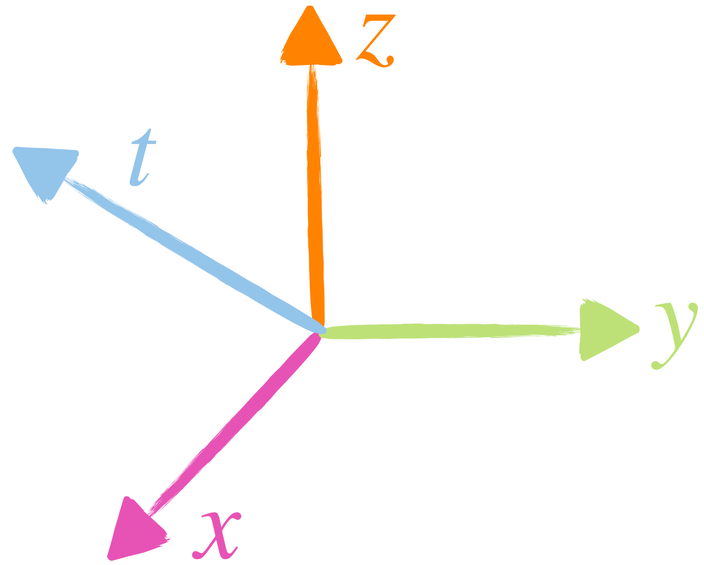
Lorentz Local Canonicalization

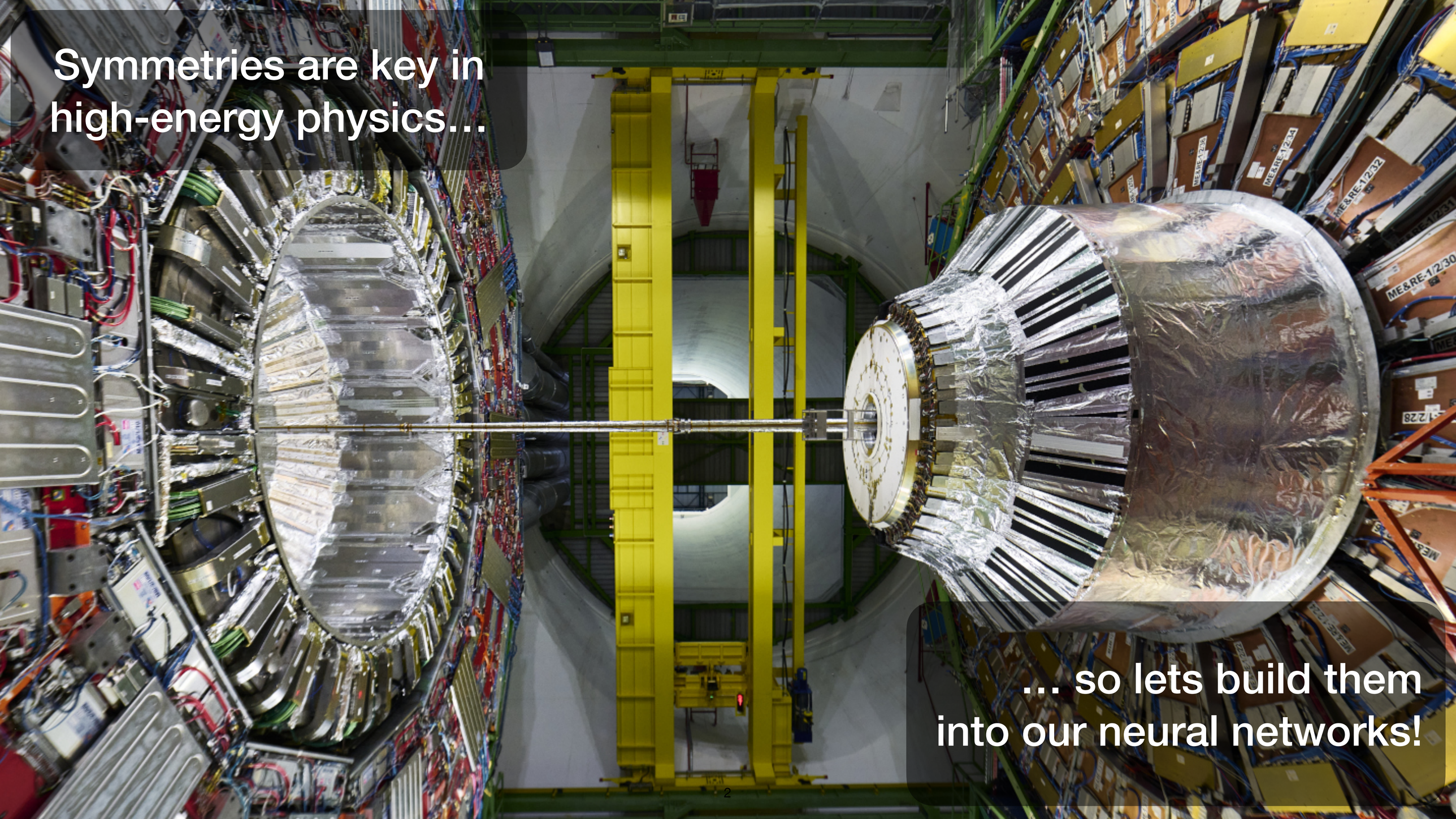
How to make any Network Lorentz-equivariant



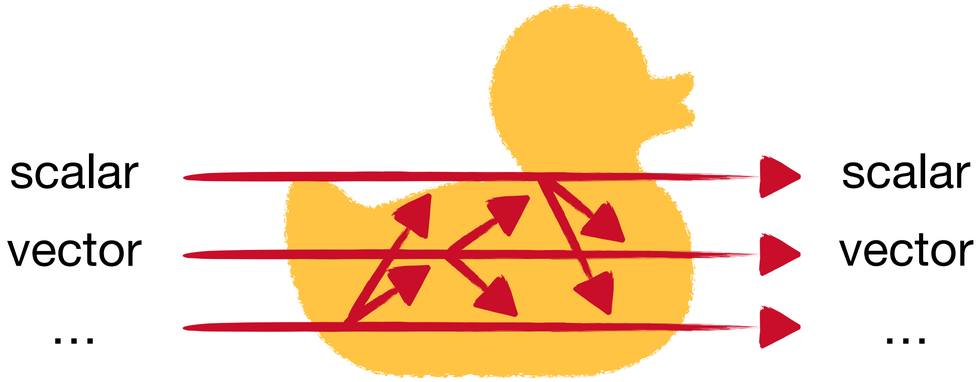
Jonas Spinner*, Luigi Favaro*, Peter Lippmann*,
Sebastian Pitz, Gerrit Gerhartz,
Huilin Qu, Tilman Plehn, Fred A. Hamprecht
arXiv:2505.20280 [stat.ML]
arXiv:2508.????? [hep-ph]

IAIFI Summer Workshop
14.08.2025

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386


Symmetries are key in high-energy physics...
... so lets build them into our neural networks!

