# Jet tagging algorithm respecting Lorentz group symmetry

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

**Congqiao Li (李聪乔)**, *Peking University*

*in collaboration with*

**Shiqi Gong** (*AMSS, CAS; MSRA*), **Qi Meng**, **Jue Zhang**, **Tie-Yan Liu** (*MSRA*)
**Huilin Qu** (*CERN*), **Sitian Qian**, **Qiang Li** (*PKU*)

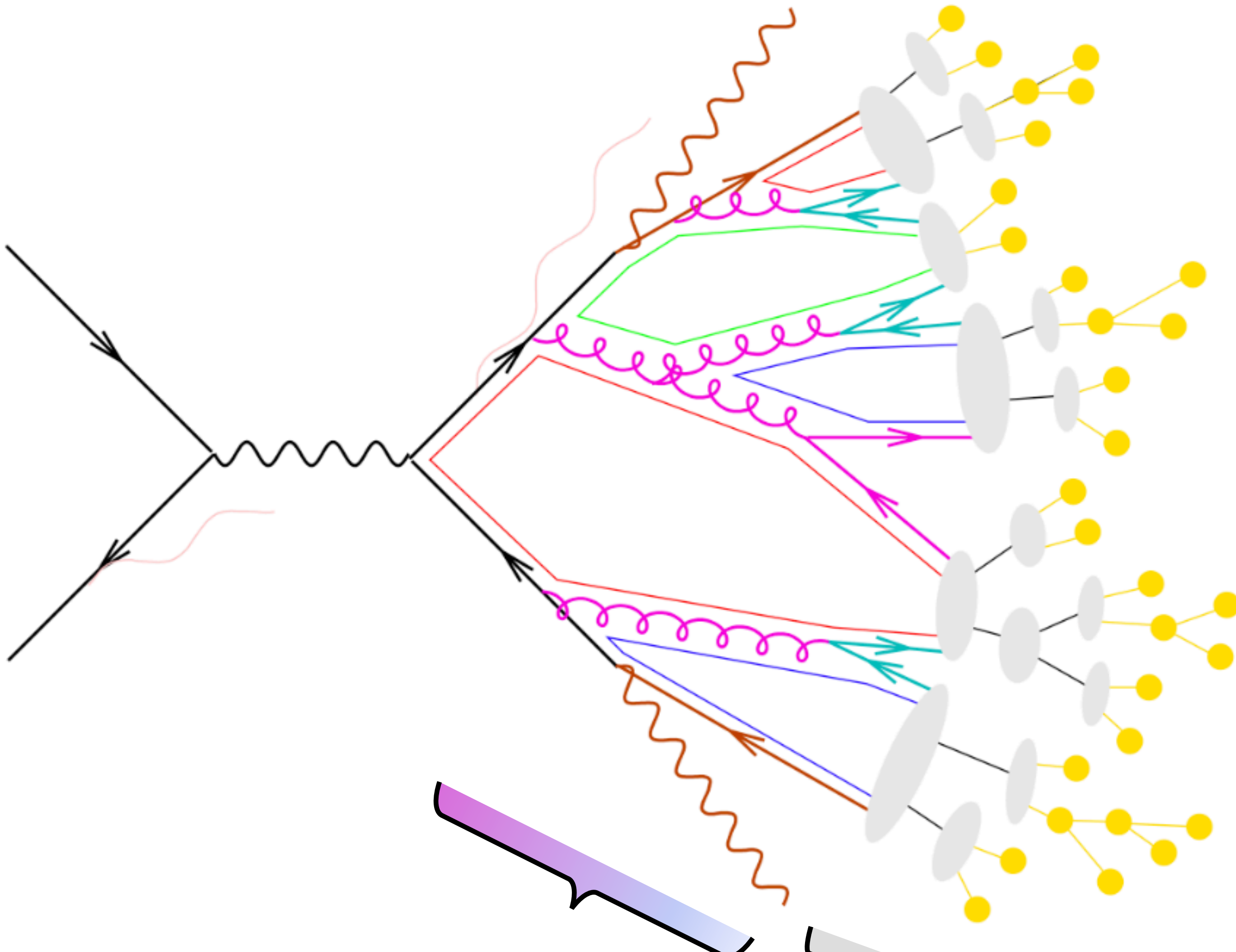
EPD Seminar · IHEP, CAS
2 September, 2022

# Preview of this talk

# Part I: Background

# Jets in hadron colliders

*"A jet is a collimated shower of particles produced by the hadronization of a quark or gluon"*

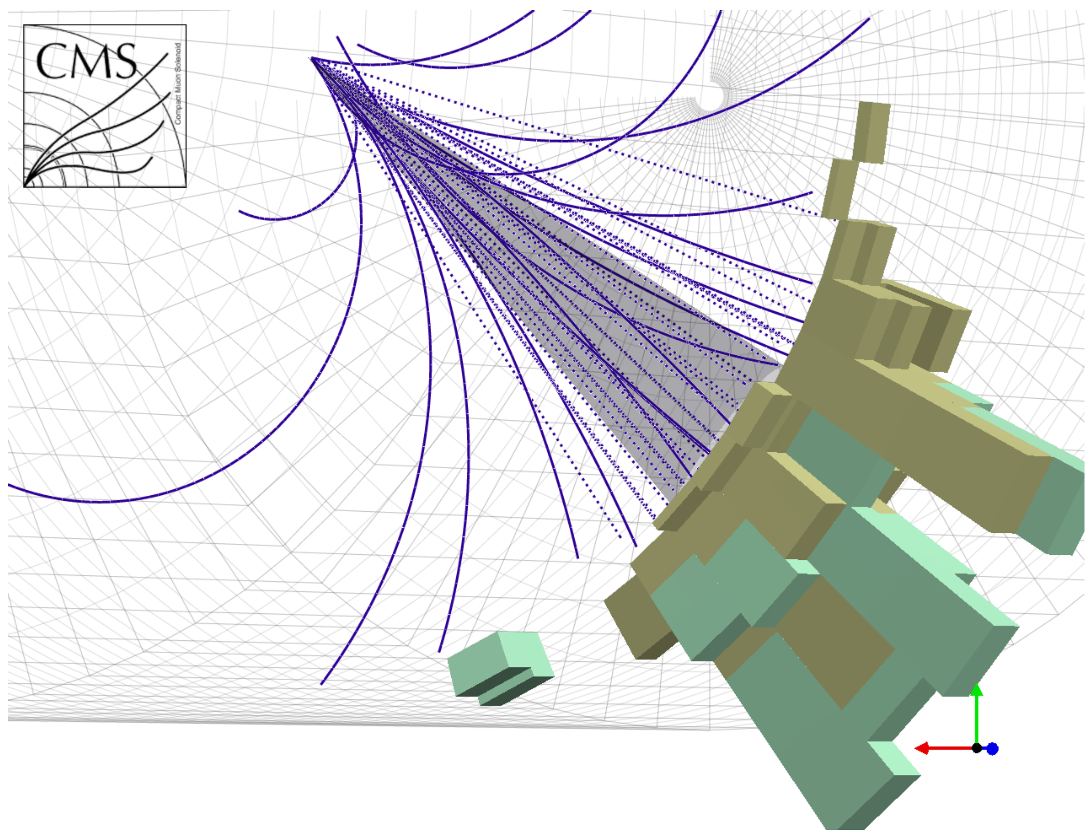


*parton showering*

*hadronization*

*hadronic decay*

*⇒ stable hadrons*



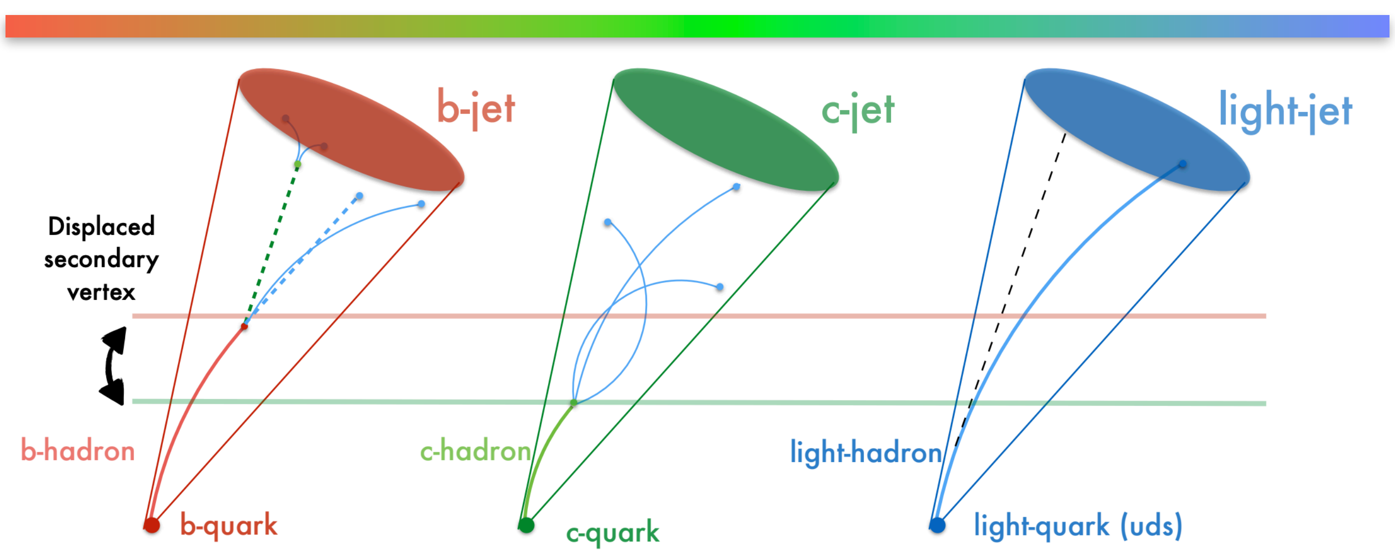

*raw data from tracker & calorimeter*

*→ reconstruct to particle records*

# Jet tagging

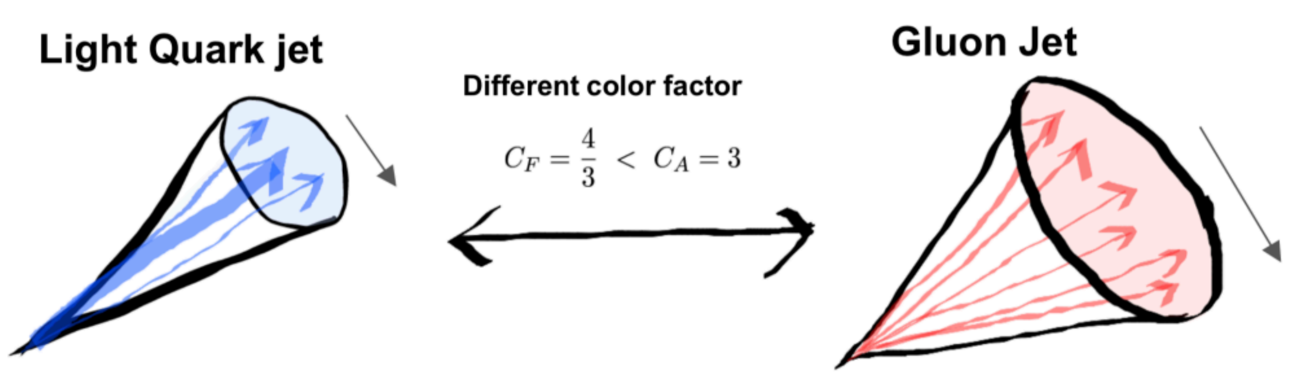
- ➔ Jet tagging: determine the origin of a jet
- ➔ Two jet tagging prototypes

