Jet tagging algorithm respecting Lorentz group symmetry

based on: S.Gong *et al.* *JHEP 07 (2022) 030* (*arXiv.2201.08187*);
C.Li *et al.* *arXiv:2208.07814*

Congqiao Li (*Peking University*)

BOOST 2022 · Hamburg
17 August, 2022

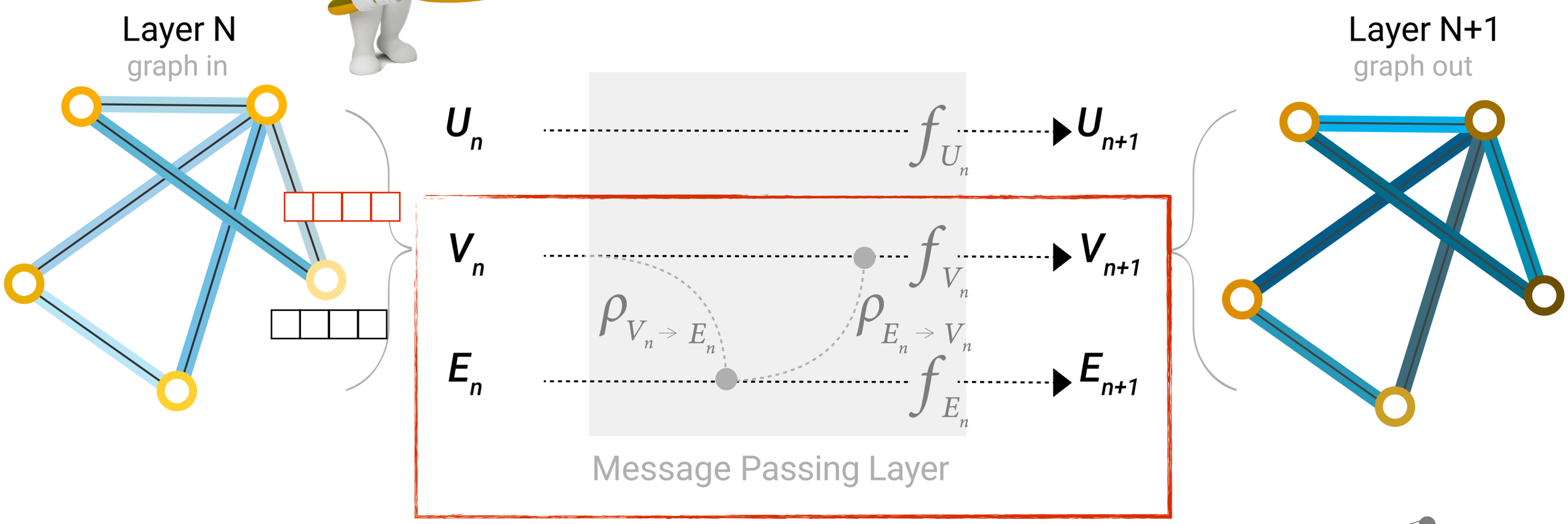
Introduction

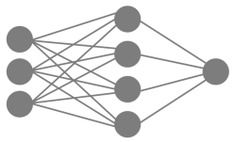
- ➔ ***HEP community is eager to study novel network design***
  - ❖ may directly lead to more accurate physics measurements at the LHC
- ➔ ***Full Lorentz symmetry has been studied for quite some time***
  - ❖ efforts including LoLa, Lorentz Boost Network, Lorentz Group Network, etc.
  - ❖ still cannot demonstrate the potential advantages of such a design
    - ▸ there lacks a fair comparison with the network without symmetric design
  - ❖ can we really see improvements with the use of Lorentz symmetry?
- ➔ ***LorentzNet makes a successful attempt***
  - ❖ made up of GNN + Lorentz symmetry preservation
  - ❖ then, are we able to understand where stands the underlying reason for benefits from LorentzNet?

➔ Let's start our journey in LorenzNet, then try to answer a broader question: which role does Lorentz-symmetry play in jet tagging

