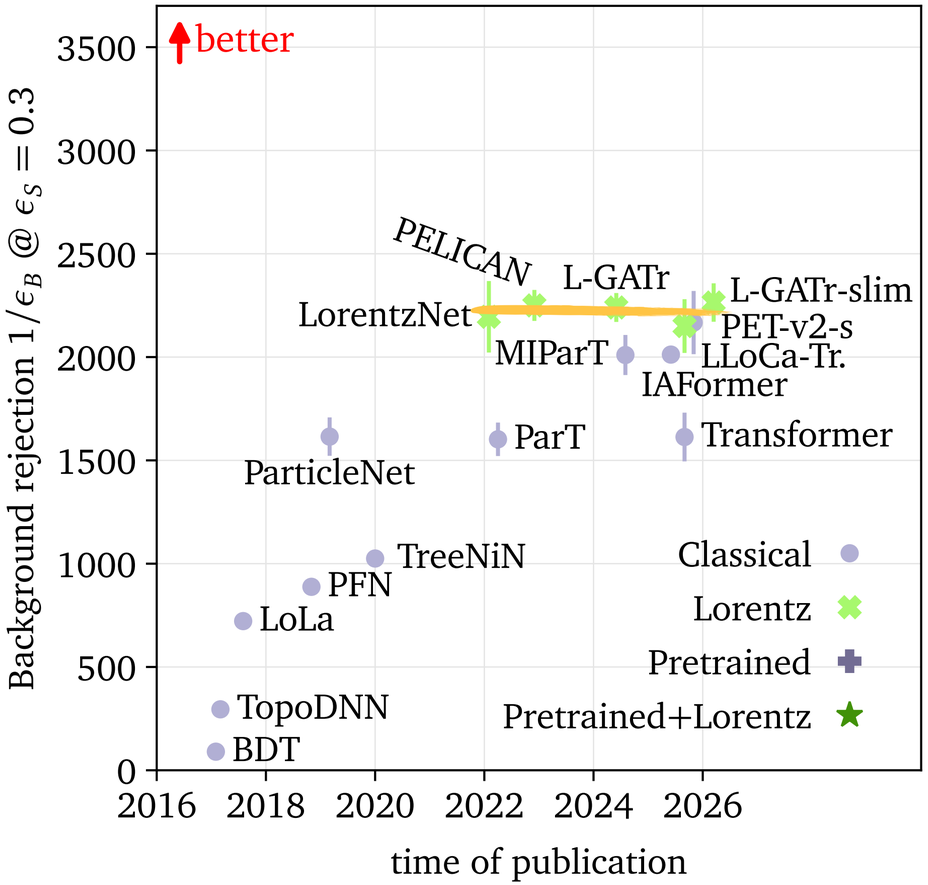
#5 Internal representations

Vectors + scalars is often enough

LLoCa allows arbitrary
internal representations

Amplitude regression:

Method	MSE ($\times 10^{-6}$)
Non-equivariant	11.9 ± 0.8
Global canonical.	5.9 ± 0.4
LLoCa (16 scalars)	54.0 ± 3
LLoCa (single 2-tensor)	2.4 ± 0.3
LLoCa (4 vectors)	2.0 ± 0.2
LLoCa (8 scalars, 2 vectors)	**1.5** ± 0.1



With sufficient expressivity,
all Lorentz+Permutation networks
reach the same performance

arXiv:2508.14898

#6 Lorentz subgroups

Symmetry breaking to subgroup $\mathscr{R}$

‘Architecture’ strategy:
$\mathscr{R}$-equivariant architecture

vs

‘Input’ strategy:
Lorentz-equivariant architecture plus

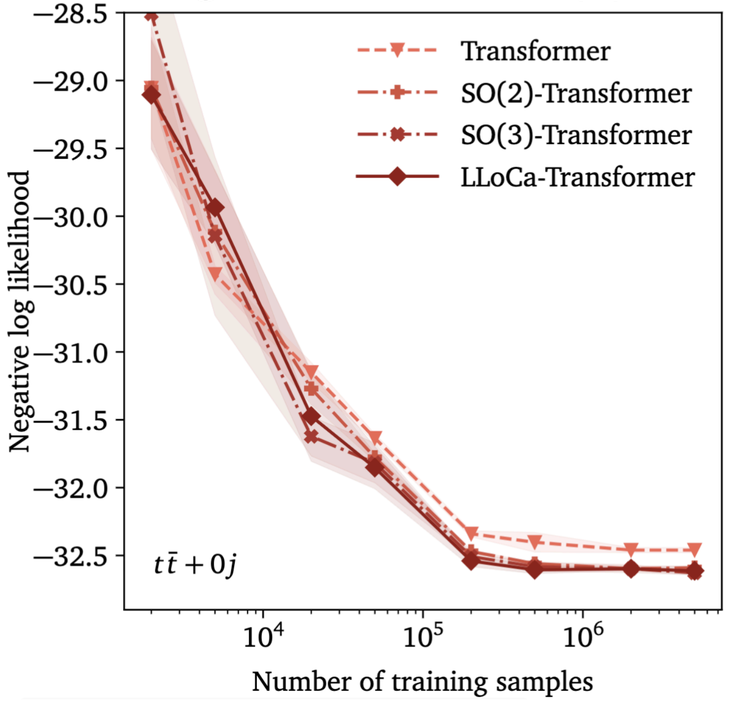
- Reference vectors $\mathscr{R}v = v$
- Auxiliary scalars $s(\mathscr{R}p) = s(p)$