*(1) jet flavour tagging*




**heavy flavour tagging:**

✤ jets initiated from a b/c/light quark differ by constituent multiplicity and trajectory displacement



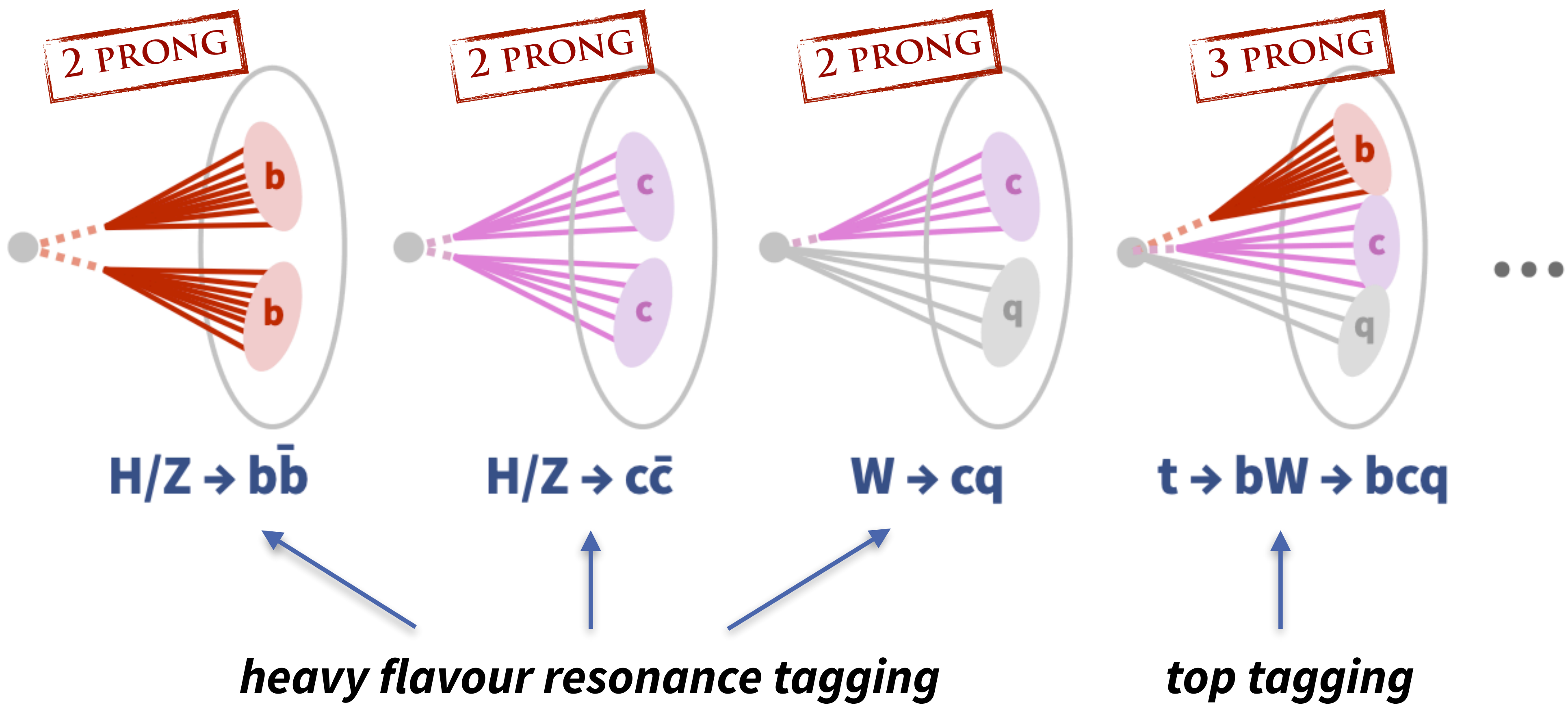

**quark/gluon jet discrimination:**

✤ a quark jet has more constituent particles and is more collimated to the jet axis

# Jet tagging

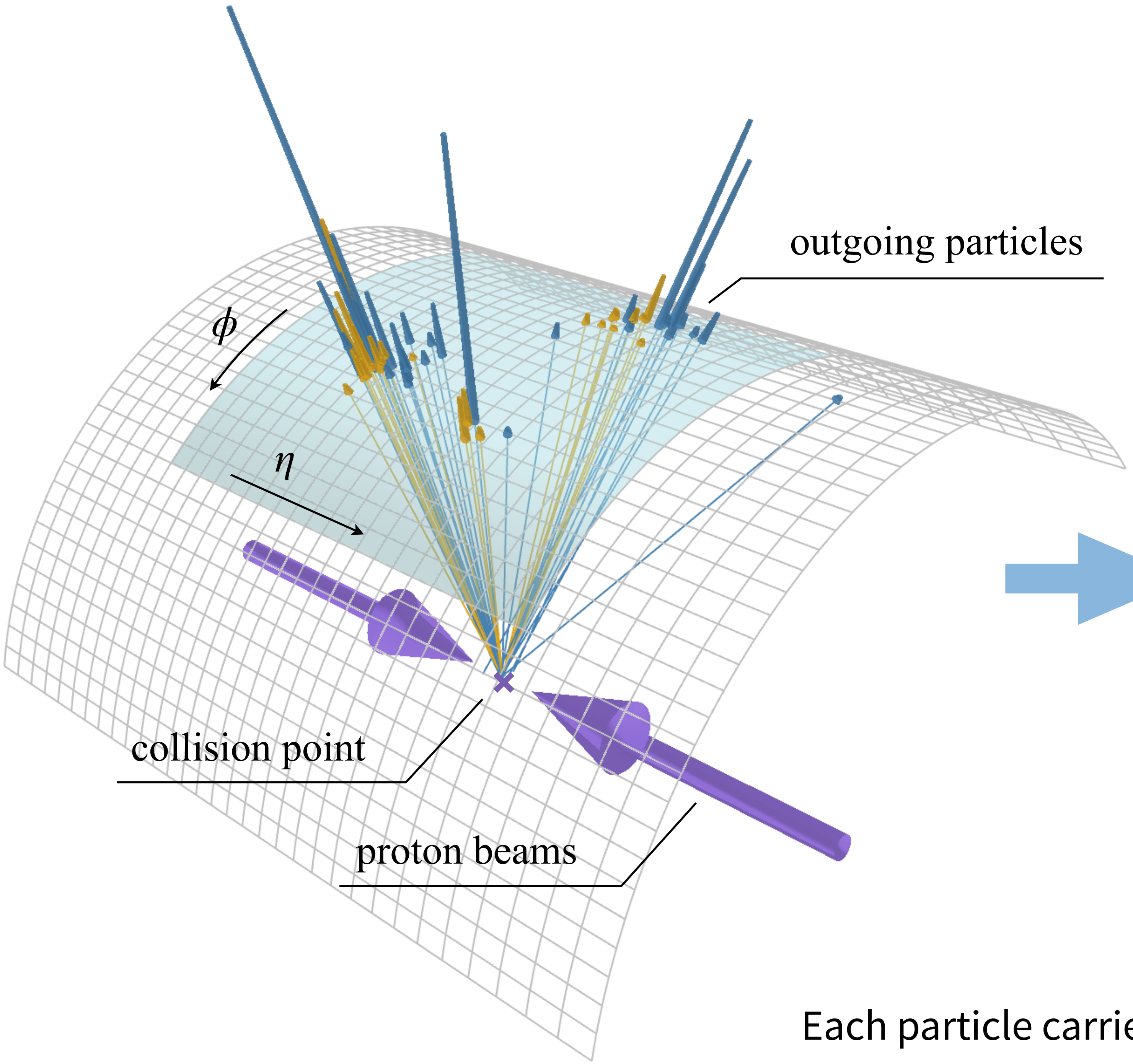
- Jet tagging: determine the origin of a jet
- Two jet tagging prototypes

*(2) boosted jet tagging*

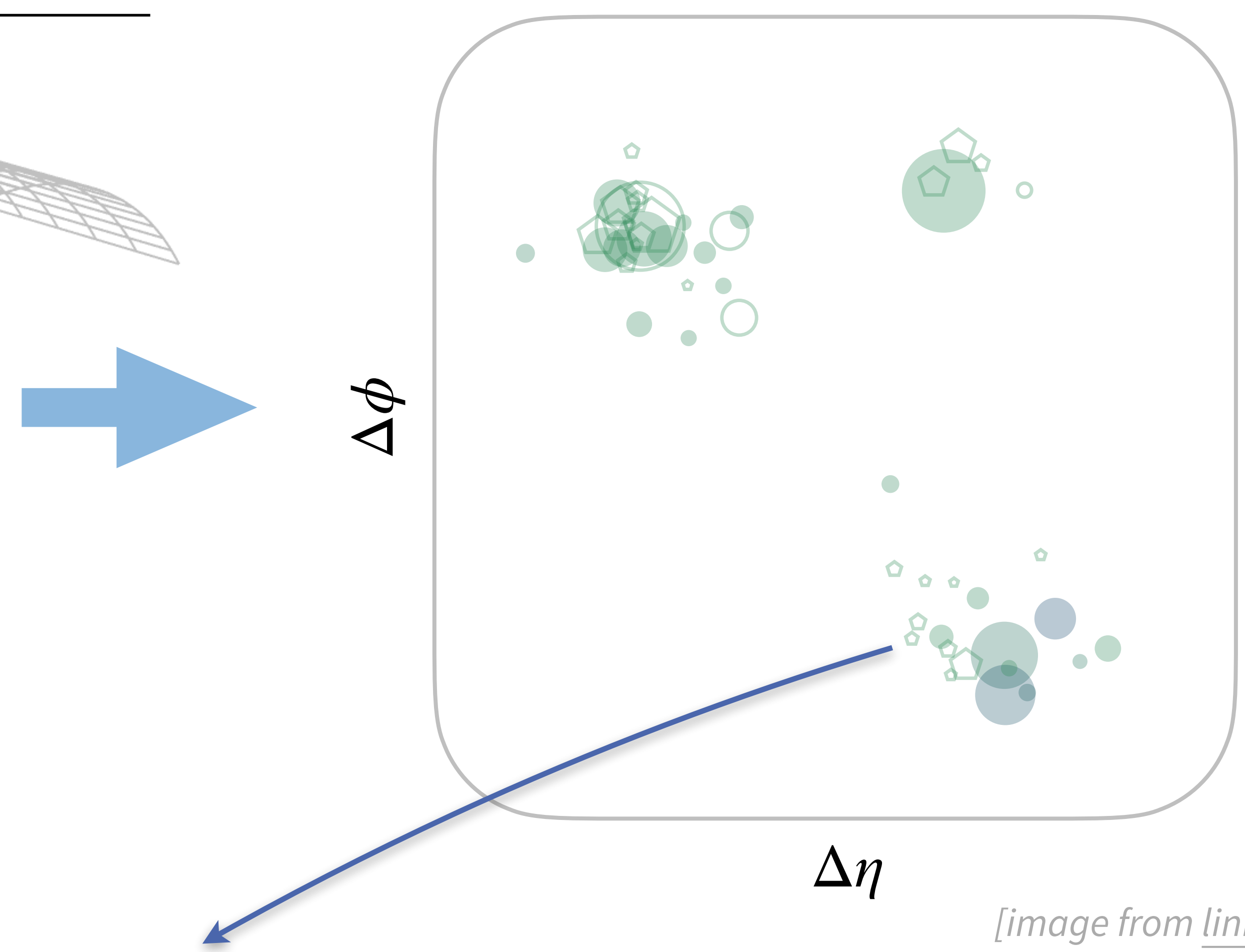



✤ differs by (1) proneness; (2) existence of heavy flavour subjets (initiated by b/c quarks)