Lorentz-equivariance with specialised layers



What about **higher-order** internal representations?

Efficient Lorentz-equivariance?

How to make my **domain-specific** architecture Lorentz-equivariant?

PELICAN

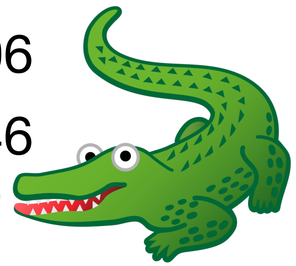
arXiv:2211.00454
arXiv:2307.16506

LorentzNet

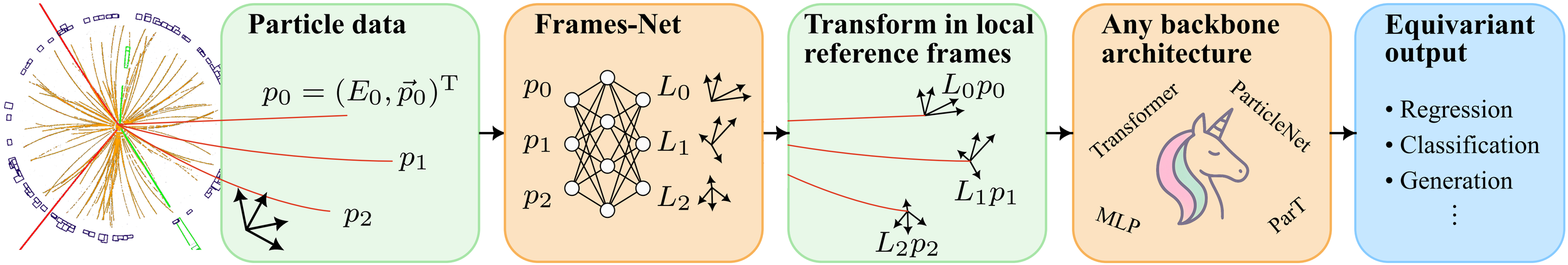
arXiv:2201.08187

L-GATr

arXiv:2405.14806
arXiv:2411.00446
pip install lgatr

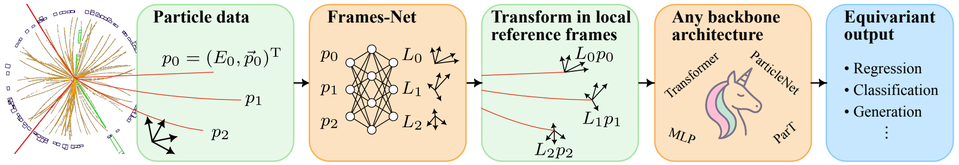


Lorentz Local Canonicalization



LLoCa

Local frames



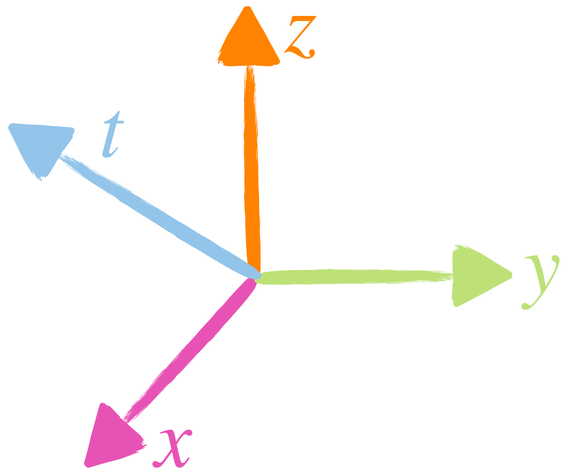
Features $x_L = Lx$ in local frame are **invariant** by definition:

$$x_L = Lx \xrightarrow{\Lambda} L'x' = L'\Lambda x \overset{!}{=} x_L \qquad \Rightarrow \qquad L' = L\Lambda^{-1}$$

Achieve this by stacking **Lorentz vectors** $u_a{}^\mu$, $a = 0,1,2,3$:

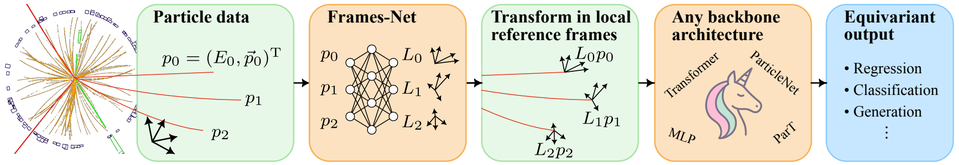
$$L = \begin{pmatrix} — & u_0^T g & — \\ — & u_1^T g & — \\ — & u_2^T g & — \\ — & u_3^T g & — \end{pmatrix} \xrightarrow{\Lambda} L' = \begin{pmatrix} — & u_0^T \Lambda^T g & — \\ — & u_1^T \Lambda^T g & — \\ — & u_2^T \Lambda^T g & — \\ — & u_3^T \Lambda^T g & — \end{pmatrix} = L\Lambda^{-1}$$

$$\sum_{\mu,\nu} u_a{}^\mu u_b{}^\nu g_{\mu\nu} = g_{ab}$$



LLoCa

Equivariance from canonicalisation



Latent features in the local frame $f_L = \rho(L)f$ are **invariant**

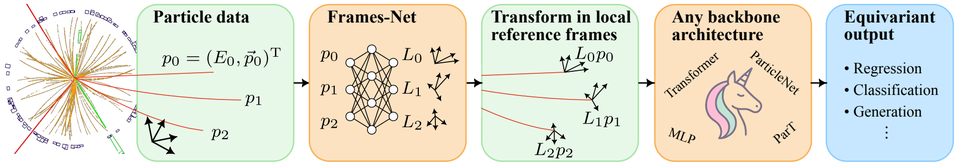
$$f_L \xrightarrow{\Lambda} f'_L = \rho(L')f' = \rho(L\Lambda^{-1})\rho(\Lambda)f = \rho(L\Lambda^{-1}\Lambda)f = \rho(L)f = f_L$$