Jet tagging:

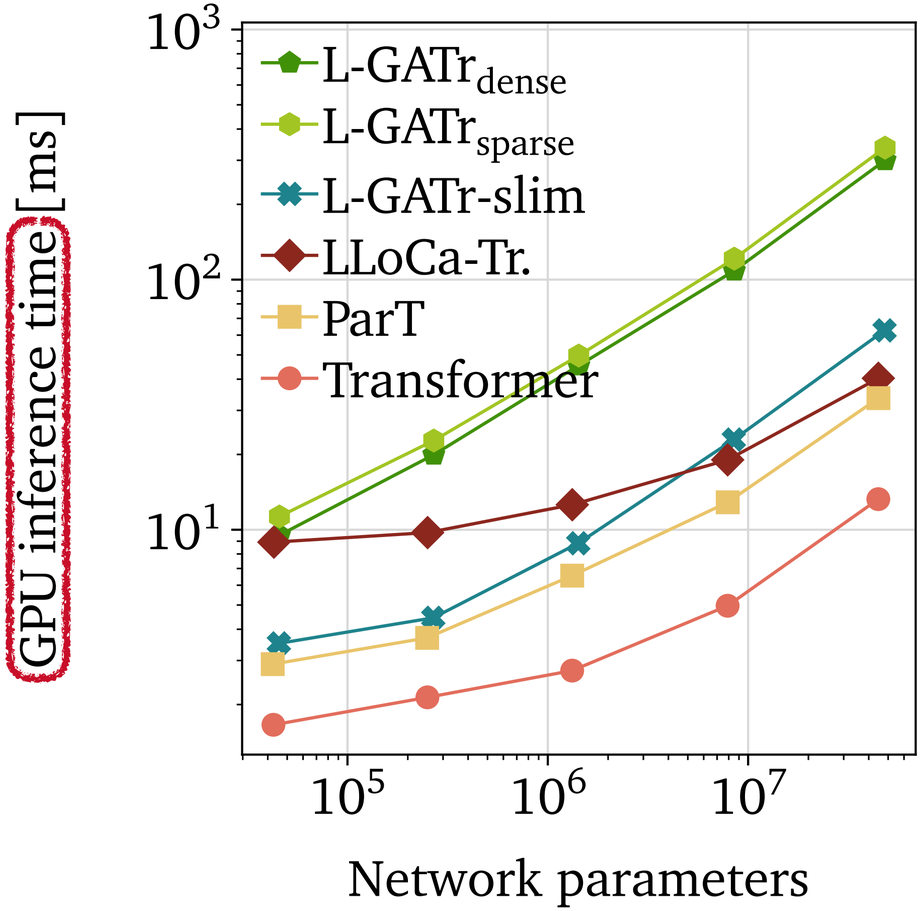
Residual symmetry $\mathscr{R}$	Accuracy Arch.	Accuracy Input	AUC Arch.	AUC Input
Non-equivariant	0.855	0.864	0.9867	0.9982
$\mathrm{SO}(2)_{\mathrm{beam}}$	0.862	0.864	0.9878	0.9882
$\mathrm{SO}^+(1,1)_{\mathrm{beam}} \times \mathrm{SO}(2)_{\mathrm{beam}}$	0.863	0.864	0.9881	0.9882
$\mathrm{SO}(3)$	0.859	0.860	0.9875	0.9876
$\mathrm{SO}^+(1,3)$ (Lorentz)	0.856		0.9870	