# View of a jet




*Viewing jets on the η-ϕ space*




*[image from link]*

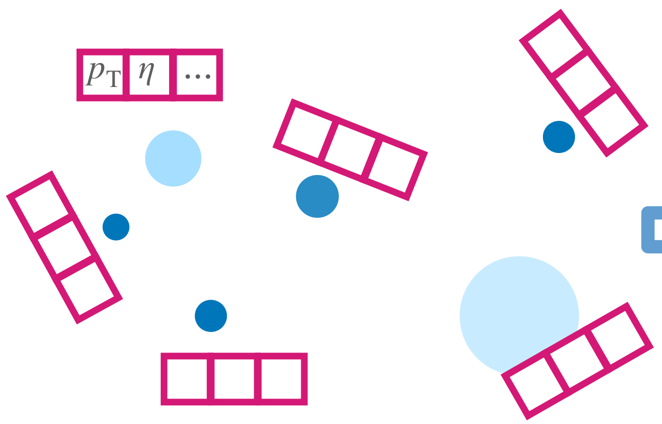
Each particle carries features:

- four-momentum; or equiv. $(E, p_{\mathrm{T}}, \eta, \phi)$
- particle ID *
- track displacement (for charged particle) *

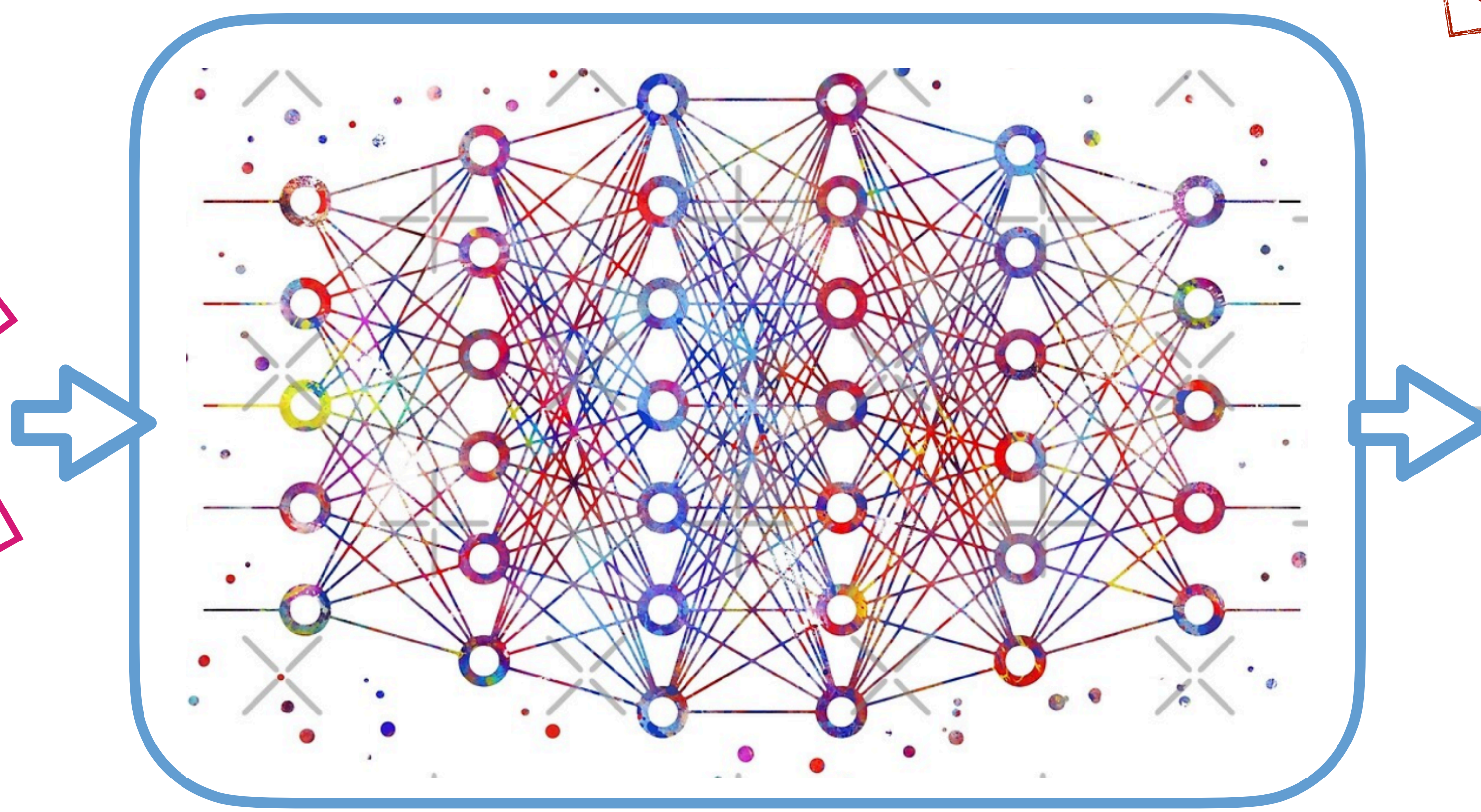
** not necessarily exists*

# Jet physics meets deep learning

INPUT

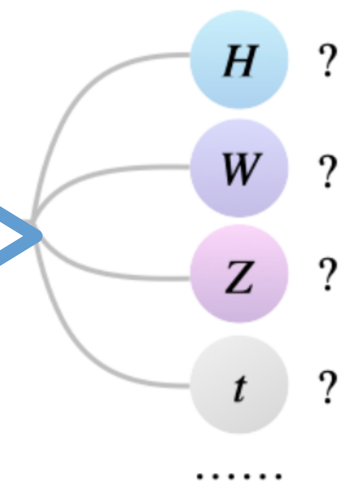



A jet: described by particle-level (low-level) features



OUTPUT

**original of the jet?**




**other jet properties?**

We are embracing a new era of mankind in which AI starts to reshape science and industries. (the stage regarded as “The 4th Industrial Revolution”)

Future collider physics will be highly influenced by the advancement in AI and deep learning. Jet physics is one of the entry points.

# Jet physics meets deep learning

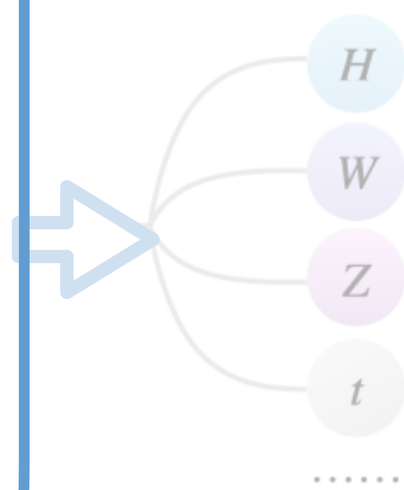
INPUT



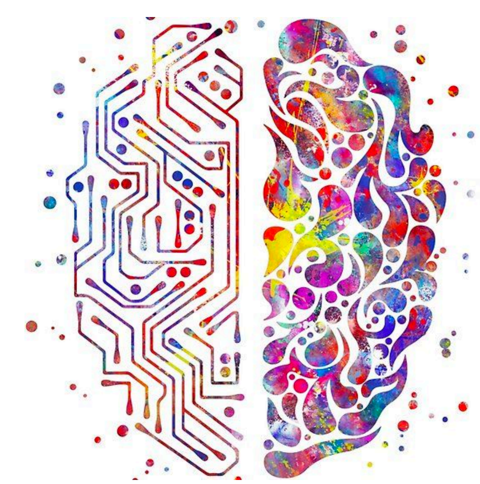

A jet: described by particle-level (low-level) features

OUTPUT

**original of the jet?**



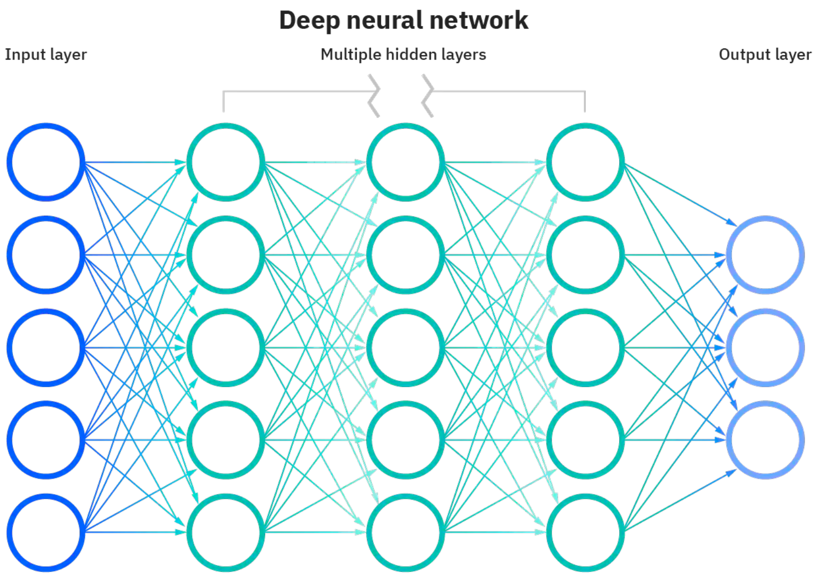

*Our research objective:*
*design a **most performant** deep learning model dedicated to jet physics*