Output features in the global frame $y = \rho(L^{-1})y_L$ are **equivariant**

$$y \xrightarrow{\Lambda} y' = \rho(L'^{-1})y'_L = \rho(\Lambda L^{-1})y_L = \rho(\Lambda)\rho(L^{-1})y_L = \rho(\Lambda)y$$

LLoCa

Message-passing between local frames



Standard message-passing

$$f_i^{\text{updated}} = \psi\left(f_i, \bigoplus_{j=1}^{N} \phi(m_j)\right)$$

Tensorial message-passing

between distinct local frames

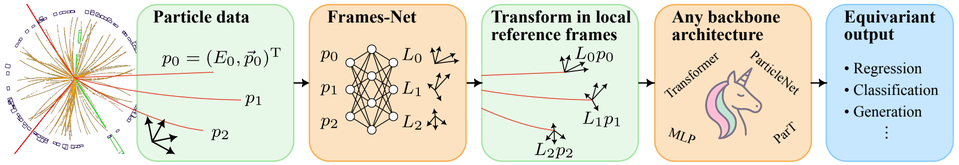
$$f_{i,L_i}^{\text{updated}} = \psi\left(f_{i,L_i}, \bigoplus_{j=1}^{N} \phi\left(\rho_m(L_i L_j^{-1}) m_{j,L_j}\right)\right)$$

Message representation ρ_m is a hyperparameter

Invariant frame transformation matrix $L_i L_j^{-1}$

LLoCa

How to construct local frames



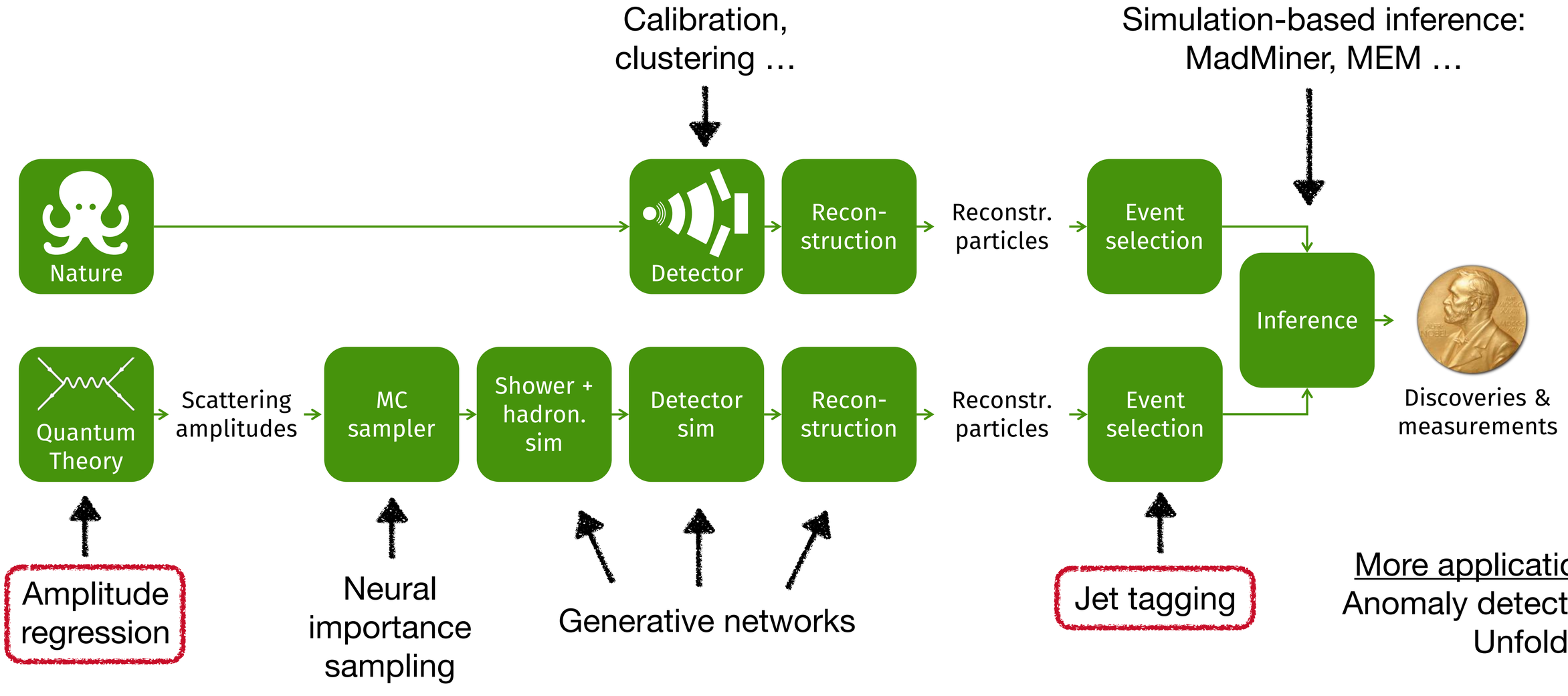
1) Predict vectors $v_{i,k}$ with simple **Lorentz-equivariant network**

$$v_{i,k} = \sum_{j=1}^{N} \text{softmax}\Big(\varphi_k(s_i, s_j, \langle p_i, p_j \rangle)\Big)(p_i + p_j) \qquad \text{for } k = 0,1,2.$$

Frames-Net (MLP)