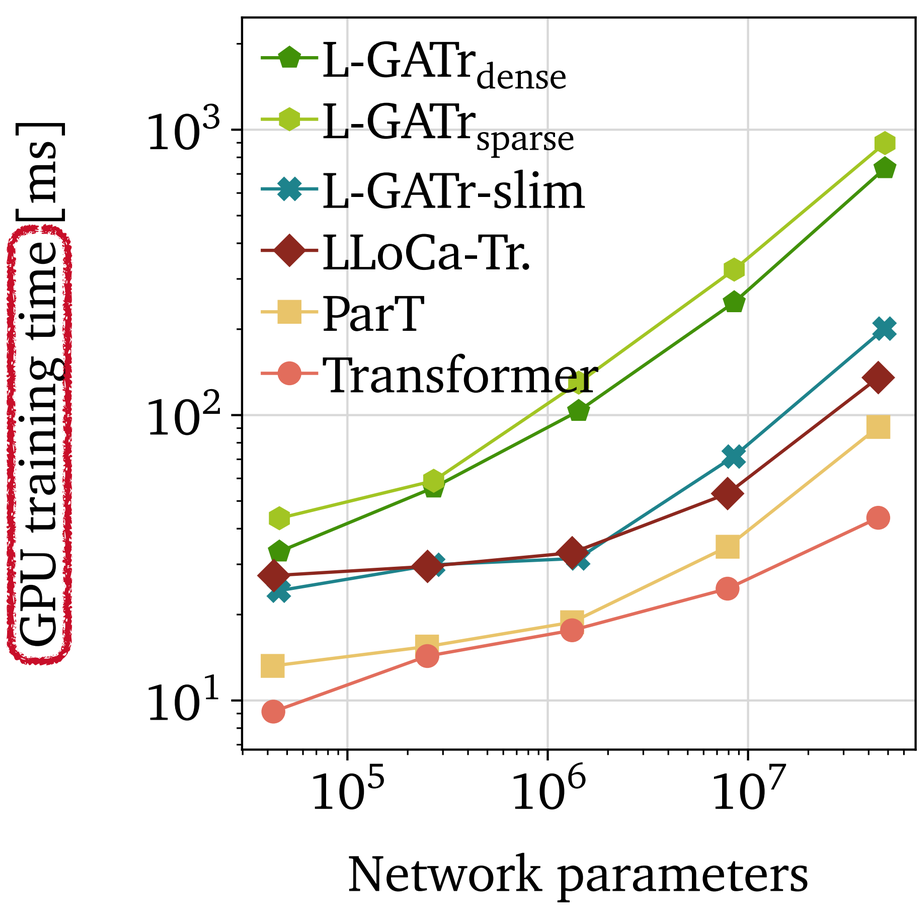
Network learns to break symmetry as needed

Event generation: SO(2) enough



What about training compute?





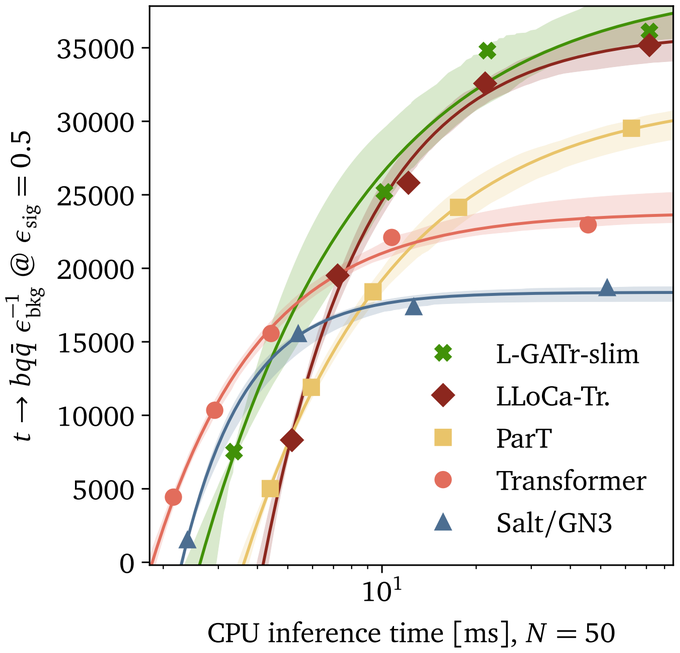
We assume that inference cost is the limiting factors at the LHC