*build a "smarter" AI*

# Roadmap of DL model for jet tagging

➔ DL model design draw from experiences in Computer Vision

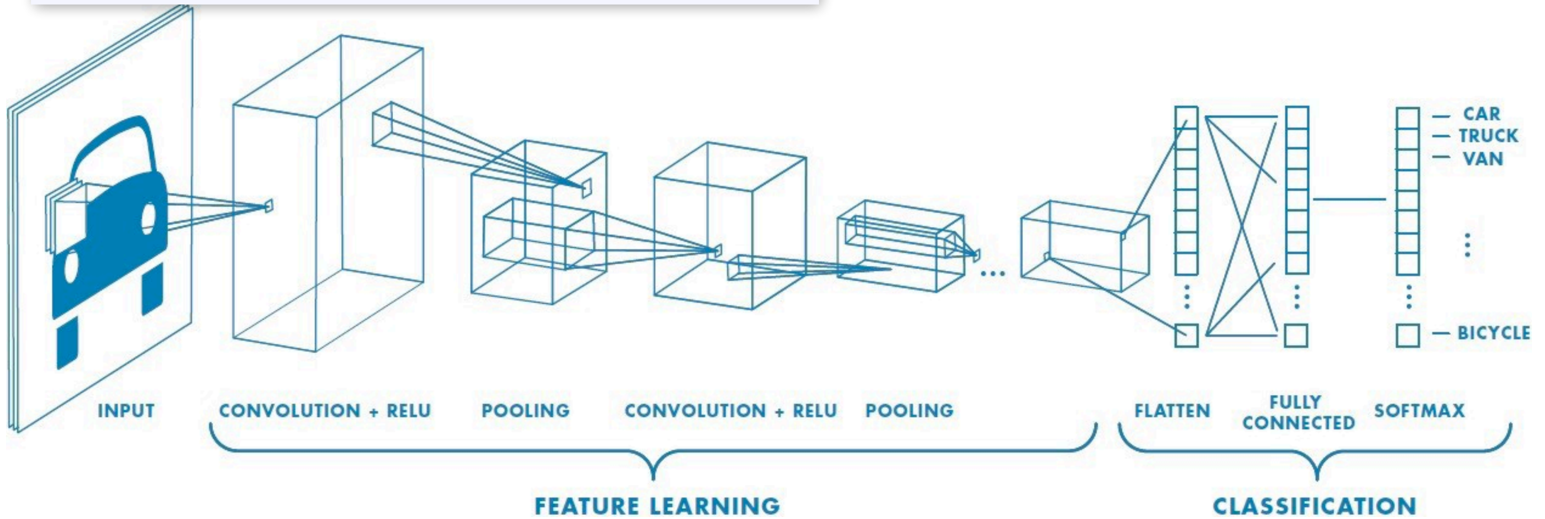



*conventional "deep neural network" or multi-layer perceptron (MLP)*

process fix-length input data

*convolutional neural network (CNN)*

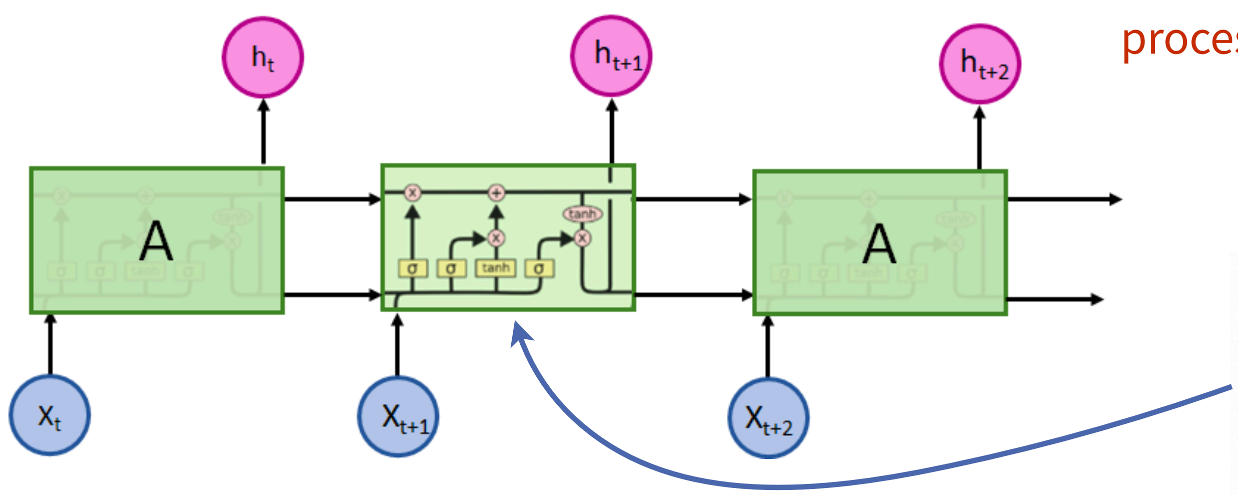
process images

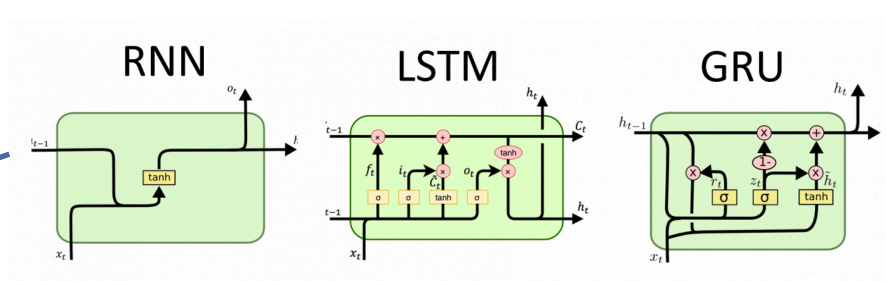



[image from link]

*recurrent neural network (RNN) & LSTM*

process "sequence" of input, e.g. sentences

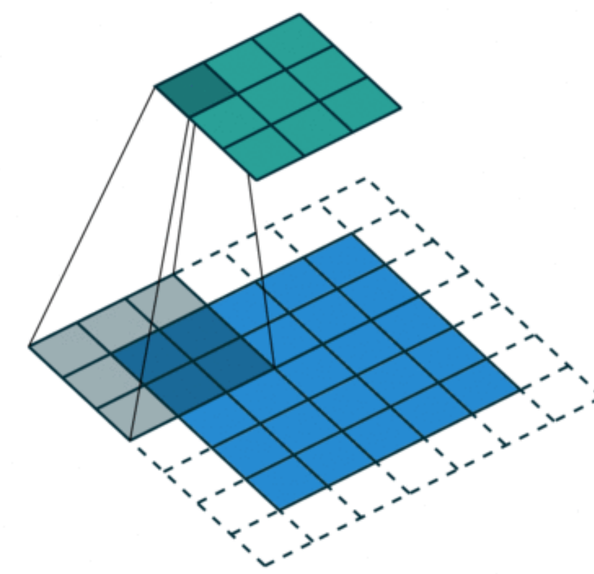


# Roadmap of DL model for jet tagging

EXAMPLE OF CNN

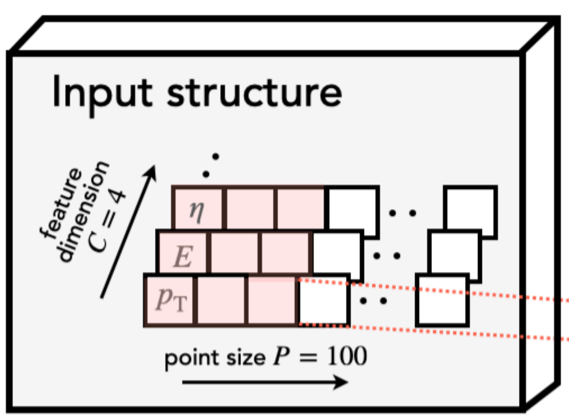
illustration of 2D convolution

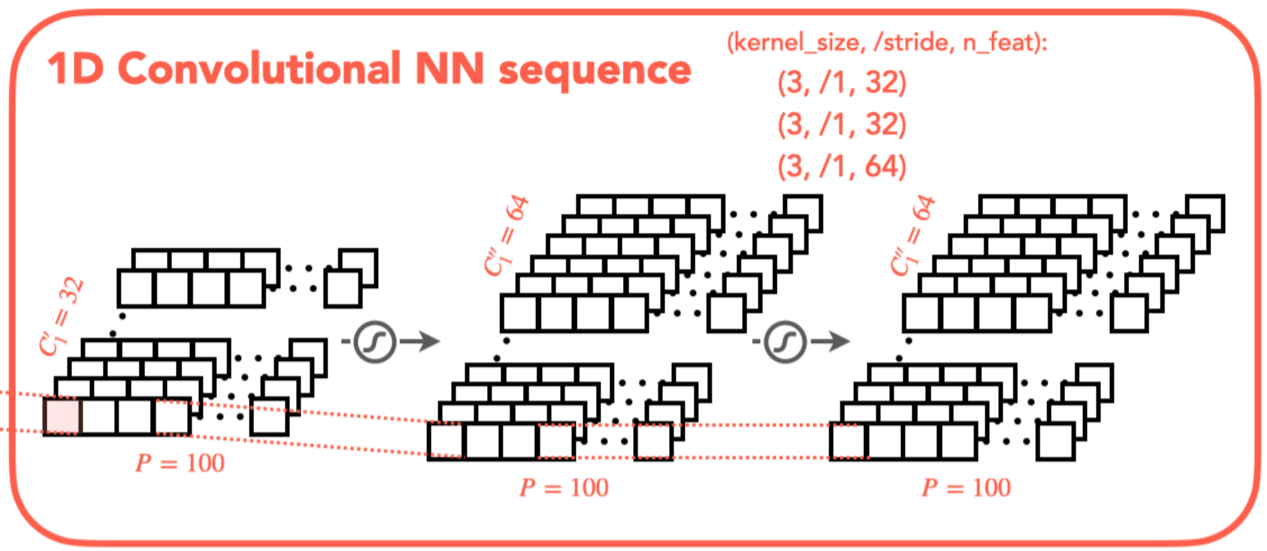


The previous jet tagging model in CMS: **DeepAK8** algorithm

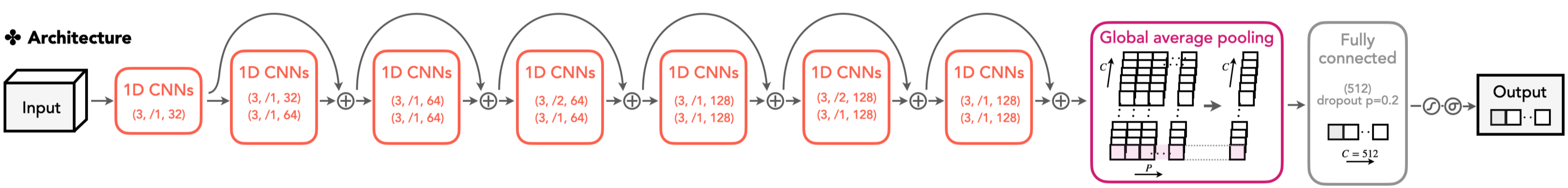
CMS, JINST 15 (2020) P06005

✤ **Building block**




✤ **Architecture**



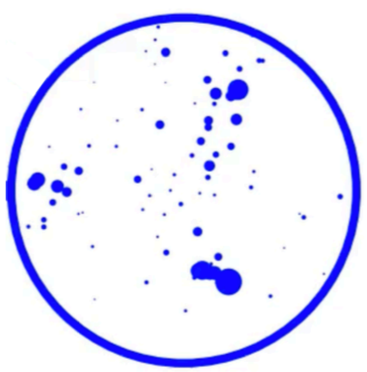

[image from link]