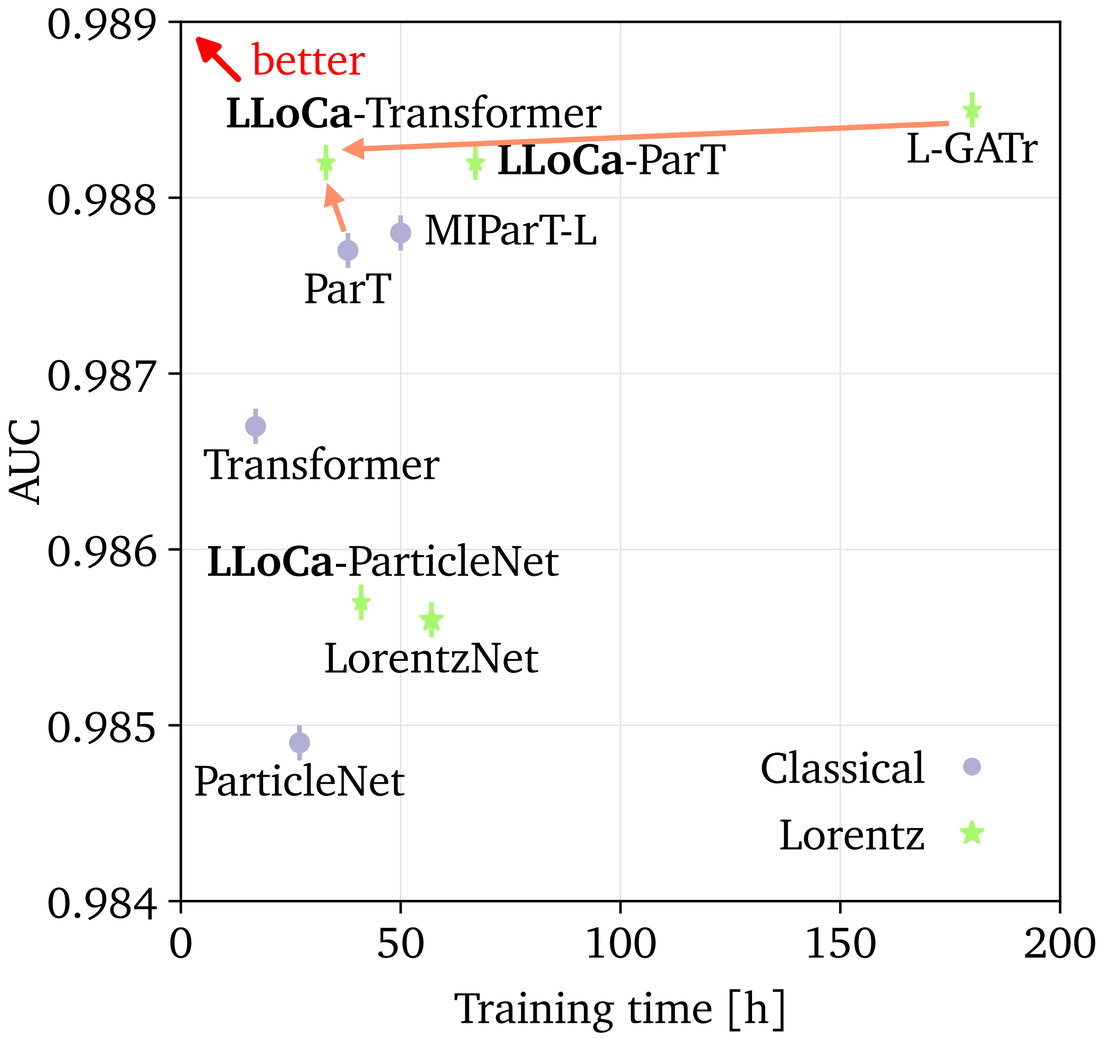
2) Orthonormalize vectors with **Gram-Schmidt algorithm** to obtain $u_{i,a}$

The LHC ML workflow



Jet Tagging

Speed vs performance on JetClass

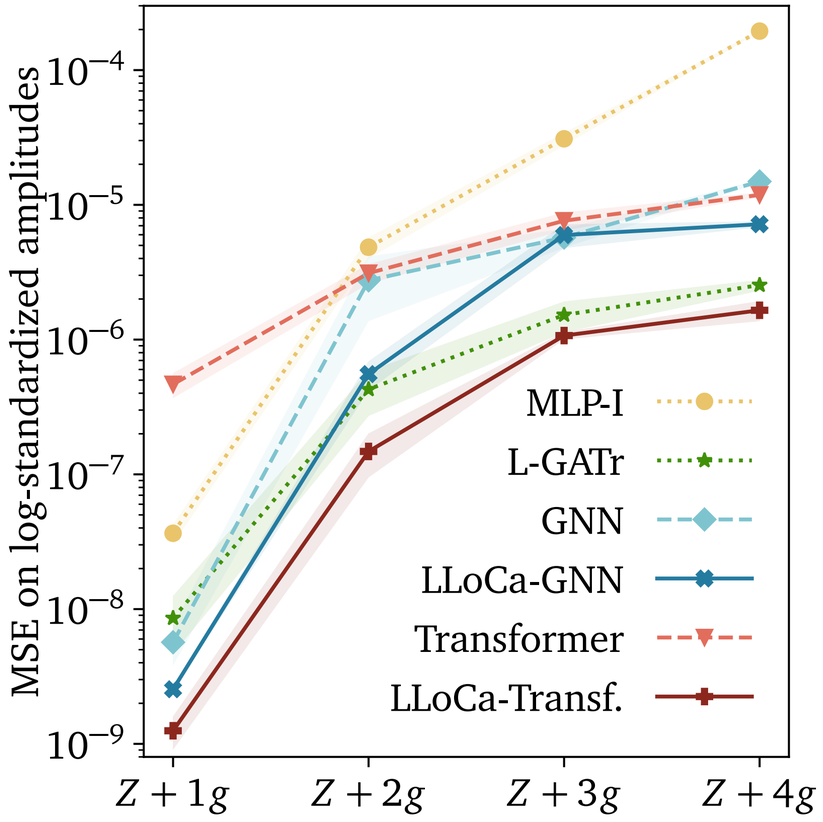


LLoCa-Transformer outperforms ParT while running 10% faster

LLoCa-Transformer matches L-GATr performance at 6x speedup

Amplitude regression

Speed vs performance



Model	MSE$\times 10^{-6}$ (↓)	FLOPs	Time
MLP-I [38]	195.0 ± 2	0.1M	0.6h
LGATr [38]	2.5 ± 0.2	1160.0M	19.5h
GNN	14.9 ± 0.3	9.9M	1.3h
LLoCa-GNN	7.2 ± 0.3	10.6M	2.3h
Transformer	11.9 ± 0.5	10.6M	3.5h
LLoCa-Transformer	**1.5** ± 0.1	11.3M	4.5h