Graph neural networks

figure from https://distill.pub/2021/gnn-intro/

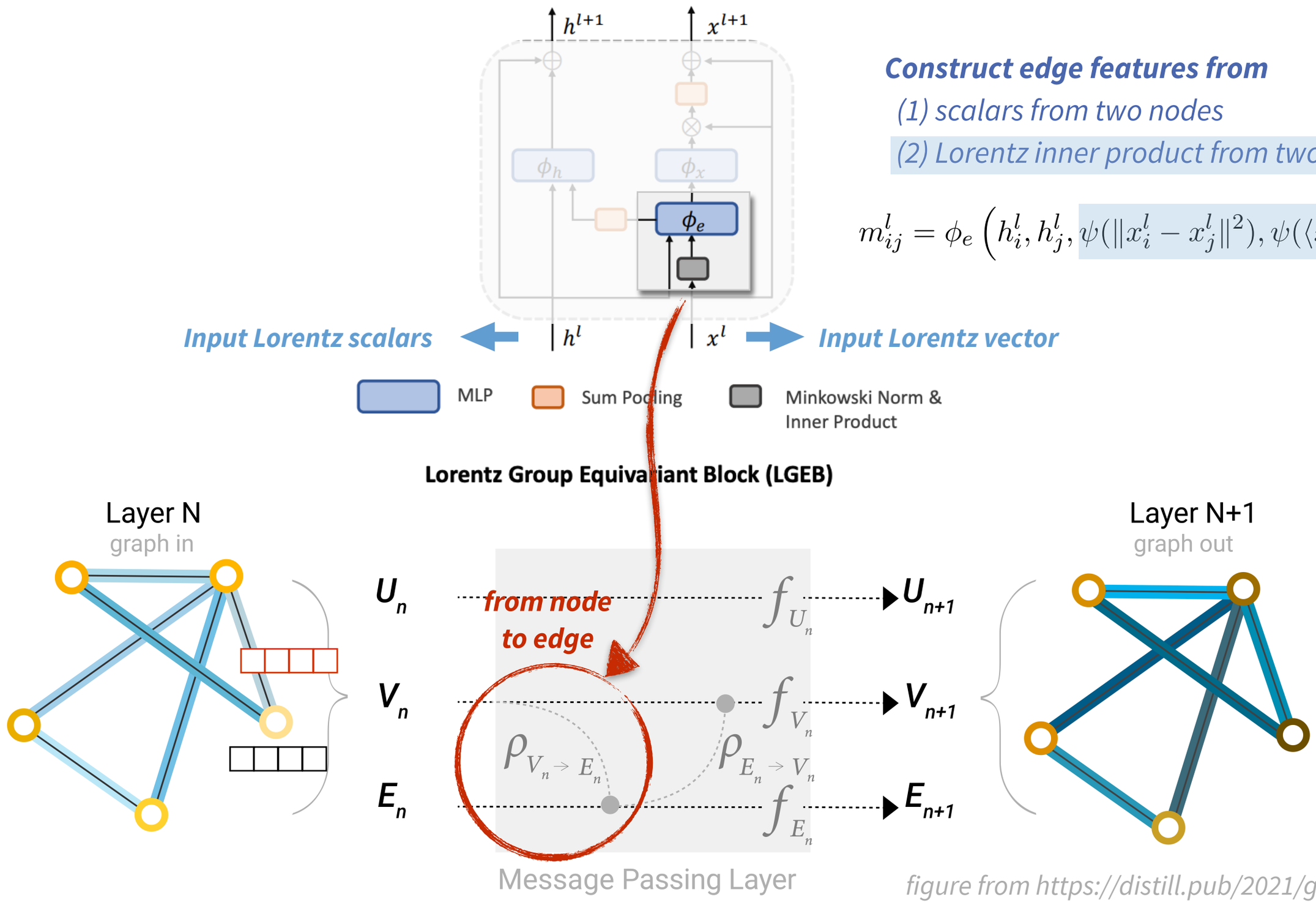


update function $f =$  , …

pooling function ρ

Message passing is an important component of GNN

LorentzNet architecture



Construct edge features from

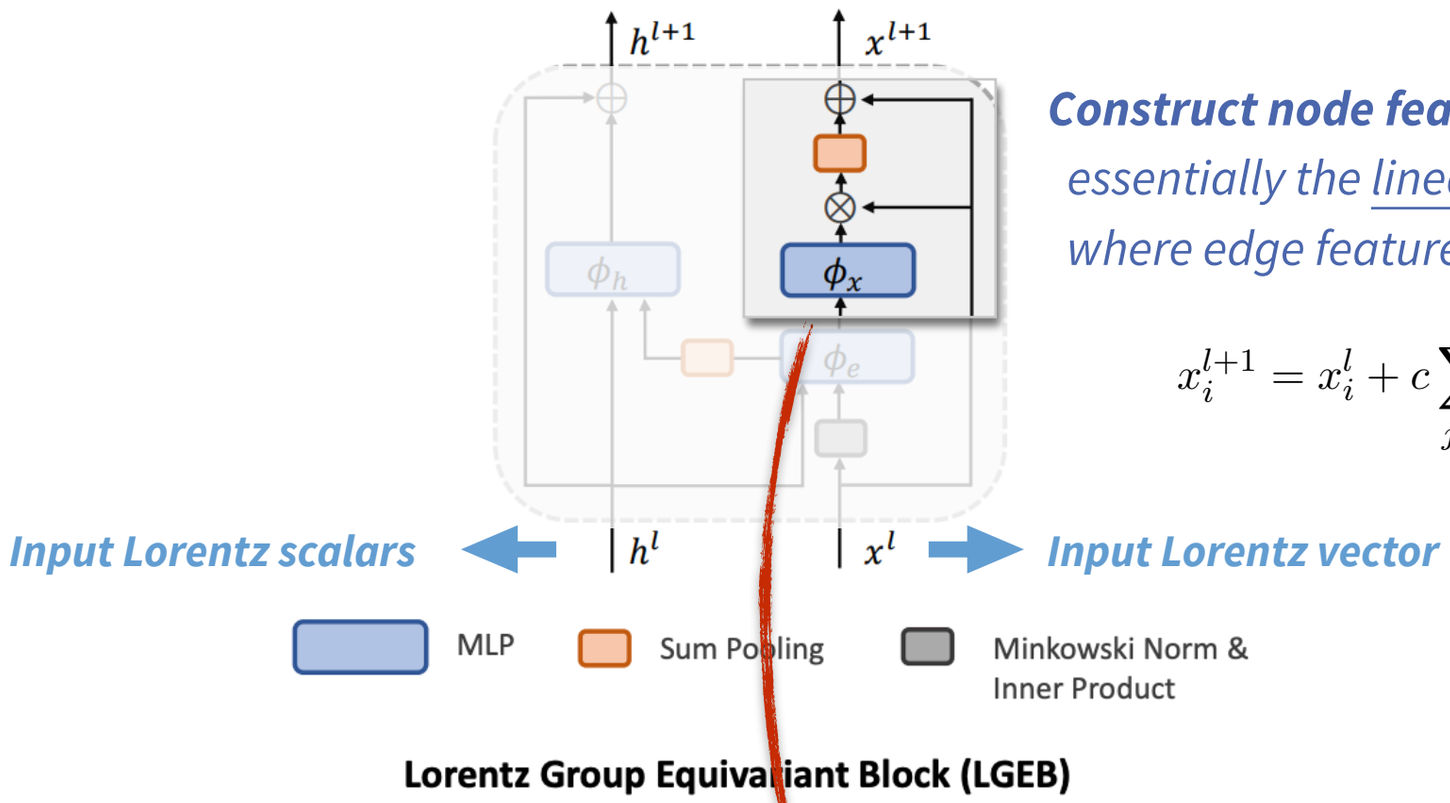
(1) scalars from two nodes

(2) Lorentz inner product from two vectors

$$m_{ij}^l = \phi_e\left(h_i^l, h_j^l, \psi(\|x_i^l - x_j^l\|^2), \psi(\langle x_i^l, x_j^l\rangle)\right)$$

figure from https://distill.pub/2021/gnn-intro/

LorentzNet architecture



Construct node features (vector):

essentially the linear combination of all vectors, where edge features are "weights"

$$x_i^{l+1} = x_i^l + c \sum_{j \neq i} \phi_x(m_{ij}^l) \cdot x_j^l$$

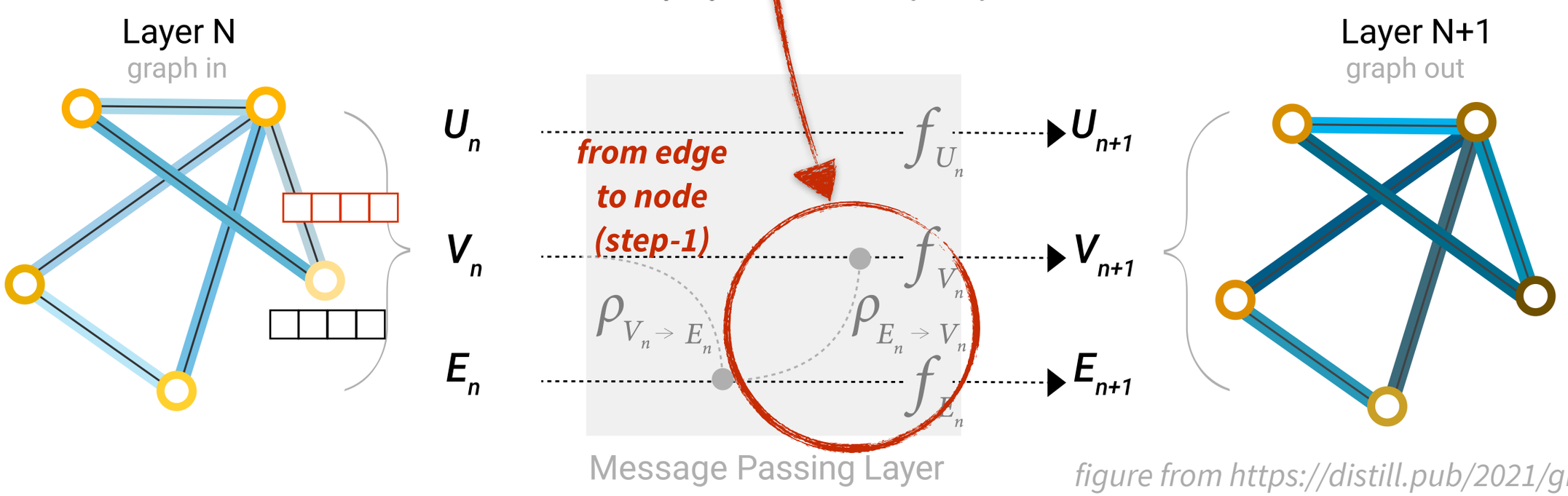


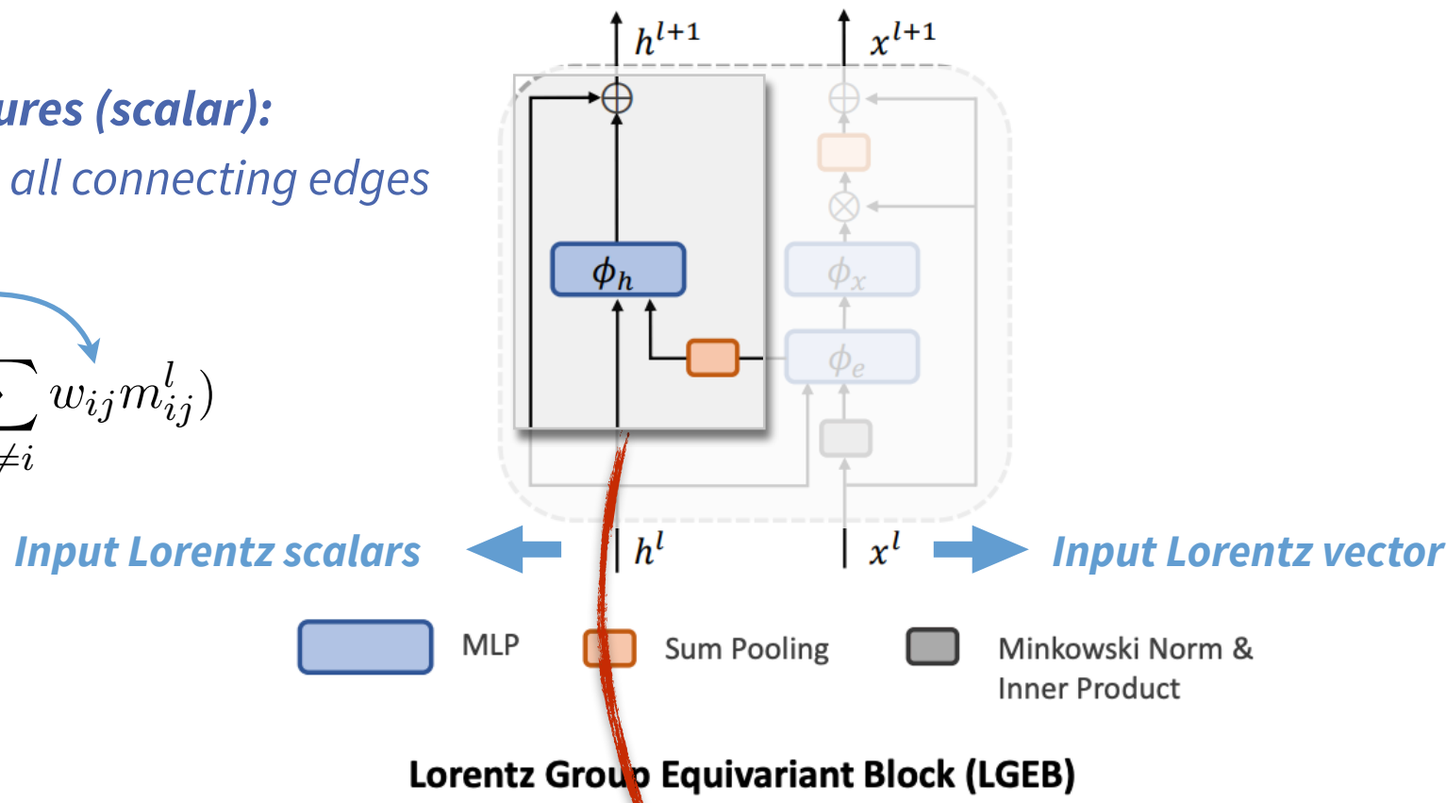
figure from https://distill.pub/2021/gnn-intro/