# Roadmap of DL model for jet tagging

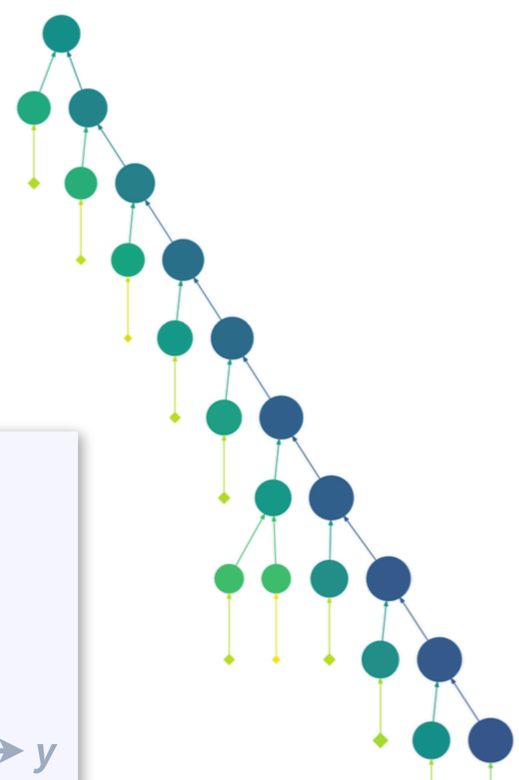
➔ Graph neural networks: view input particles as a set / graph

❖ guarantee the *permutational invariance* of input particles

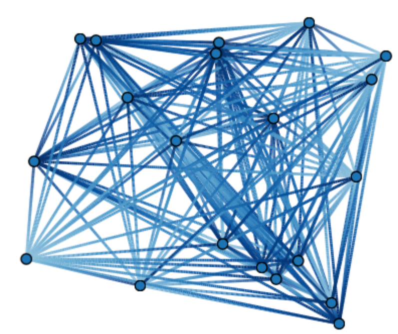
*Set: no edges*



*Hierarchical trees:*
- *decay chain*
- *jet clustering history*

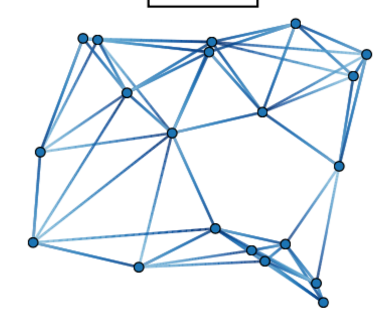


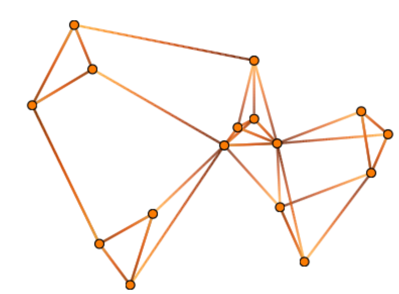
*Fully connected graph*
- *i.e., connect each node to all other nodes*



*Locally connected graph*
- *i.e., connect each node only to neighbor nodes*
  - *k-nearest neighbors*
  - *fixed radius*

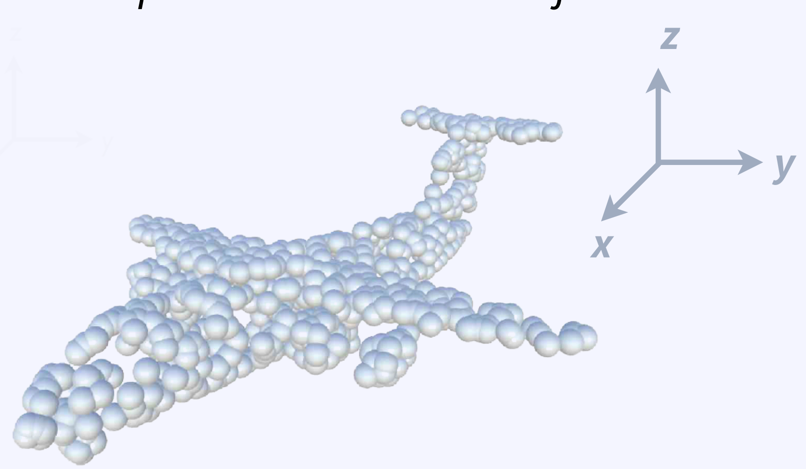
static





(dynamically) learned

*analog to point cloud representation of 3D objects*




*[image from link]*

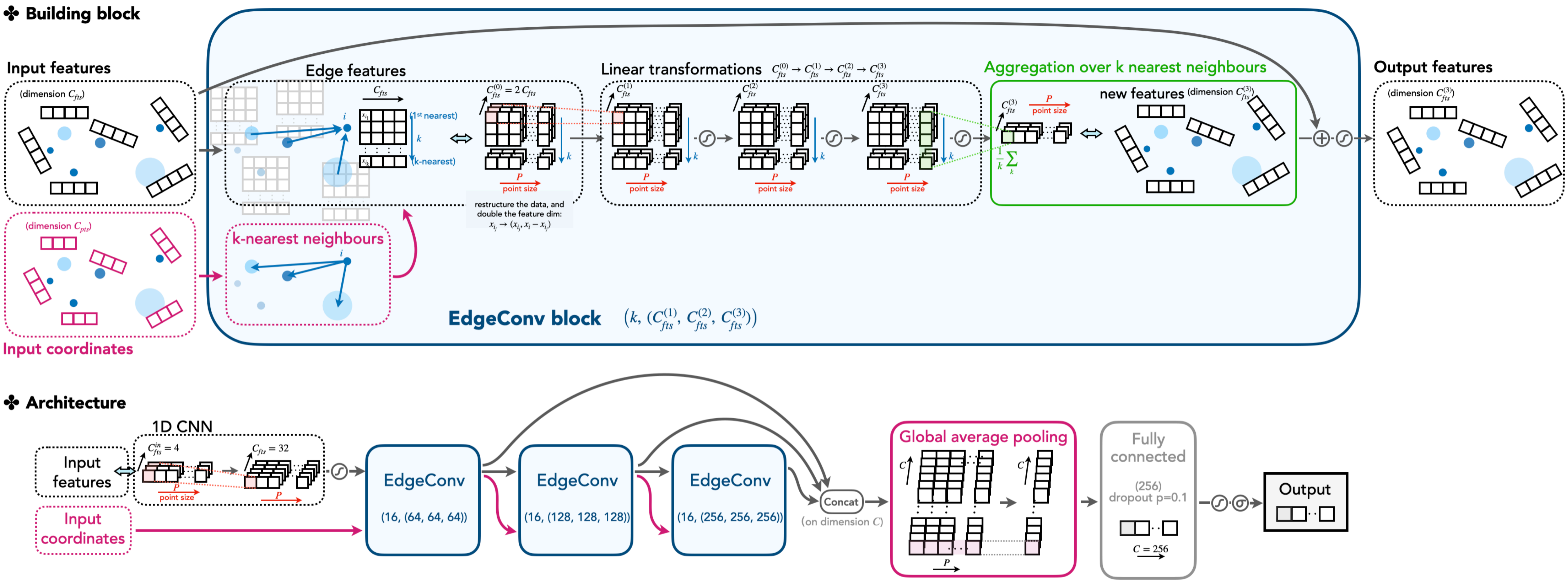
# Roadmap of DL model for jet tagging

RECAP ON PARTICLENET

H.Qu, L.Gouskos. PRD 101 (2020) 056019

A powerful and popular model in the HEP community with a variety of applications

[image from link]