LLoCa-Transformer and L-GATr achieve similar performance

… but LLoCa-Transformer is 4x faster and requires 100x less FLOPs!

Amplitude regression

Effect of internal representation ρ_m

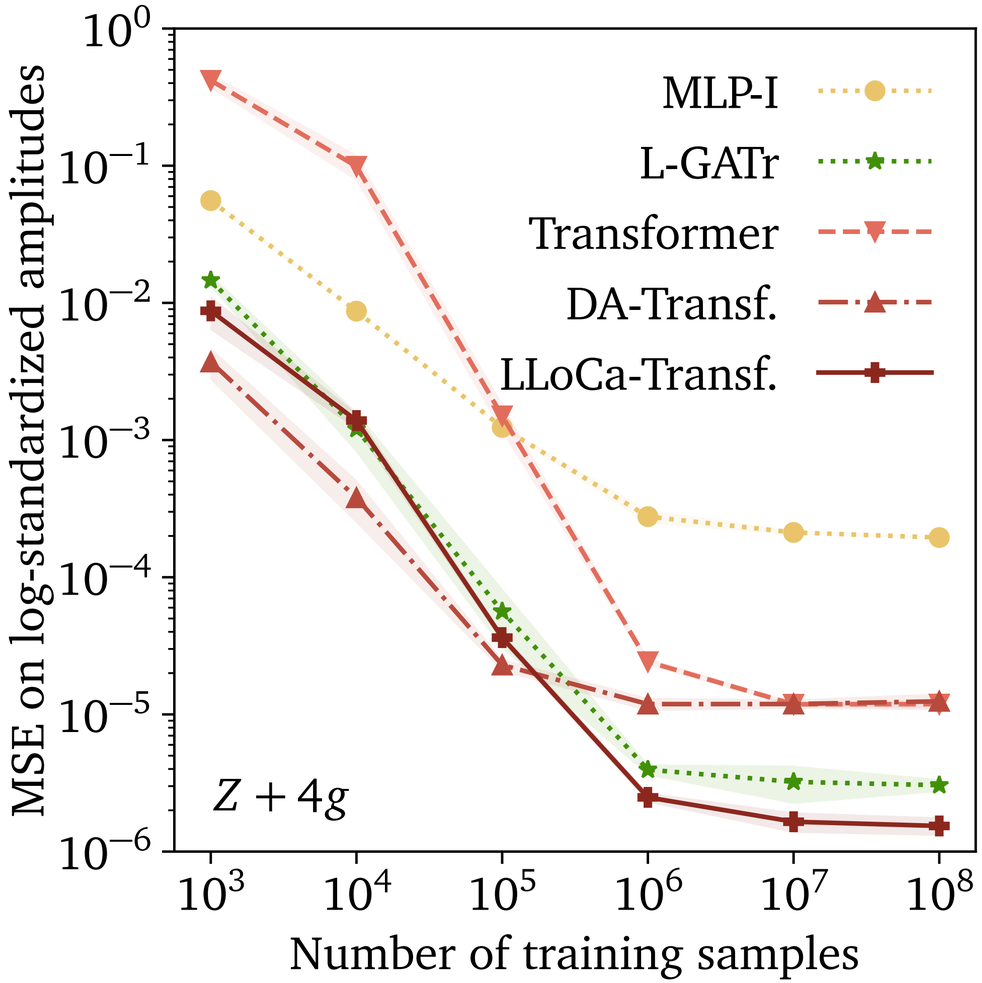
Method	MSE ($\times 10^{-6}$)
Non-equivariant	11.9 ± 0.8
Global canonical.	5.9 ± 0.4
LLoCa (16 scalars)	54.0 ± 3
LLoCa (single 2-tensor)	2.4 ± 0.3
LLoCa (4 vectors)	2.0 ± 0.2
LLoCa (8 scalars, 2 vectors)	**1.5** ± 0.1

LLoCa handles arbitrary tensorial representations

Across our experiments, an equal mix of scalar and vector representations works best.

Amplitude regression

Equivariance vs data augmentation



Data augmentation (DA) emerges from LLoCa as a special choice of local frame

Lorentz-equivariance wins in the large-data regime

Data augmentation wins in the small-data regime

Jet Tagging

Lorentz symmetry breaking

Symmetry breaking Network	Architecture Accuracy	Architecture AUC	Input Accuracy	Input AUC
Non-equivariant	0.855	0.9867	0.863	0.9880
SO(2)-equivariant	0.862	0.9878	0.864	0.9882
Lorentz-equivariant	0.856	0.9870	0.856	0.9870

= LLoCa-Transformer

Jet tagging score is only SO(2)-invariant

Lorentz-equivariance + symmetry breaking outperforms SO(2)-equivariance!