LorentzNet architecture

Construct node features (scalar):

attentive pooling on all connecting edges

$$h_i^{l+1} = h_i^l + \phi_h(h_i^l, \sum_{j \neq i} w_{ij} m_{ij}^l)$$



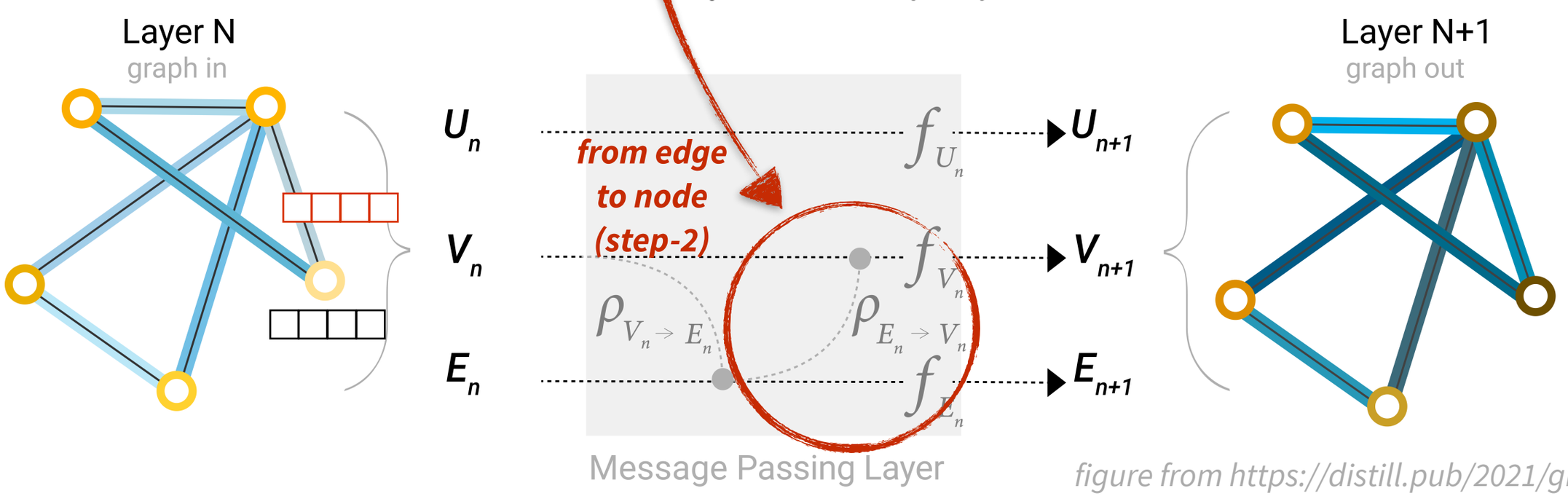
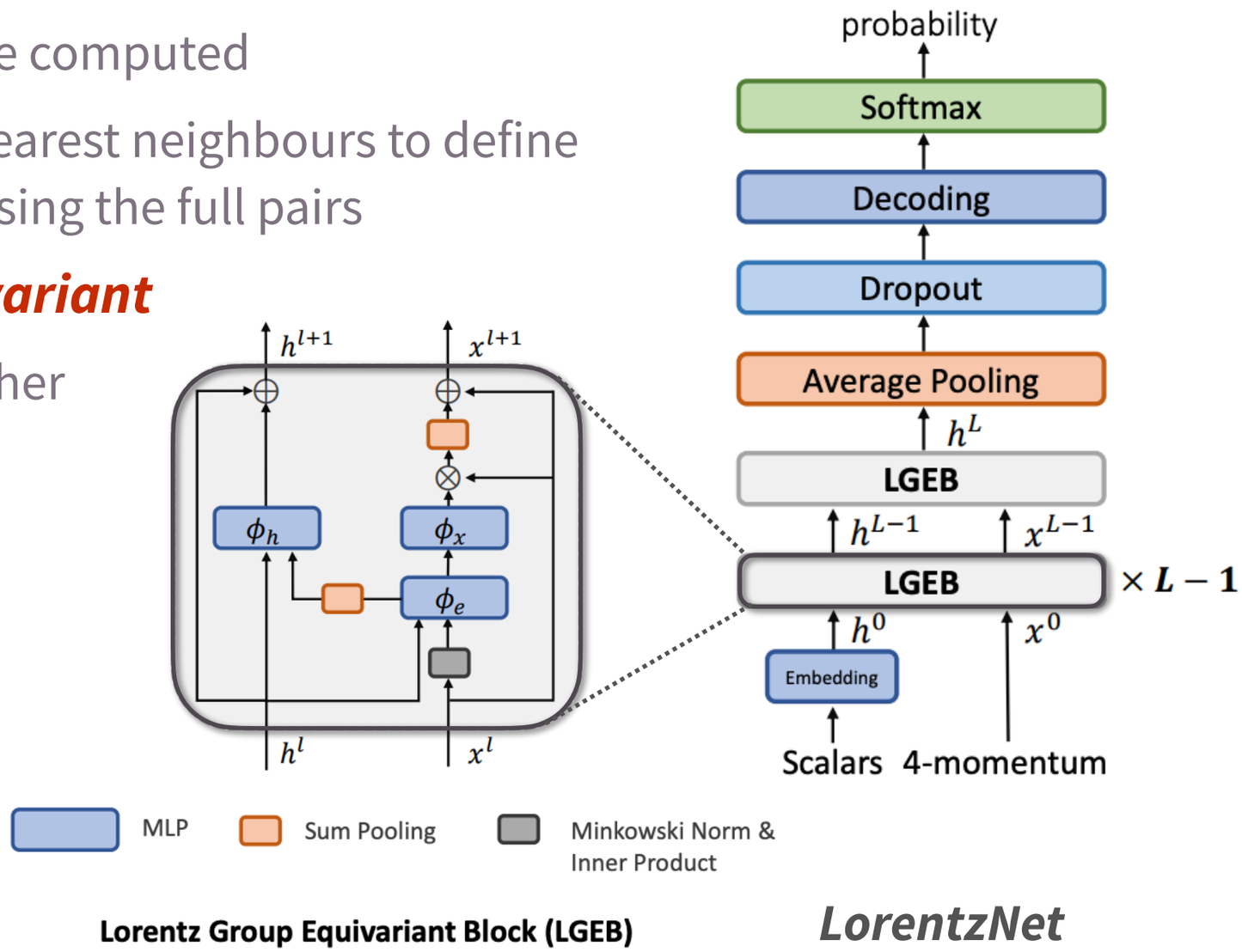


figure from https://distill.pub/2021/gnn-intro/

Summary of architecture

➔ Now let's summarize the main architecture of LorentzNet

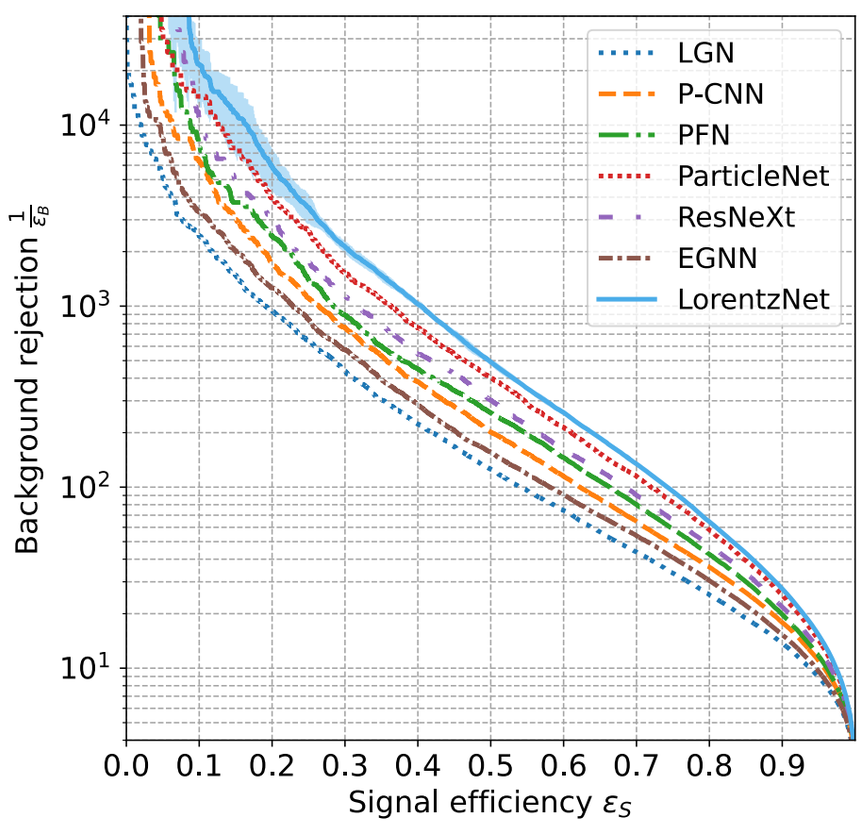
- ***Graph neural network*** as backbone
- ***Fully connected***
  - i.e., all $N(N-1)/2$ edges are computed
  - ParticleNet use dynamic *k*-nearest neighbours to define edges (DGCNN), so it is not using the full pairs
- ***Fully Lorentz invariant/equivariant***
  - nodes can be grouped by either Lorentz scalars or vectors



Performance

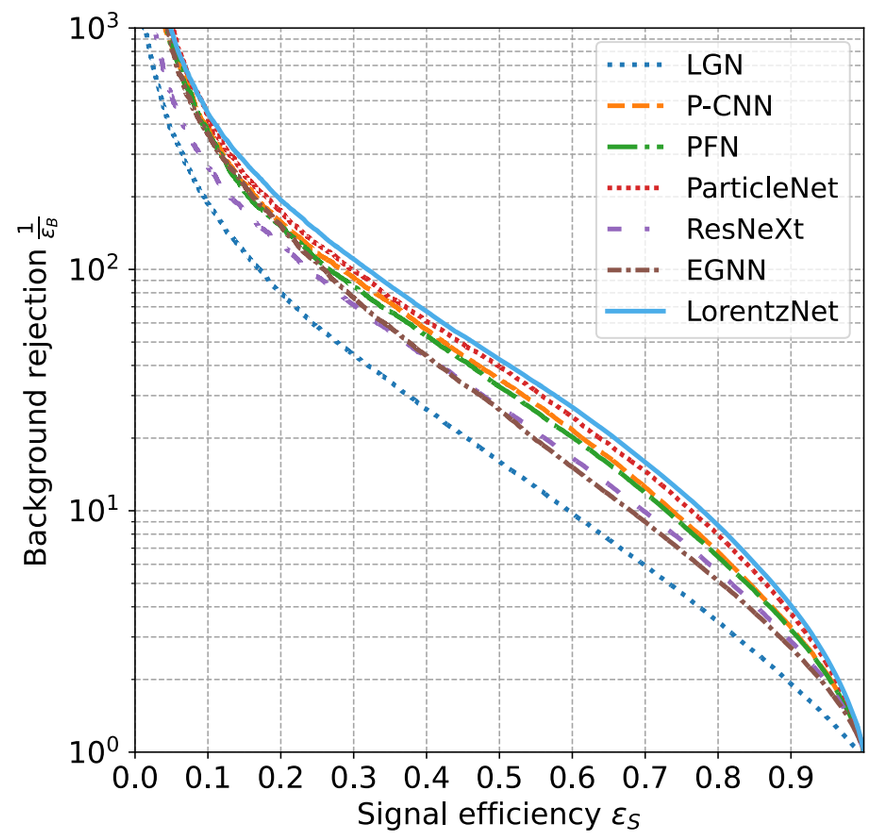
Top tagging benchmark [SciPost Phys. 7 (2019) 014]