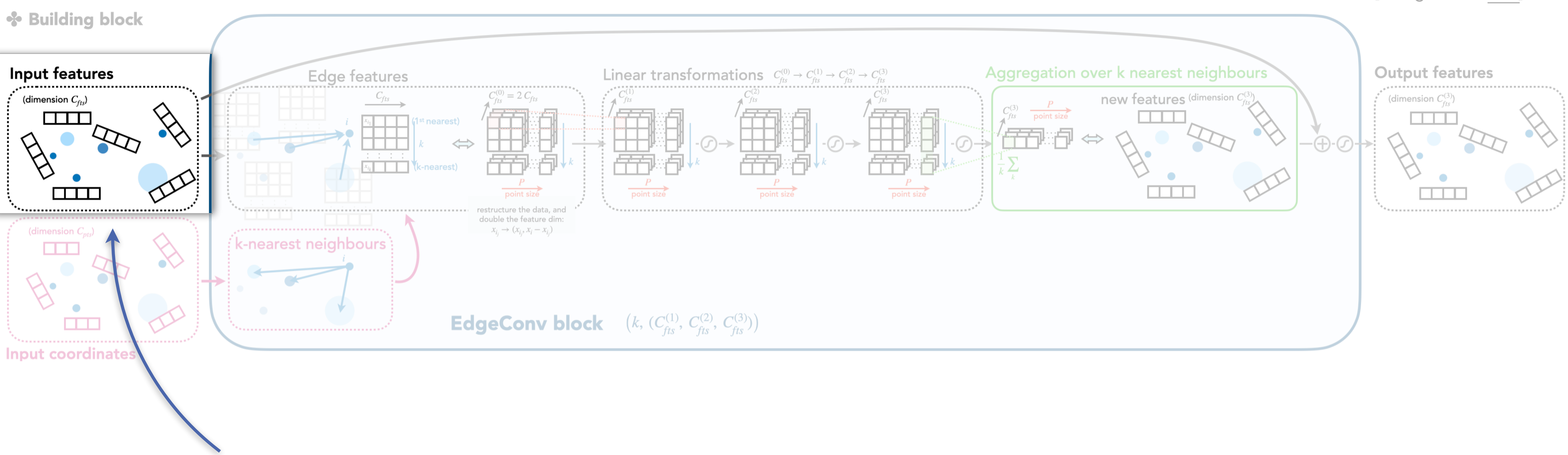
# Roadmap of DL model for jet tagging

RECAP ON PARTICLENET

H.Qu, L.Gouskos. PRD 101 (2020) 056019

A powerful and popular model in the HEP community with a variety of applications

*[image from link]*




*Point cloud representation of jet*

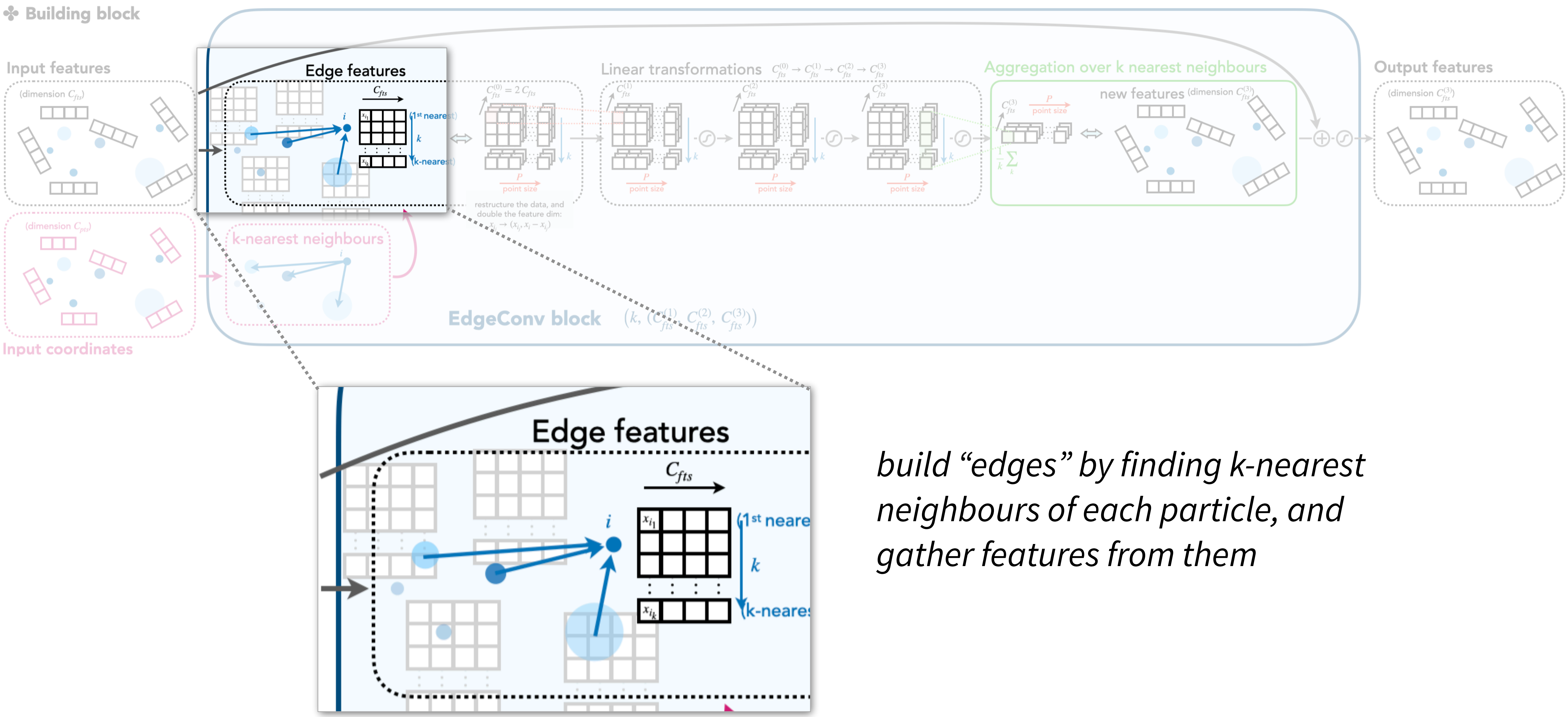
# Roadmap of DL model for jet tagging

RECAP ON PARTICLENET

H.Qu, L.Gouskos. PRD 101 (2020) 056019

A powerful and popular model in the HEP community with a variety of applications

[image from link]




*build "edges" by finding k-nearest neighbours of each particle, and gather features from them*

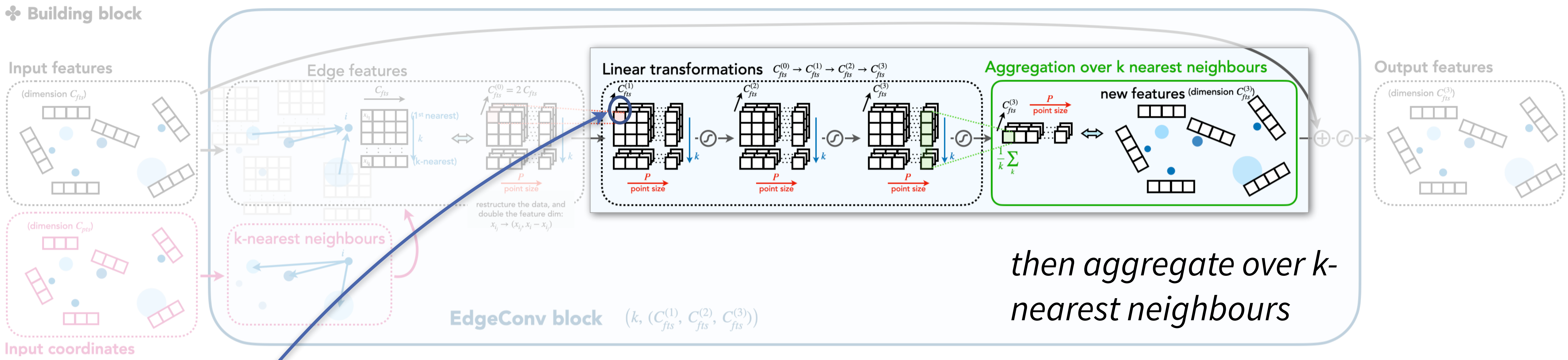
# Roadmap of DL model for jet tagging

RECAP ON PARTICLENET

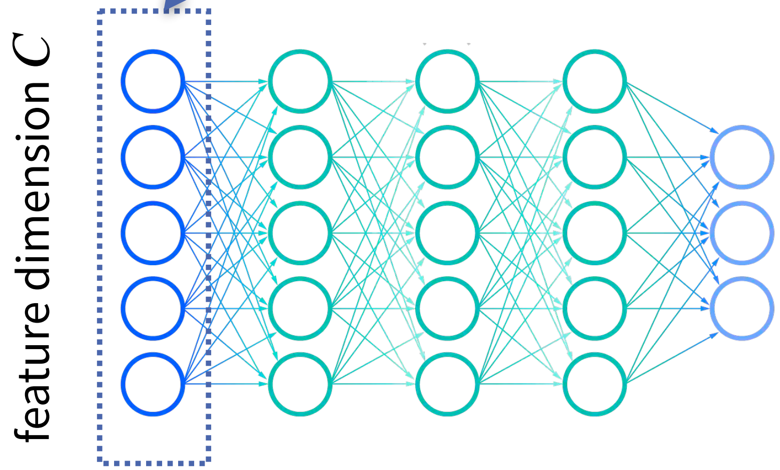
H.Qu, L.Gouskos. PRD 101 (2020) 056019

A powerful and popular model in the HEP community with a variety of applications

[image from link]