Summary

- Canonicalization makes Lorentz-equivariance fast + flexible
  - More efficient than specialised layers
  - Arbitrary internal representations
  - Make any network Lorentz-equivariant
  - Can easily reduce the symmetry group
- Data augmentation wins in the small-data regime


Luigi Favaro


Peter Lippmann


Sebastian Pitz


Gerrit Gerhartz


Huilin Qu

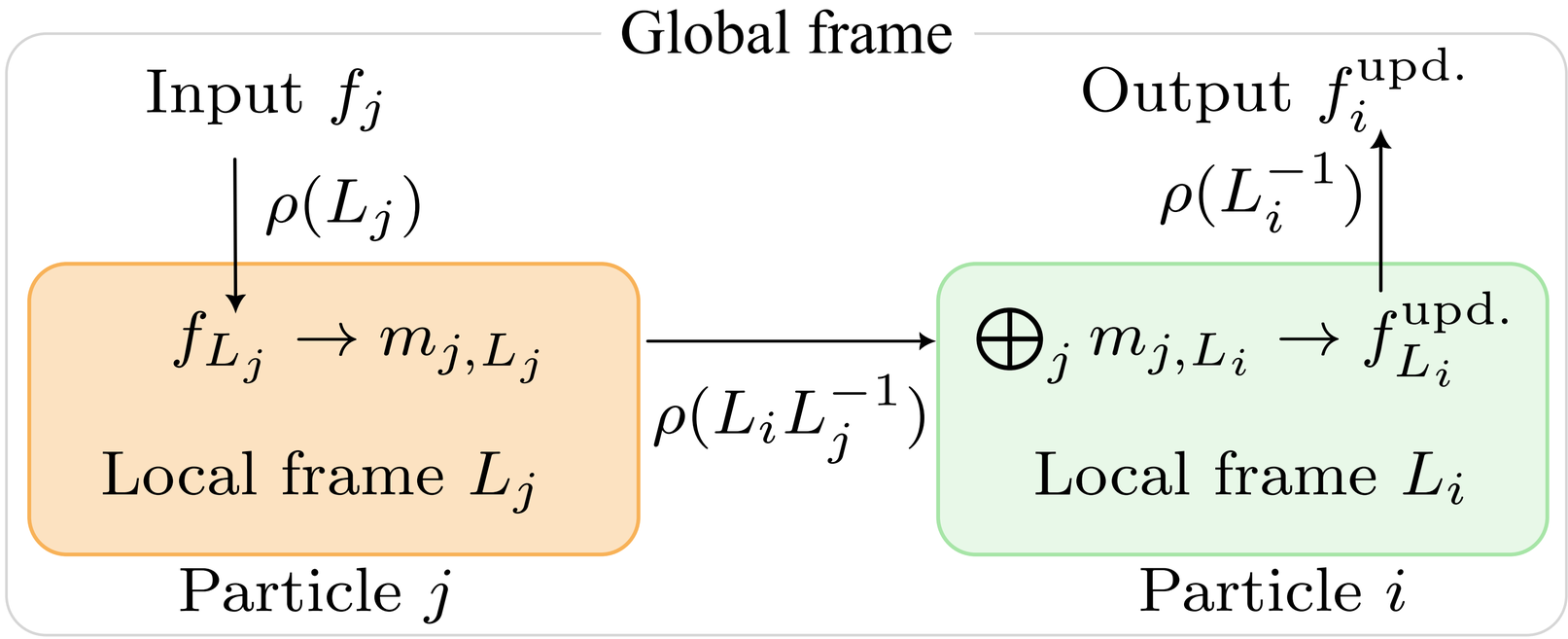

Tilman Plehn


Fred Hamprecht

Bonus material

LLoCa

Tensorial message-passing



LLoCa

Efficient Minkowski attention

$$f^{\text{updated}}_{i,L_i} = \sum_{j=1}^{N} \text{softmax}_j\left(\frac{1}{\sqrt{d}}\left\langle q_{i,L_i}, \rho_k(L_i L_j^{-1}) k_{j,l_j}\right\rangle\right)\rho_v\left(L_i L_j^{-1}\right) v_{j,L_j}$$

$$\left\langle q_{i,L_i}, \rho_k(L_i L_j^{-1}) k_{j,L_j}\right\rangle = \left\langle \rho_k(L_i^{-1}) q_{i,L_i}, \rho_k(L_j^{-1}) k_{j,L_j}\right\rangle$$

Symmetry breaking

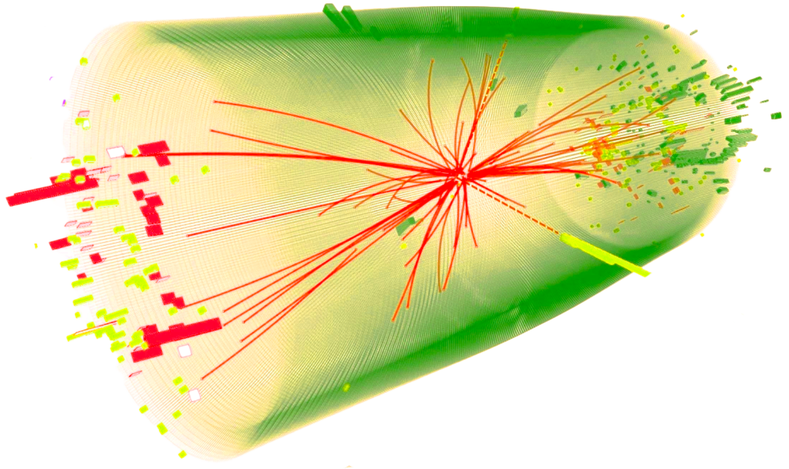
Symmetry breaking with reference vectors

The jet tagging score is not Lorentz-equivariant

- Beam direction breaks invariance under rotations around the x- and y-axis
- Detector breaks boost invariance

Add reference vectors as extra particles to break Lorentz symmetry

- Beam reference vector: $x^V = (1,0,0,1)$
- Time reference vector: $x^V = (1,0,0,0)$



All Lorentz-equivariant taggers use symmetry-breaking inputs

Jet tagging

Top tagging dataset

Network	Accuracy	AUC	$1/\epsilon_B$ ($\epsilon_S = 0.5$)	$1/\epsilon_B$ ($\epsilon_S = 0.3$)
PFN [56]	0.932	0.9819	247 ± 3	888 ± 17
ParticleNet [55]	0.940	0.9858	397 ± 7	1615 ± 93
ParT [21]	0.940	0.9858	413 ± 16	1602 ± 81
MIParT [53]	0.942	0.9868	505 ± 8	2010 ± 97
LorentzNet* [11]	0.942	0.9868	498 ± 18	2195 ± 173
CGENN* [13]	0.942	0.9869	500	2172
PELICAN* [12]	0.9426 ± 0.0002	0.9870 ± 0.0001	–	2250 ± 75
L-GATr* [14]	0.9423 ± 0.0002	0.9870 ± 0.0001	540 ± 20	2240 ± 70
LLoCa-Transformer*	0.9416 ± 0.0001	0.9866 ± 0.0001	492 ± 15	2150 ± 130