Training and inference time are roughly proportional

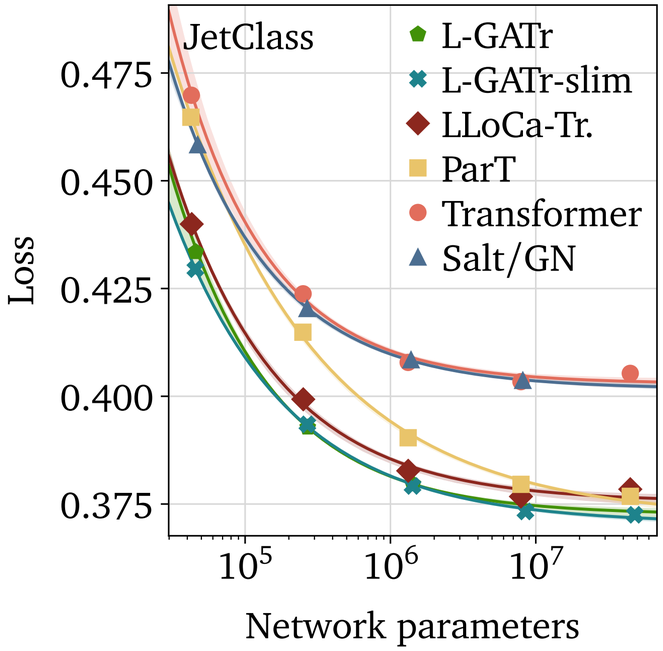
Salt comparison

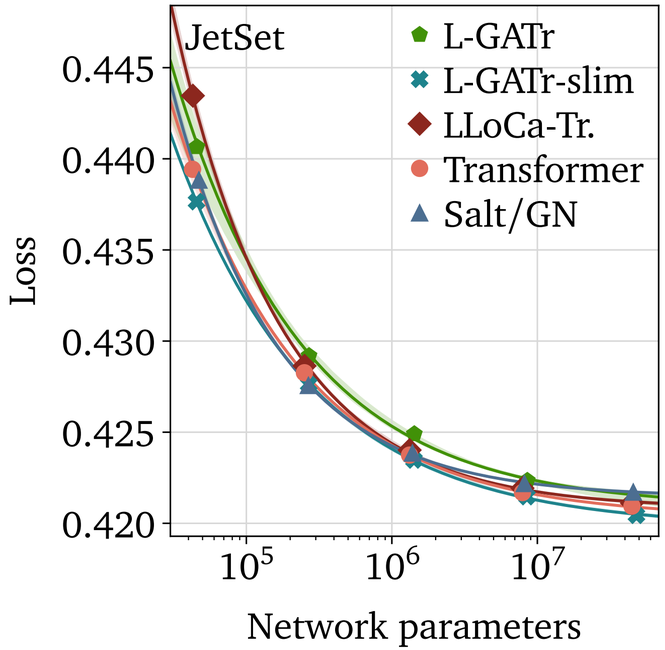
Impact of extra design choices in salt (v0.12)
compared to our baseline transformer?

April 2024 preliminary results:
Salt worse than Transformer?

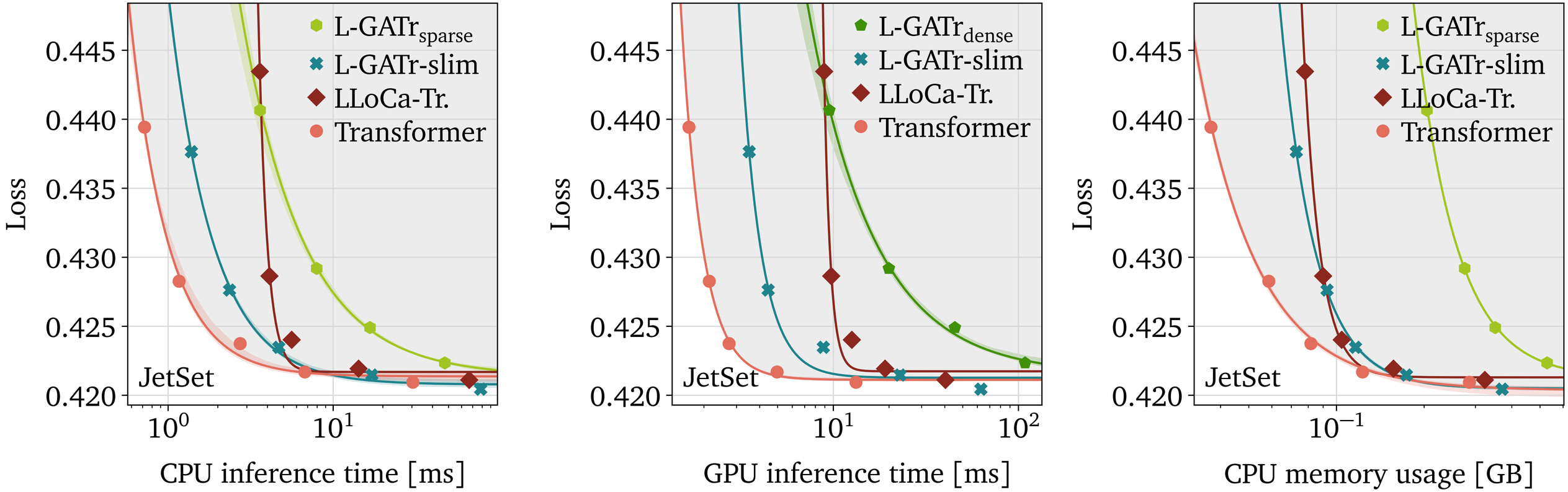


Similar results in final code version
for JetClass and JetSet





Fixed evaluation cost on JetSet



Lorentz-equivariance does not win!

Rejection rates

Small benefit from Lorentz-equivariance for light jets