Model	Accuracy	AUC	$1/\varepsilon_B$ ($\varepsilon_S = 0.5$)	$1/\varepsilon_B$ ($\varepsilon_S = 0.3$)
ResNeXt	0.936	0.9837	302 ± 5	1147 ± 58
P-CNN	0.930	0.9803	201 ± 4	759 ± 24
PFN	0.932	0.9819	247 ± 3	888 ± 17
ParticleNet	0.940	0.9858	397 ± 7	1615 ± 93
EGNN	0.922	0.9760	148 ± 8	540 ± 49
LGN	0.929	0.9640	124 ± 20	435 ± 95
LorentzNet	**0.942**	**0.9868**	$\mathbf{498 \pm 18}$	$\mathbf{2195 \pm 173}$

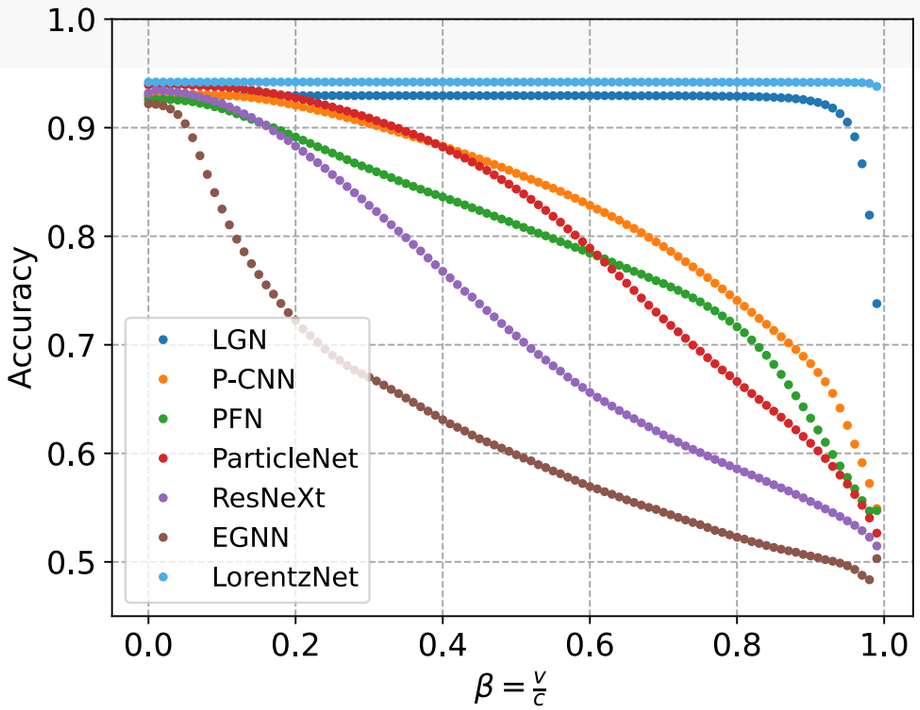


Quark-gluon tagging benchmark [JHEP 01 (2019) 121]

Model	Accuracy	AUC	$1/\varepsilon_B$ ($\varepsilon_S = 0.5$)	$1/\varepsilon_B$ ($\varepsilon_S = 0.3$)
ResNeXt	0.821	0.8960	30.9	80.8
P-CNN	0.827	0.9002	34.7	91.0
PFN	–	0.9005	34.7 ± 0.4	–
ParticleNet	0.840	0.9116	39.8 ± 0.2	98.6 ± 1.3
EGNN	0.803	0.8806	26.3 ± 0.3	76.6 ± 0.5
LGN	0.803	0.8141	8.30	15.2
LorentzNet	**0.844**	**0.9156**	$\mathbf{42.4 \pm 0.4}$	$\mathbf{110.2 \pm 1.3}$



Additional tests



➔ ***Equivariance test***:

- LorentzNet is more robust when the input jet undergoes a Lorentz transformation (consider Lorentz boosts on *x*-axis)

➔ ***Small training sample size***:

- LorentzNet is able to perform much better when trained on a smaller size of sample

Training Fraction	Model	Accuracy	AUC	$1/\varepsilon_B$ ($\varepsilon_S = 0.5$)	$1/\varepsilon_B$ ($\varepsilon_S = 0.3$)
0.5%	ParticleNet	0.913	0.9687	77 ± 4	199 ± 14
	LorentzNet	**0.929**	**0.9793**	$\mathbf{176 \pm 14}$	$\mathbf{562 \pm 72}$
1%	ParticleNet	0.919	0.9734	103 ± 5	287 ± 19
	LorentzNet	**0.932**	**0.9812**	$\mathbf{209 \pm 5}$	$\mathbf{697 \pm 58}$
5%	ParticleNet	0.931	0.9807	195 ± 4	609 ± 35
	LorentzNet	**0.937**	**0.9839**	$\mathbf{293 \pm 12}$	$\mathbf{1108 \pm 84}$

➔ ***Ablation study on Lorentz equivariant preserving structure***

- replacing the pairwise scalar (mass) has a negative effect on the network

Model	Equivariance	Accuracy	AUC	$1/\varepsilon_B$ ($\varepsilon_S = 0.5$)	$1/\varepsilon_B$ ($\varepsilon_S = 0.3$)
LorentzNet (w/o)	✗	0.934	0.9832	290 ± 30	1105 ± 59
LorentzNet	✓	**0.942**	**0.9868**	$\mathbf{498 \pm 18}$	$\mathbf{2195 \pm 173}$

Conclusion

S.Gong *et al.* *JHEP 07 (2022) 030*

➔ We present LorentzNet, a Lorentz group equivariant GNN

- ❖ the network has now reached state-of-the-art performance, when trained and evaluated on two mainstream benchmarks
- ❖ its equivariance property confirmed on Lorentz-transformed test dataset
- ❖ ablation study shows Lorentz-symmetry-preserving mechanism does help the network

- END OF SLIDES -

Conclusion (?)

S.Gong *et al.* *JHEP 07 (2022) 030*

➔ We present LorentzNet, a Lorentz group equivariant GNN

- ❖ the network has now reached state-of-the-art performance, when trained and evaluated on two mainstream benchmarks
- ❖ its equivariance property confirmed on Lorentz-transformed test dataset
- ❖ ablation study shows Lorentz-symmetry-preserving mechanism does help the network

- END OF SLIDES -

➔ Wait.. **there are still questions to ask/solve**

- ❖ many novel network designs have been proposed and evaluated on these two benchmarks, but see no significant improvement - why?
  - why LorentzNet behave surprising well on them?
- ❖ can we dig deeper to extract the key component in LorentzNet? can it be applied to other networks as well?

➔ Let's continue our game 🎮

LorentzNet performance on JetClass

NEW

JetClass *[arXiv:2202.03772, proceedings of 39th ICML, Vol.162]*

	All classes		$H \to b\bar{b}$	$H \to c\bar{c}$	$H \to gg$	$H \to 4q$	$H \to \ell\nu qq'$	$t \to bqq'$	$t \to b\ell\nu$	$W \to qq'$	$Z \to q\bar{q}$
	Accuracy	AUC	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{99\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{99.5\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$
PFN	0.772	0.9714	2924	841	75	198	265	797	721	189	159
P-CNN	0.809	0.9789	4890	1276	88	474	947	2907	2304	241	204
ParticleNet	0.844	0.9849	7634	2475	104	954	3339	10526	11173	347	283
NEW LorentzNet	0.855	0.9869	9217	3425	117	1550	4425	19802	12500	480	353
ParT	**0.861**	**0.9877**	**10638**	**4149**	**123**	**1864**	**5479**	**32787**	**15873**	**543**	**402**
ParT (plain)	0.849	0.9859	9569	2911	112	1185	3868	17699	12987	384	311