*then aggregate over k-nearest neighbours*

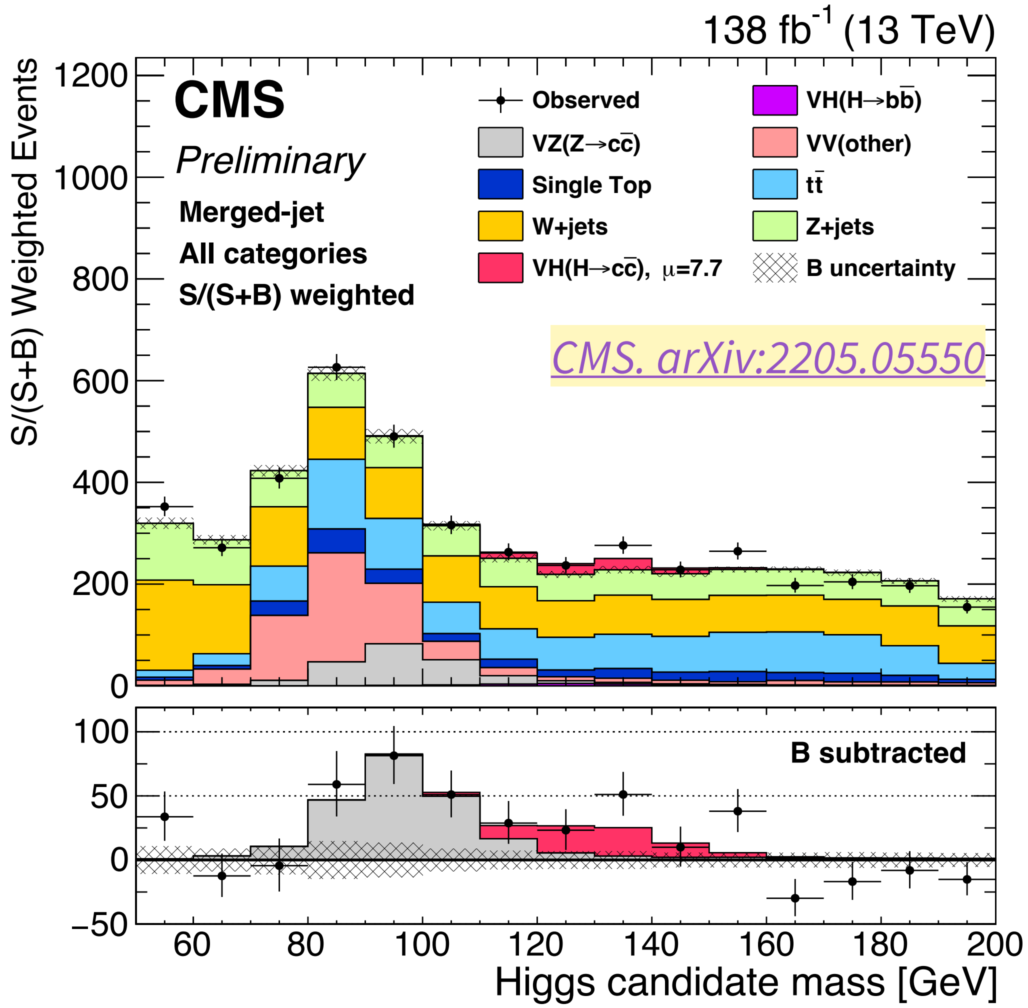


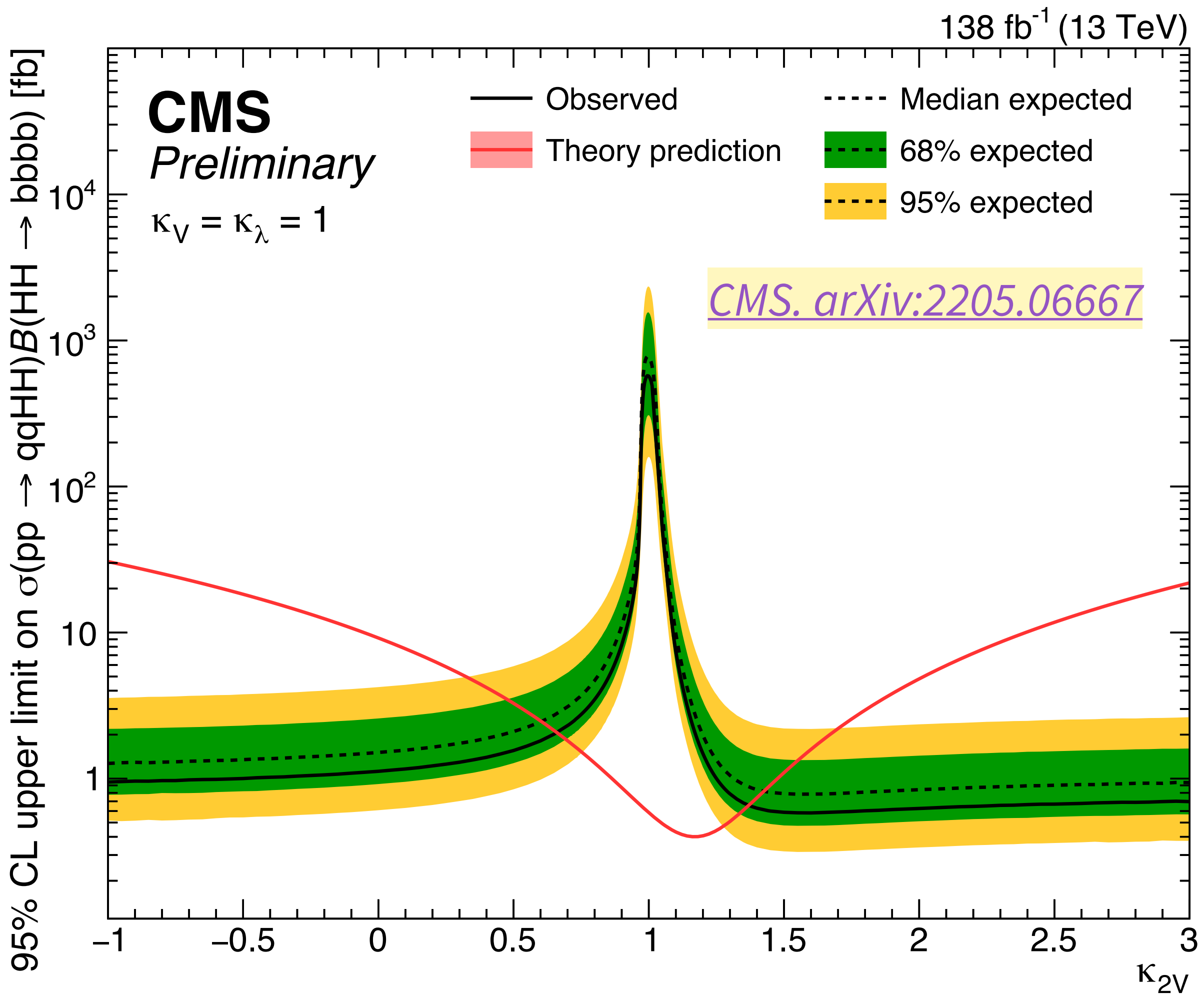

*do linear transformation*

# Roadmap of DL model for jet tagging

RECAP ON PARTICLENET

H.Qu, L.Gouskos. PRD 101 (2020) 056019




application to VH→cc̄ search
Most stringent limit on H-c coupling to date

application to SM boosted HH→4b search
First time excluding $\kappa_{2V} = 0$

# "Post-ParticleNet" DL studies

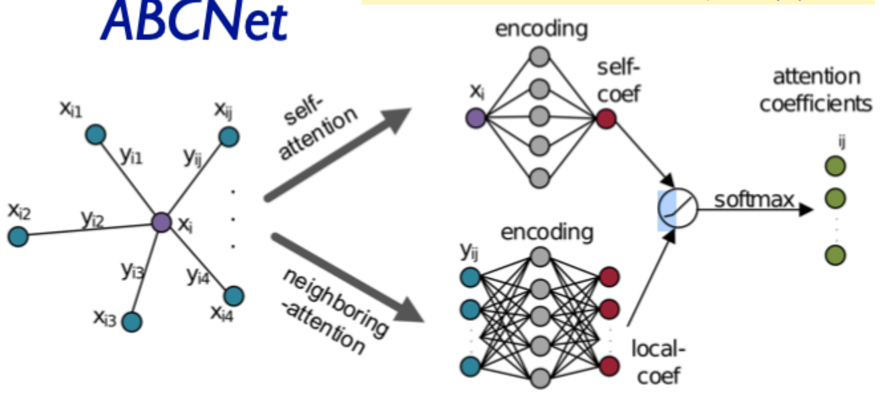
*disclaimer: only shows a part of relevant works*

➔ Further study to enhance the jet tagging model mainly divided into two approaches

***More advanced model***

## ABCNet

V. Mikuni et al. EPJC 2020; 135(6): 463

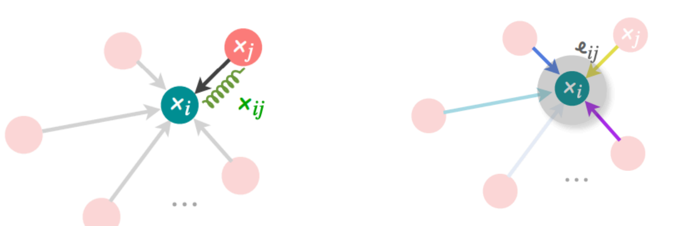



*add attention mechanism in addition to graph convolution*

## ParticleNeXt

H.Qu. Talk@ML4Jets2021

*attentive pooling; multi-scale aggregation; additional pairwise features for edge*



$e_{ij} = MLP(x_i, x_j, \boldsymbol{x}_{ij})$

$attn_{ij} = MLP(e_{ij})$

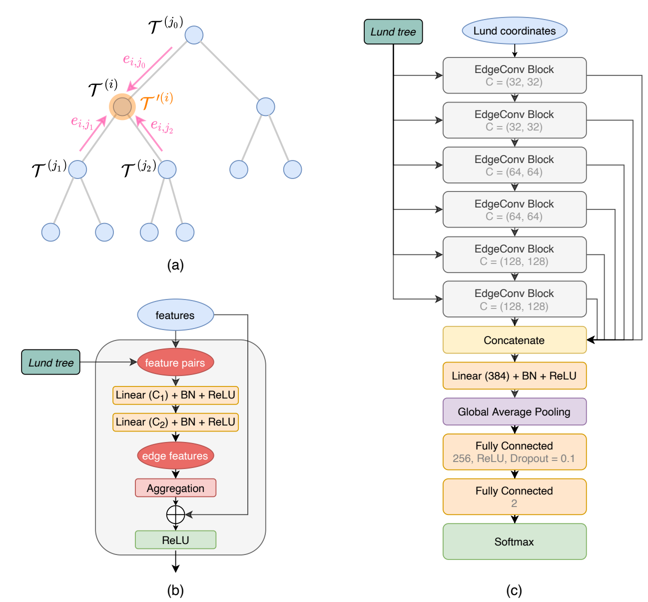
$w_{ij} = softmax_j(attn_{ij})$

$z_i = \Sigma_j(w_{ij}\, e_{ij})$

***physics-inspired design/modifications***

## LundNet

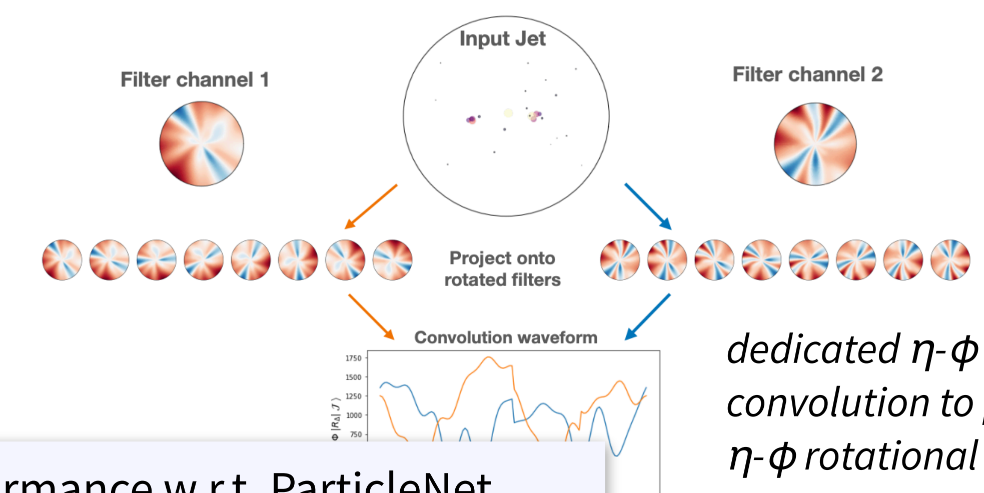
F.Dreyer et al. JHEP 03 (2021) 052




*fix graph structure according to Lund plane; use physics variables to build edges*

C.Shimmin. arXiv:2107.02908

## Particle Convolution Network




*dedicated $\eta$-$\phi$ convolution to preserve $\eta$-$\phi$ rotational symmetry*

small improvement or comparable performance w.r.t. ParticleNet, evaluated on two mainstream benchmarks

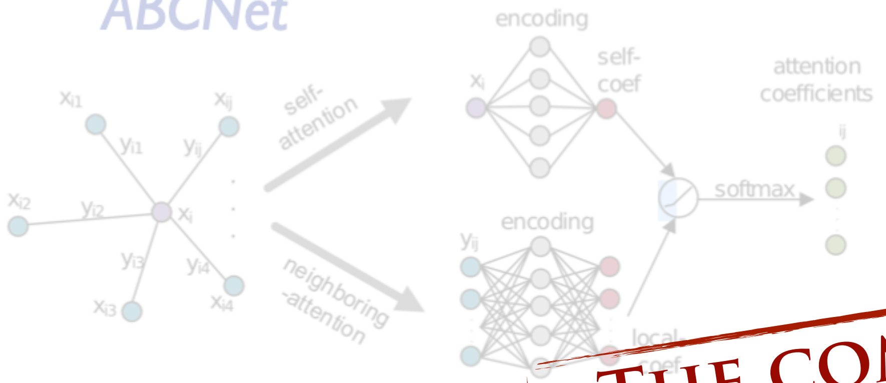
# "Post-ParticleNet" DL studies

*disclaimer: only shows a part of relevant works*

➔ Further study to enhance the jet tagging model mainly divided into two approaches

***More advanced model***

**ABCNet** — V. Mikuni et al. EPJC 2020; 135(6): 463



*add attention mechanism in addition to graph convolution*

**ParticleNeXt** — H.Qu, Talk@ML4Jets2021



*attentive pooling; multi-scale aggregation; additional pairwise features for edge*

$e_{ij} = MLP(x_i, x_j, x_{ij})$

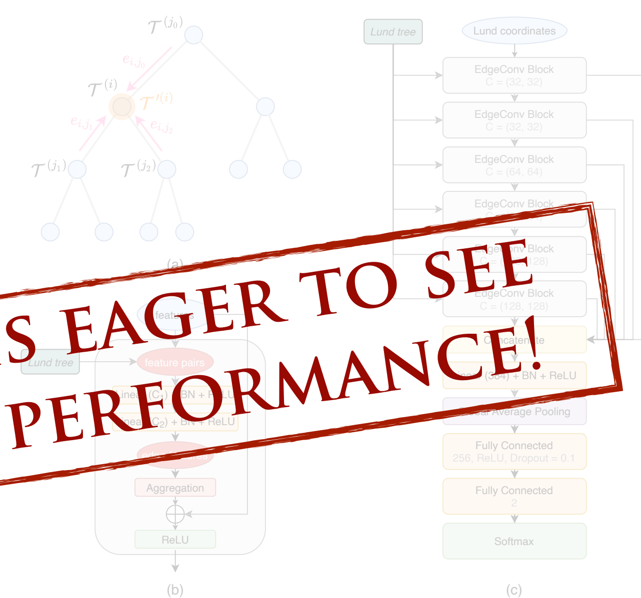
$attn_{ij} = MLP(e_{ij})$

$w_{ij} = softmax_j(attn_{ij})$

$z_i = \Sigma_j(w_{ij}\, e_{ij})$

***physics-inspired design/modifications***

**LundNet** — F.Dreyer et al. JHEP 03 (2021) 052



*fix graph structure according to Lund plane; use physics variables to build edges*