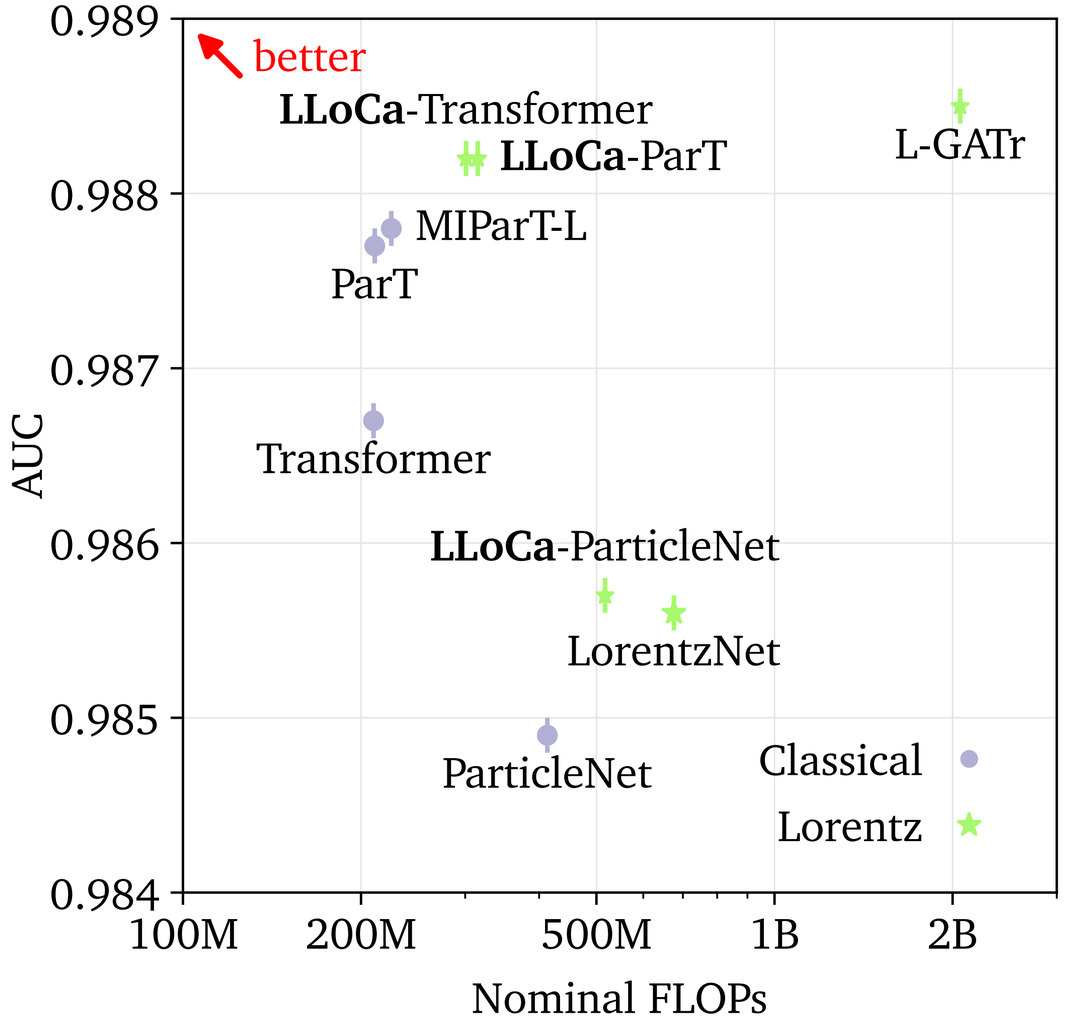
Jet tagging

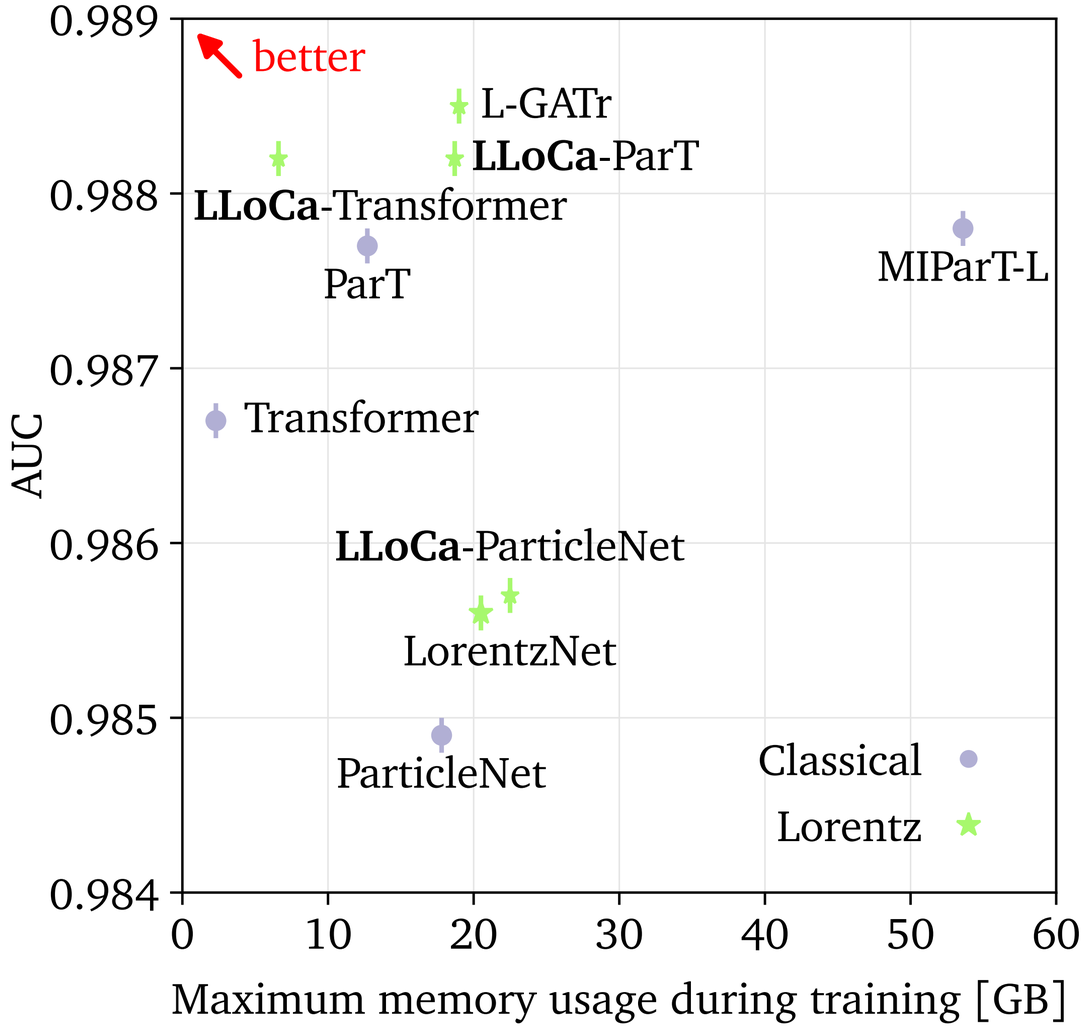
JetClass dataset

Network	Accuracy	AUC	Time	FLOPs	Memory	Parameters
PFN [56]	0.772	0.9714	3h	3M	1.6G	86k
P-CNN [56]	0.809	0.9789	3h	12M	3.0G	354k
MIParT-L [53]	0.861	0.9878	47h	225M	53.6G	2380k
LorentzNet* [11]	0.847	0.9856	57h	676M	20.5G	223k
L-GATr* [15]	0.866	0.9885	180h	2060M	19.0G	1079k
ParticleNet [55]	0.844	0.9849	27h	413M	16.5G	366k
LLoCa-ParticleNet* [22]	0.848	0.9857	41h	517M	23.5G	385k
ParT [21]	0.861	0.9877	38h	211M	13.3G	2141k
LLoCa-ParT* [22]	0.864	0.9882	67h	315M	19.9G	2160k
Transformer [22]	0.855	0.9867	17h	210M	2.3G	1979k