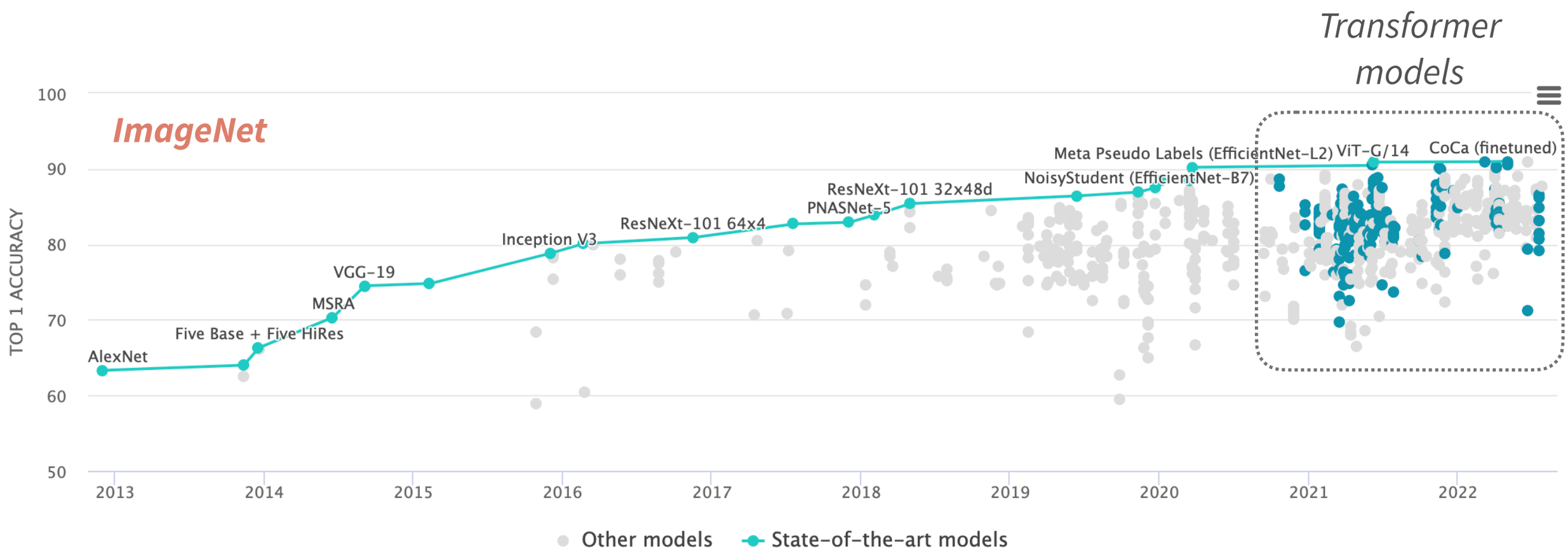
- ➔ LorentzNet performs much better than ParticleNet, slightly worse than ParT
  - note that for #params LorentzNet < ParticleNet, and << ParT
  - can prove that **LorentzNet is still more performant**
- ➔ We may want to understand why there is performance difference between *top tagging dataset* and *JetClass*

	Accuracy	# params	FLOPs
PFN	0.772	86.1 k	4.62 M
P-CNN	0.809	354 k	15.5 M
ParticleNet	0.844	370 k	540 M
LorentzNet	0.855	233 k	2.01 G
ParT	**0.861**	2.14 M	340 M
ParT (plain)	0.849	2.13 M	260 M

Model, data size and "inductive bias"

➔ Lessons from image classification from Computer Vision

- Training on ImageNet and its extension (224x224 pixel image classification)
  - Transformer models have led the performance, since the first application in 2020

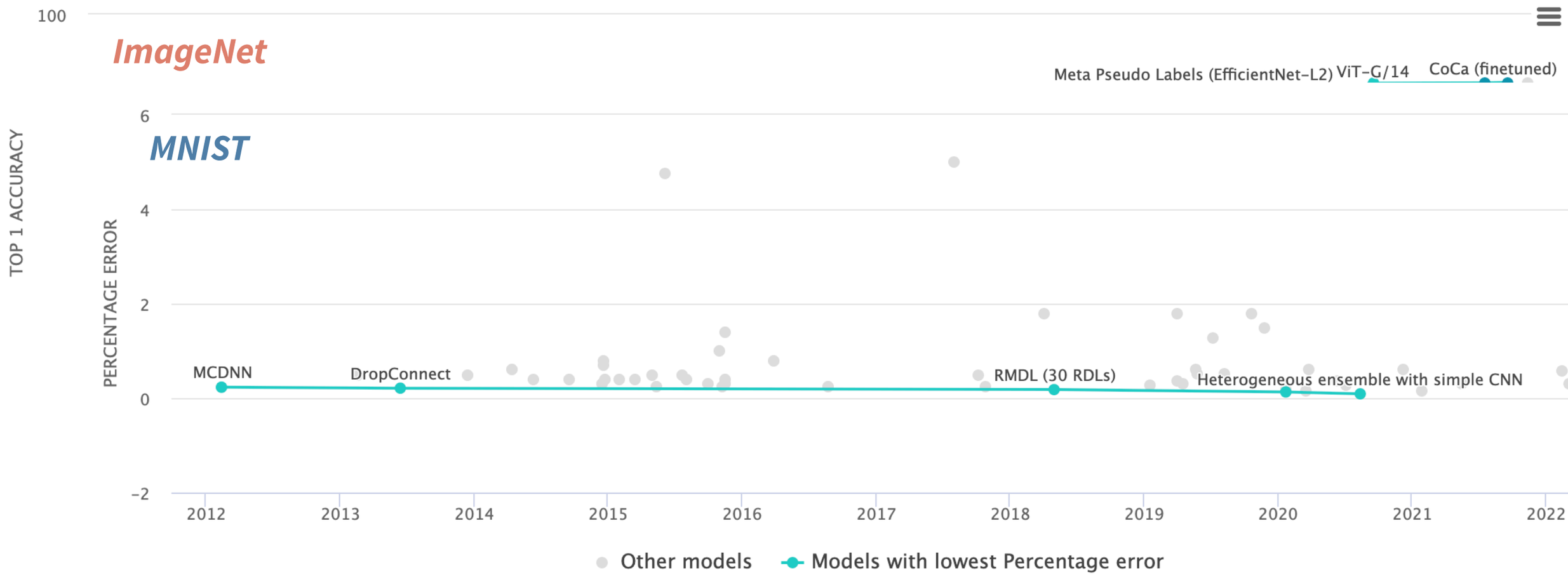


https://paperswithcode.com/sota/image-classification-on-imagenet

Model, data size and "inductive bias"

➔ Lessons from image classification from Computer Vision

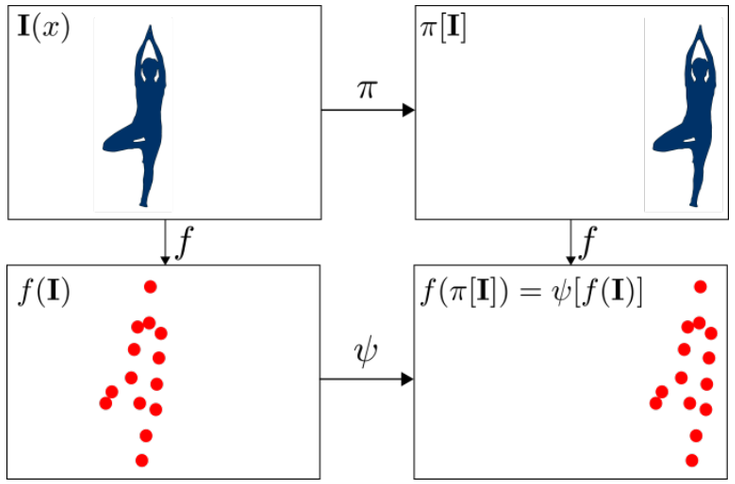
- Training on ImageNet and its extension (224x224 pixel image classification)
  - Transformer models have led the performance, since the first application in 2020
- But if we look back to MNIST dataset (hand-written digit classification)
  - still CNN-based networks rank higher



Model, data size and "inductive bias"

➔ Lessons from image classification from Computer Vision

- Training on ImageNet and its extension (224x224 pixel image classification)
  - Transformer models have led the performance, since the first application in 2020
- But if we look back to MNIST dataset (hand-written digit classification)
  - still CNN-based networks rank higher
- ***Possible explanations***:
  - for MNIST dataset, we want more "efficient" model when training on small dataset
  - to be more efficient, cooperating with "***inductive bias***" in the network design is crucial
  - CNN respects the local translational symmetry, which is an inductive bias when processing real-world images



Model, data size and "inductive bias"

➔ Lessons from image classification from Computer Vision

- Training on ImageNet and its extension (224x224 pixel image classification)
  - Transformer models have led the performance, since the first application in 2020
- But if we look back to MNIST dataset (hand-written digit classification)
  - still CNN-based networks rank higher
- ***Possible explanations***:
  - for MNIST dataset, we want more "efficient" model when training on small dataset
  - to be more efficient, cooperating with "***inductive bias***" in the network design is cr
  - C s when p

- We think it is the same situation for the top tagging dataset
  - the discrimination between top vs. QCD jets is an easier task, and performance are more saturated
  - it is the **"more efficient model"** that tends to win the top tagging competition
- Now we need to verify if LorentzNet is using some inductive bias to be efficient

Interpret Lorentz-symmetry as an inductive bias

- ➔ Goal:
  - ❖ we want to verify that Lorentz-symmetry preservation is a key component
  - ❖ even better if we isolate “a patch” from LorentzNet, which can be applied to wider range of networks
- ➔ Our experiments
  - ❖ devise multiple choices of pairwise features, which are invariant to some or all Lorentz transformations
  - ❖ want to see if this affects network performance as we expect

C.Li *et al.* *arXiv:2208.07814*

(following slides present a part of this work)

Interpret Lorentz-symmetry as an inductive bias

- ➔ Goal:
  - ❖ we want to verify that Lorentz-symmetry preservation is a key component
  - ❖ even better if we isolate "a patch" from LorentzNet, which can be applied to wider range of networks
- ➔ Our experiments
  - ❖ devise multiple choices of pairwise features, which are **invariant to some or all Lorentz transformations**
  - ❖ want to see if this affects network performance as we expect

C.Li *et al.* *arXiv:2208.07814*

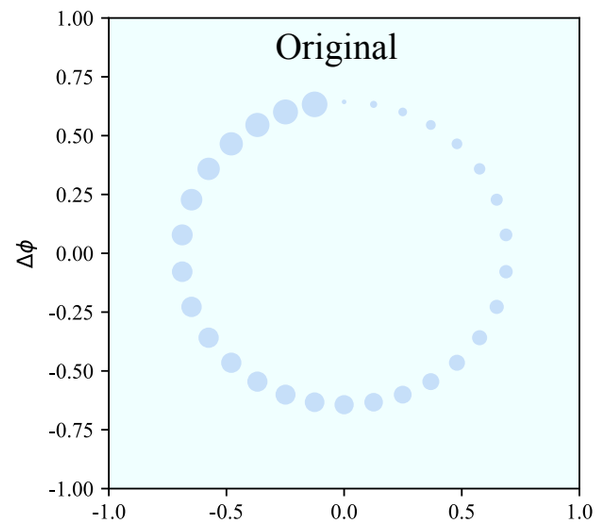
(following slides present a part of this work)

First of all, we need to find a good way to categorize the possible Lorentz transformations acted on the jet

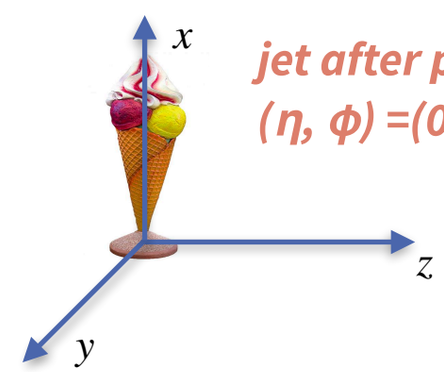
Lorentz transformations

➔ By HEP convention, jets are represented on η-ϕ plane w.r.t. its mean axis

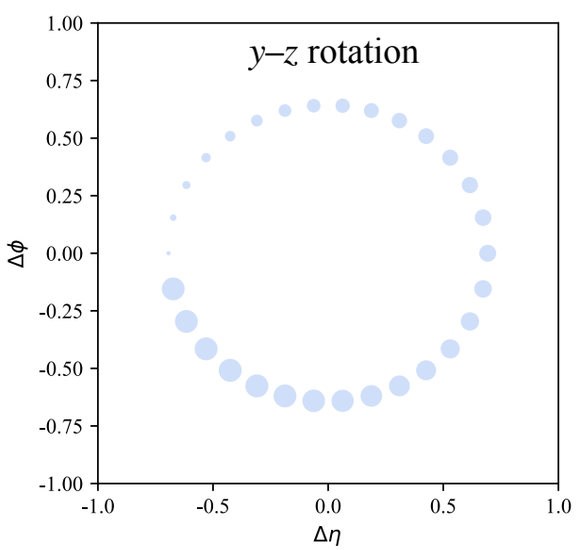
- equivalent as: boosts on *z*-axis → rotation on *x*-*y* plane (transverse plane) → jet points to the *x*-axis
- then we have ***four additional DoFs*** for Lorentz transformation!



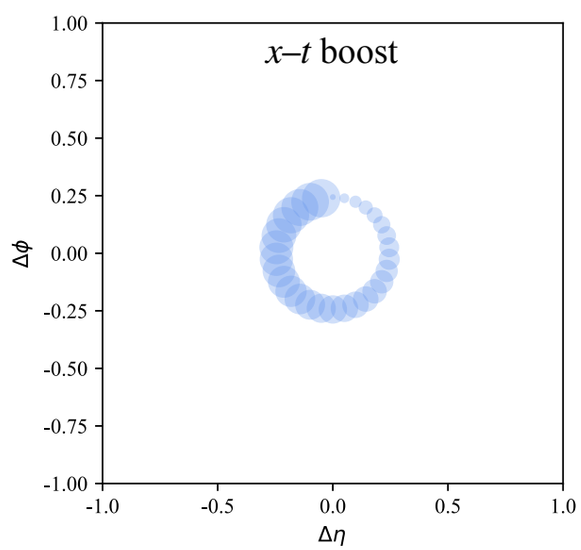
≈ Viewing the jet from x-direction

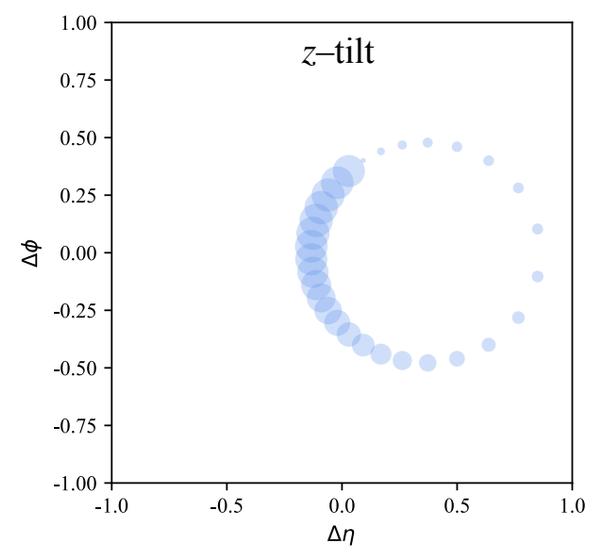


jet after preprocessing:
(η, ϕ) =(0, 0)

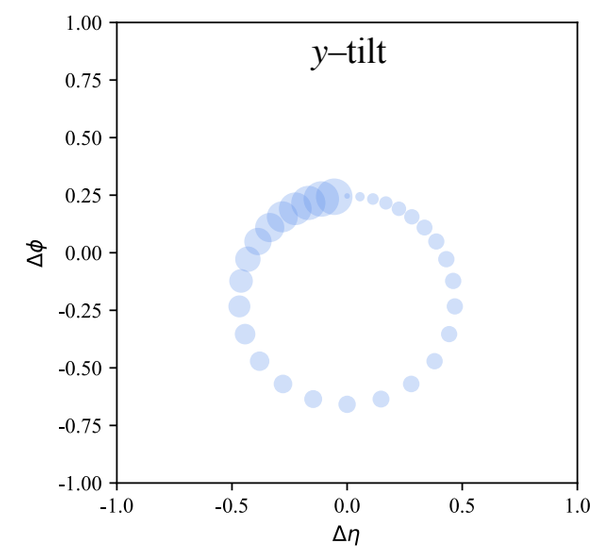


≈ *η*-*ϕ* rotation





z-boost + x-z rotation



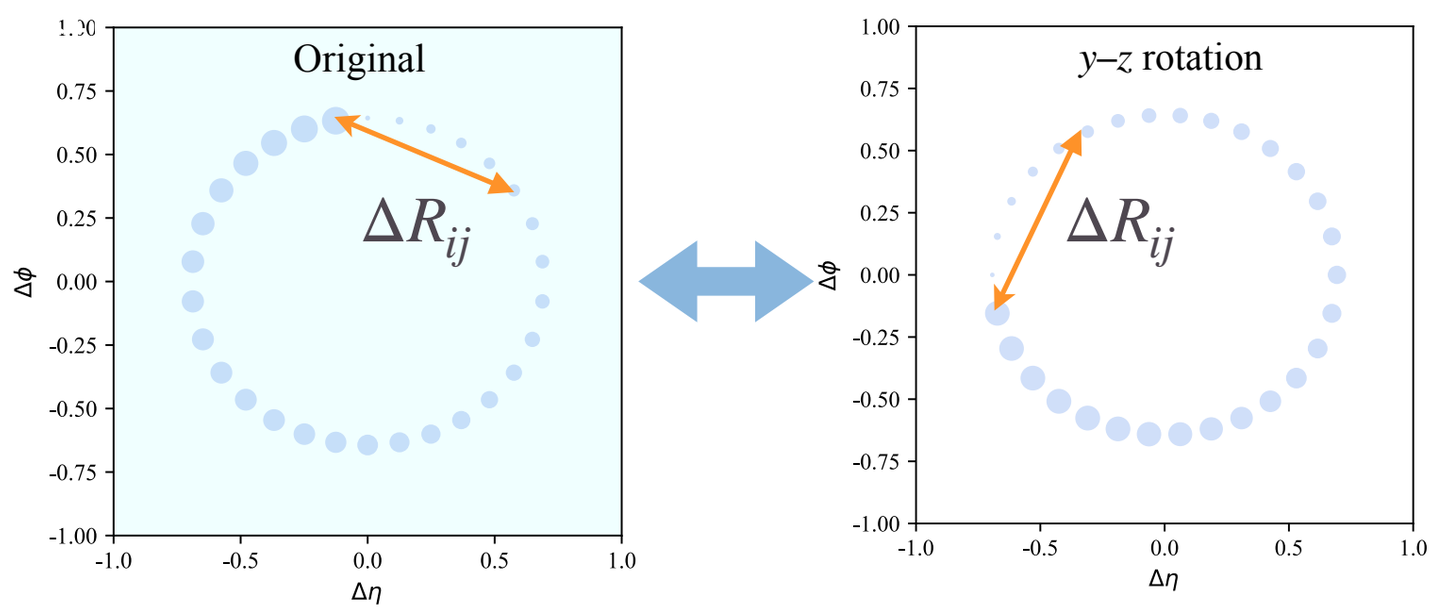
y-boost + *x*-*y* rotation

Variable design

Pairwise variable	z–t boost	x–y rotation	y–z rotation ($y_{y,z} \sim o(1)$)	x–t boost ($y_{y,z} \sim o(1)$)	z-tilt ($y_{y,z} \sim o(1)$)	y-tilt ($y_{y,z} \sim o(1)$)
m_{ij}^2	✓	✓	✓	✓	✓	✓
ΔR_{ij}	✓	✓	✓			
$\Delta R_{ij}(p_{\mathrm{T},i} + p_{\mathrm{T},j})$	✓	✓	✓	✓		
E_{ij} (ablation study)		✓	✓			