LLoCa-Transformer* [22]	0.864	0.9882	33h	301M	6.9G	1998k

Jet tagging

FLOPs and memory





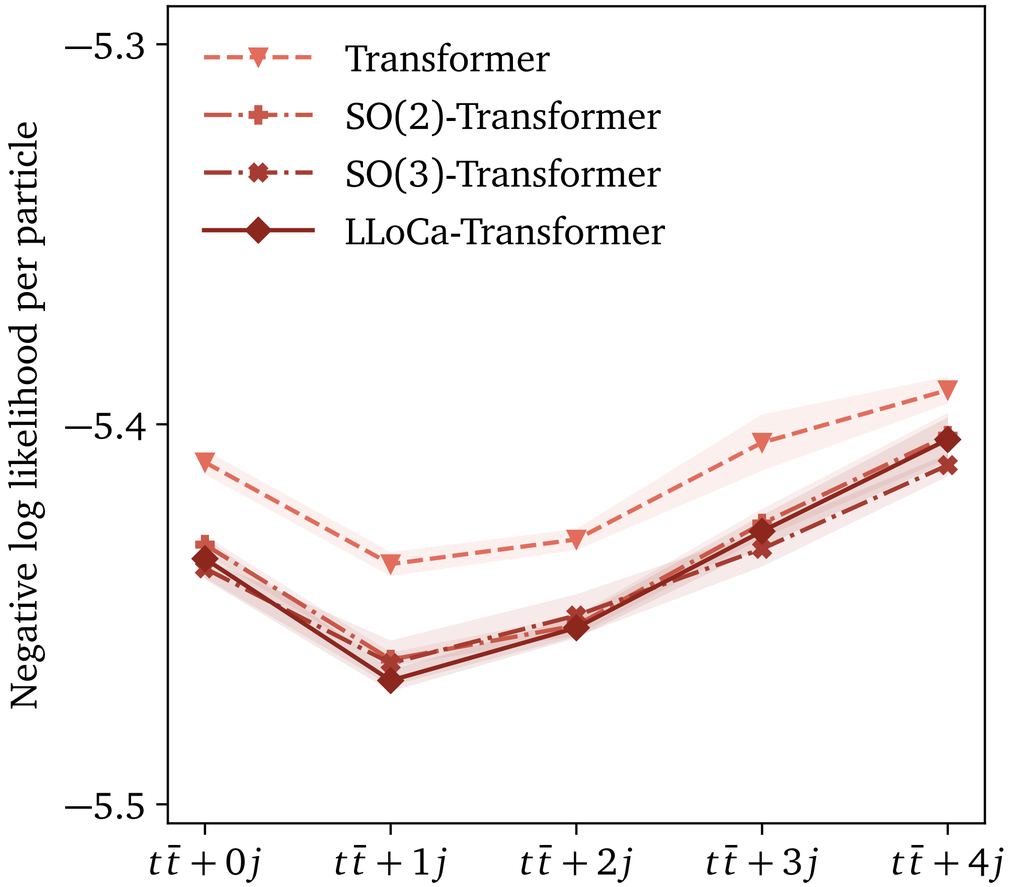
Jet tagging

Effect of internal representations

Network	Accuracy	AUC
Transformer	0.855	0.9867
ParT	0.861	0.9877
LLoCa-Transformer (global can.)	0.861	0.9878
LLoCa-Transformer (16 scalars)	0.851	0.9862
LLoCa-Transformer (4 vectors)	0.863	0.9880
LLoCa-Transformer (8 scalars, 2 vectors)	0.864	0.9882
LLoCa-Transformer (12 scalars, 1 vector)	0.864	0.9882

Event generation

Effect of internal representations



Lorentz-equivariance does not help for event generation