➔ Devise variables which are invariant under ***some or all Lorentz (sub)symmetries***

- ❖ pairwise mass: invariant under all transformations
- ❖ pairwise ΔR_{ij}: approx. invariant under y-z rotation (≈ η-ϕ rotation)
- ❖

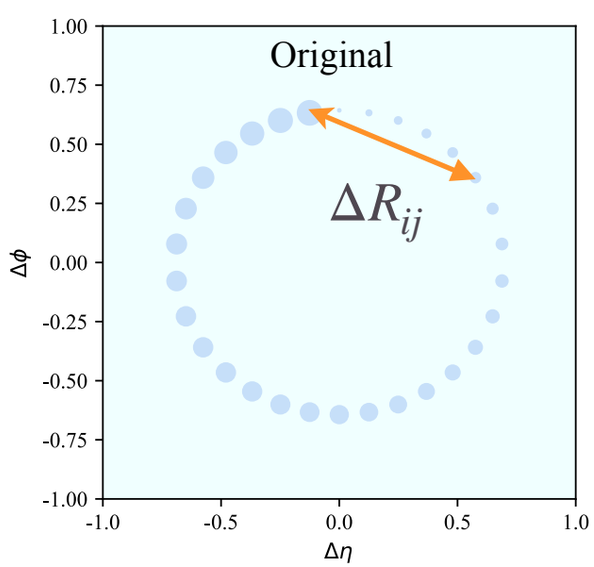


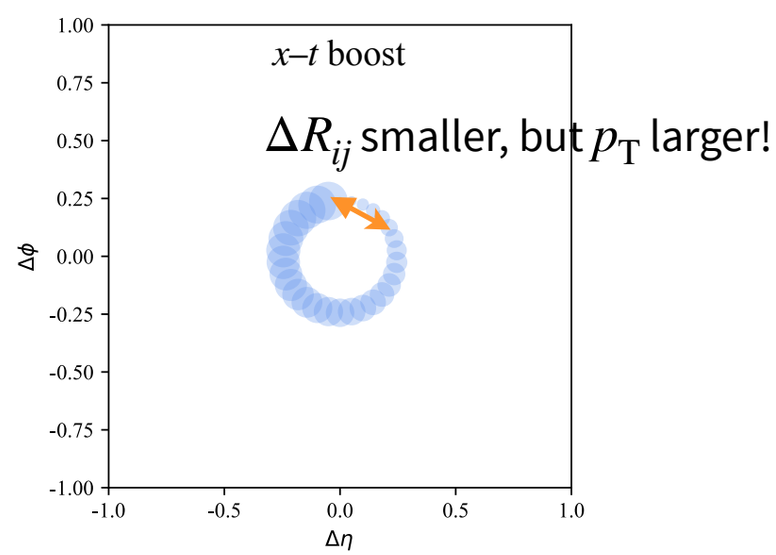
Variable design

Pairwise variable	z–t boost	x–y rotation	y–z rotation ($y_{y,z} \sim o(1)$)	x–t boost ($y_{y,z} \sim o(1)$)	z-tilt ($y_{y,z} \sim o(1)$)	y-tilt ($y_{y,z} \sim o(1)$)
m_{ij}^2	✓	✓	✓	✓	✓	✓
ΔR_{ij}	✓	✓	✓			
$\Delta R_{ij}(p_{\mathrm{T},i} + p_{\mathrm{T},j})$	✓	✓	✓	✓		
E_{ij} (ablation study)		✓	✓			

➔ Devise variables which are invariant under ***some or all Lorentz (sub)symmetries***

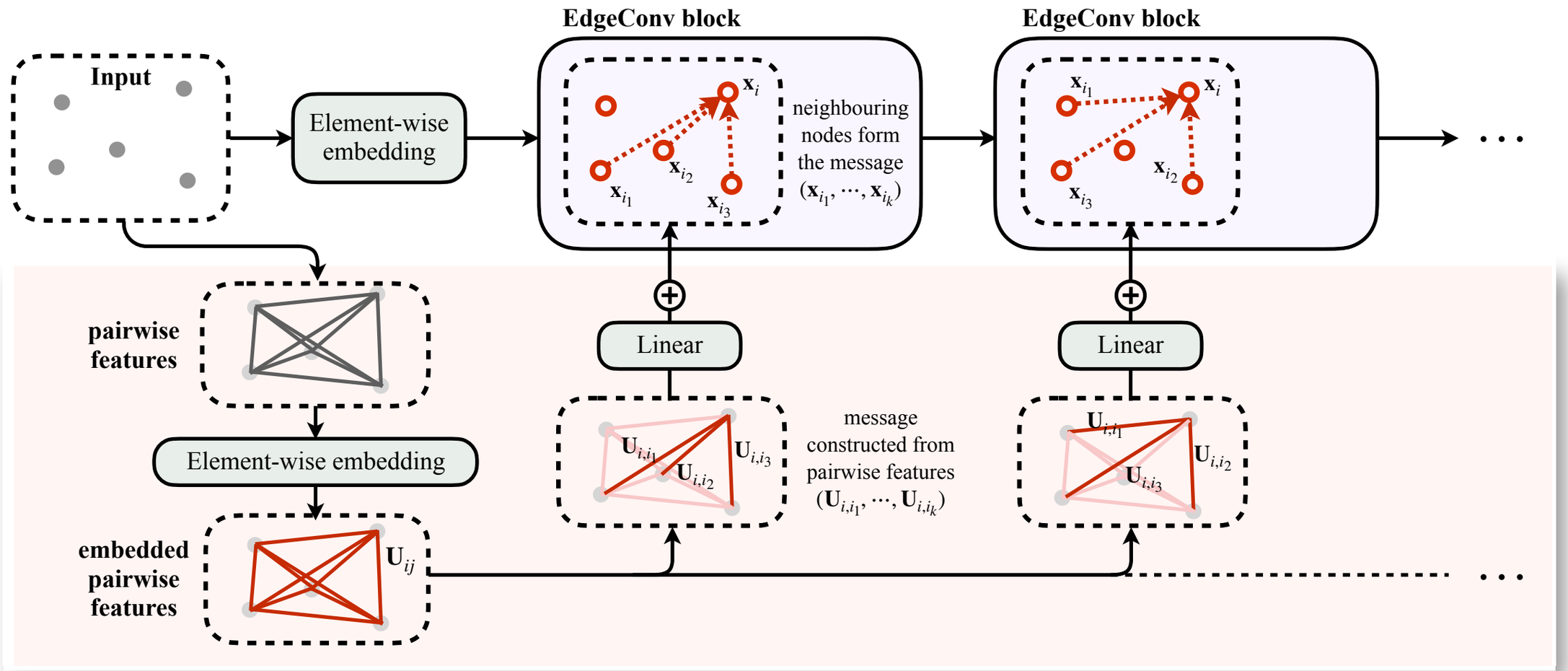
- pairwise mass: invariant under all transformations
- pairwise ΔR_{ij}: approx. invariant under y-z rotation (≈ η-ϕ rotation)
- manually construct variable $\Delta R_{ij}(p_{\mathrm{T},i} + p_{\mathrm{T},j})$: can prove that it is also approx. invariant under x-boost!





Experiments on ParticleNet and LorentzNet

from paper arXiv:2208.07814



➔ Two baseline networks to study pairwise feature effect: ParticleNet & LorentzNet$_{base}$

- ***ParticleNet:*** now add an additional patch (in red colour) to incorporate pairwise features, based on ParticleNet's intrinsic kNN pairs
- ***LorentzNet$_{base}$***: LorentzNet has already included "pairwise mass": remove it to create our baseline (but complete all node features as the case of ParticleNet)

Performance for adding pairwise features

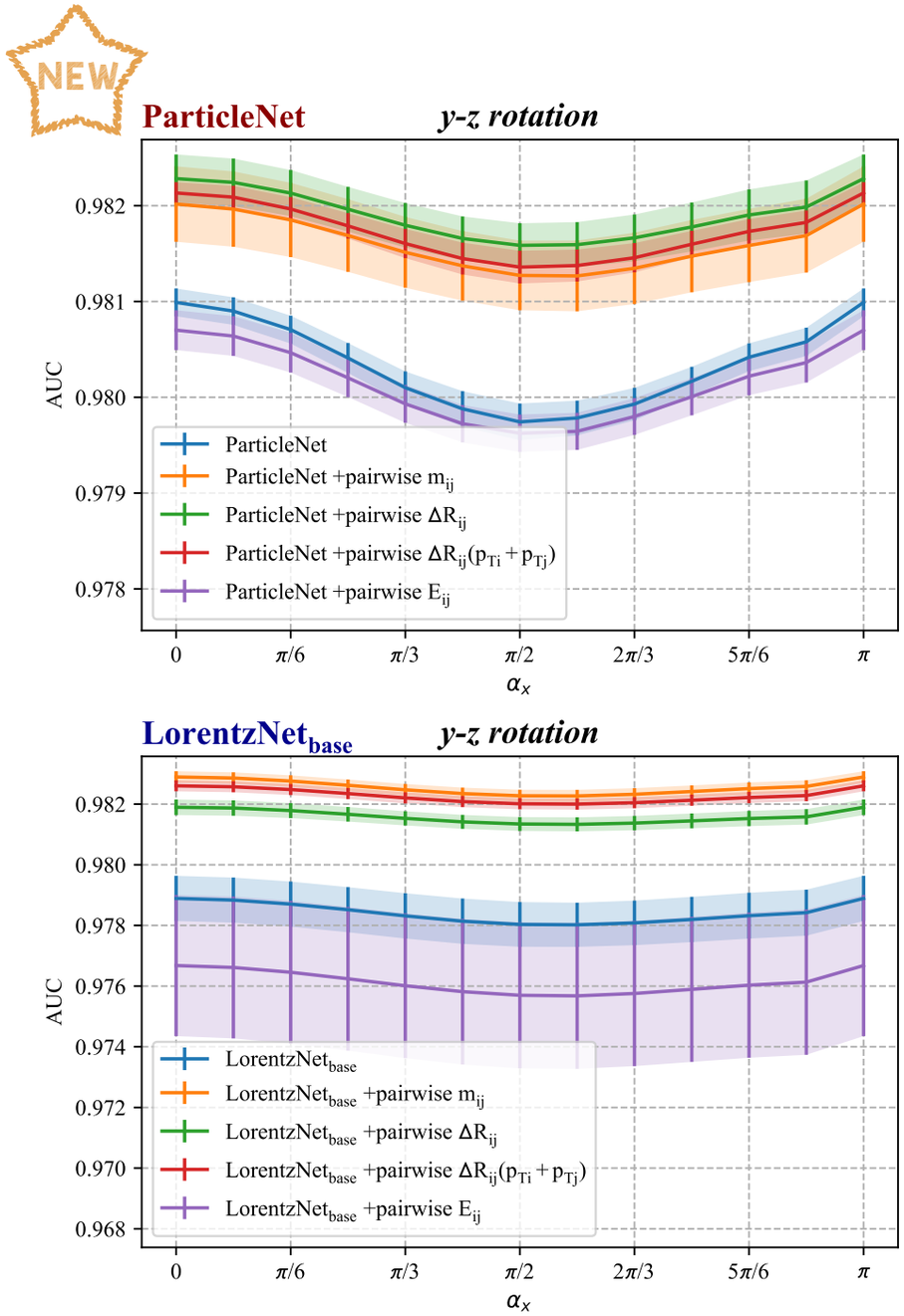
Training on 60k top tagging dataset (smaller dataset manifest the power of inductive bias)

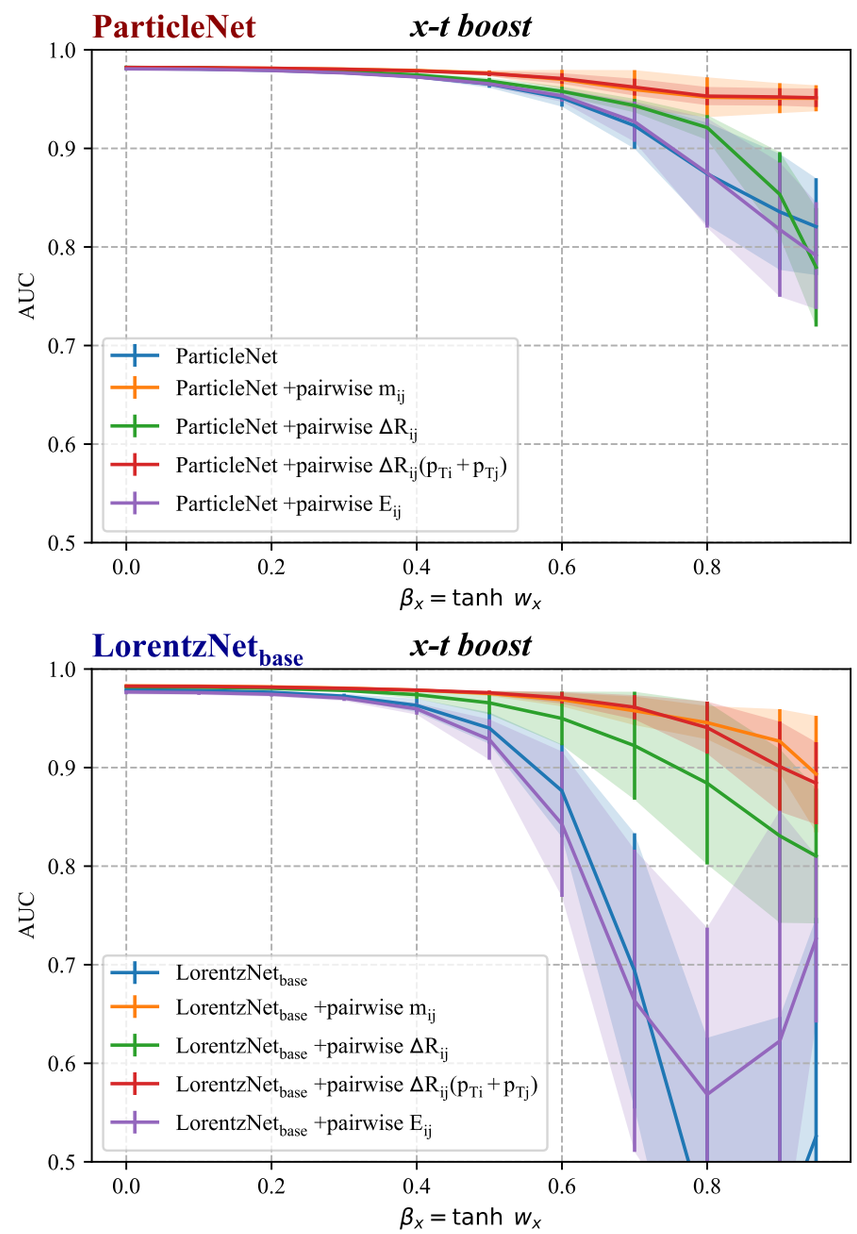
NEW

Base model	Variation	Accuracy	AUC	$1/\epsilon_B$ ($\epsilon_S = 50\%$)	$1/\epsilon_B$ ($\epsilon_S = 30\%$)
ParticleNet	—	0.9310(3)	0.9810(2)	198 ± 7	640 ± 29
	+pairwise: m_{ij}	0.9334(8)	0.9820(4)	222 ± 13	722 ± 52
	+pairwise: ΔR_{ij}	0.9334(6)	**0.9823(3)**	$\mathbf{231 \pm 10}$	$\mathbf{752 \pm 43}$
	+pairwise: $\Delta R_{ij}(p_{T,i} + p_{T,j})$	**0.9337(3)**	0.9821(1)	223 ± 6	741 ± 36
	+pairwise: E_{ij}	0.9303(5)	0.9807(2)	200 ± 6	651 ± 23
$\text{LorentzNet}_{\text{base}}$	—	0.9276(12)	0.9789(7)	172 ± 13	581 ± 53
	+pairwise: m_{ij}	**0.9347(4)**	**0.9829(2)**	$\mathbf{260 \pm 6}$	$\mathbf{931 \pm 50}$
	+pairwise: ΔR_{ij}	0.9328(4)	0.9819(3)	232 ± 10	807 ± 35
	+pairwise: $\Delta R_{ij}(p_{T,i} + p_{T,j})$	0.9342(4)	0.9826(2)	251 ± 6	919 ± 34
	+pairwise: E_{ij}	0.9243(37)	0.9767(23)	144 ± 29	485 ± 108

better compared to baselines

Performance for adding pairwise features





- ***Injecting ΔR to the network*** → *more robust to y-z rotation*
- ***Injecting ΔR(p_{Ti}+p_{Tj}) or mass*** → *more robust to y-z rotation and now also the x-boost*

Performance for adding pairwise features

➔ What does it mean?

- the full network tends to be more robust, even when we ***introduce a very small patch structure*** invariant under a certain symmetry (the original network is unaffected)
- ***it means the invariance property of the small sub-network has a big impact on the learning, and can be reflected in the entire network***

Base model	Variation	# parameters	FLOPs
PFN	—	83.84 k	4.46 M
	+node-wise	+26.19 k	+3.41 M
ParticleNet	—	366.16 k	535.73 M
	+pairwise	+34.91 k	+285.29 M
	+node-wise	+21.97 k	+2.83 M
$\text{LorentzNet}_{\text{base}}$	—	226.23 k	1997.69 M
	+pairwise	+0.43 k	+7.02 M
	+node-wise	+37.35 k	+4.8 M

Performance for adding pairwise features

➔ What does it mean?

- the full network tends to be more robust, even when we ***introduce a very small patch structure*** invariant under a certain symmetry (the original network is unaffected)
- ***it means the invariance property of the small sub-network has a big impact on the learning, and can be reflected in the entire network***

Base model	Variation	# parameters	FLOPs
PFN	—	83.84 k	4.46 M
	+node-wise	+26.19 k	+3.41 M
ParticleNet	—	366.16 k	535.73 M
	+pairwise	+34.91 k	+285.29 M
	+node-wise	+21.97 k	+2.83 M
$\text{LorentzNet}_{\text{base}}$	—	226.23 k	1997.69 M
	+pairwise	+0.43 k	+7.02 M
	+node-wise	+37.35 k	+4.8 M

- This experiment shows that "**pairwise mass**" is the key component
- **We reveal that the underlying logic lies in the Lorentz symmetry preservation!**
- Besides, we device other ways to add variables which respect Lorentz symmetries to boost the network performance → see more experiments in *arXiv:2208.07814*

Conclusion

➔ We present LorentzNet, a Lorentz group equivariant GNN

- ❖ the network has now reached state-of-the-art performance, when trained and evaluated on two mainstream benchmarks

additional take-home message

- **Key component for success of LorentzNet?**
  - fully-connected GNN as the backbone
  - Lorentz-symmetry-preserving design to respect the inductive bias of jet data
- **Possible hints to other work?**
  - the "pairwise mass" is one crucial component of the network—reason lies in the symmetry preservation
  - it can be used as a patch for many networks - in fact, *ParticleNeXt*, *LorentzNet*, *ParT* have implemented it in design
- ***Let's enjoy our new understanding of jet tagging in the DL era, and embrace the new improvement ahead***

Backup