**Particle Convolution Network** — C.Shimmin, arXiv:2107.02908




*dedicated η-ϕ convolution to preserve η-ϕ rotational symmetry*

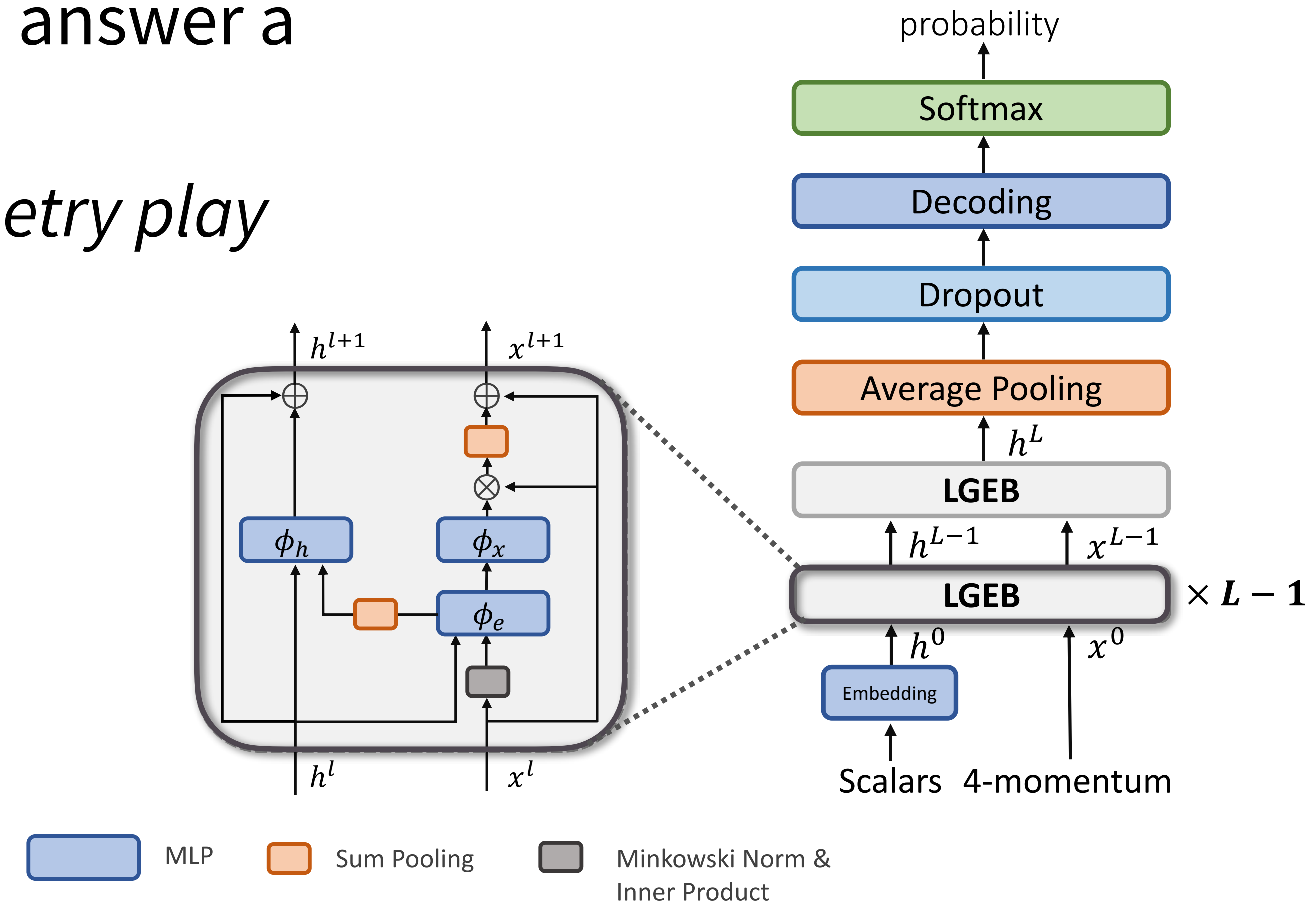
THE COMMUNITY IS EAGER TO SEE THE NEXT LEAP IN PERFORMANCE!

small improvement or comparable performance w.r.t. ParticleNet, evaluated on two mainstream benchmarks

# Part II: LorentzNet

# Introducing LorentzNet

- ***Design of LorentzNet makes a successful attempt***
  - made up of fully-connected GNN
  - its outstanding performance largely comes from Lorentz symmetry preservation
  - (note: recently the record is reset by *ParT* which we discuss in p.51)
- We'll first start our journey in LorentzNet, then in the next part we try to answer a broader question: *which role does Lorentz-symmetry play in jet tagging*

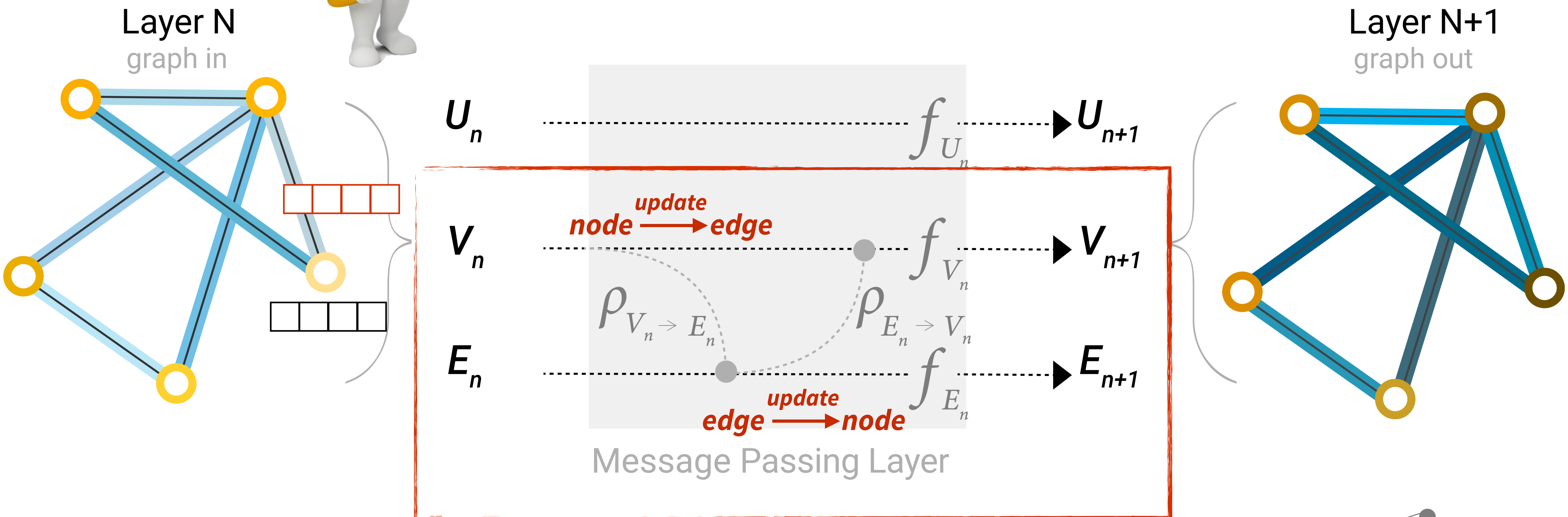



Lorentz Group Equivariant Block (LGEB)

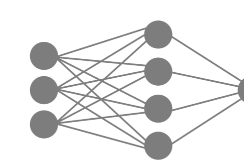
***LorentzNet***

# Graph neural networks

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




***Message passing mechanism is an important component of GNN***

update function $f =$  , …

pooling function $\rho$

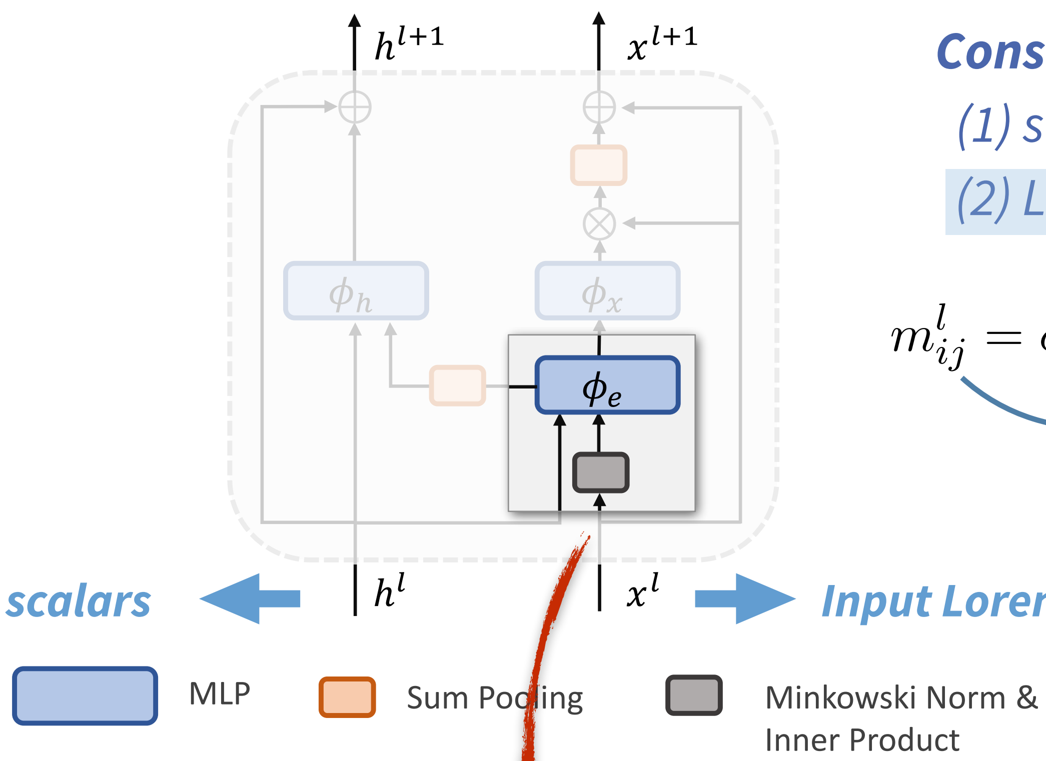
# LorentzNet architecture

*Construct edge features from*
*(1) scalars from two nodes*
*(2) Lorentz inner product from two vectors*

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

*hence it's a Lorentz scalar*

**Input Lorentz scalars**




**Input Lorentz vector**

**Lorentz Group Equivariant Block (LGEB)**

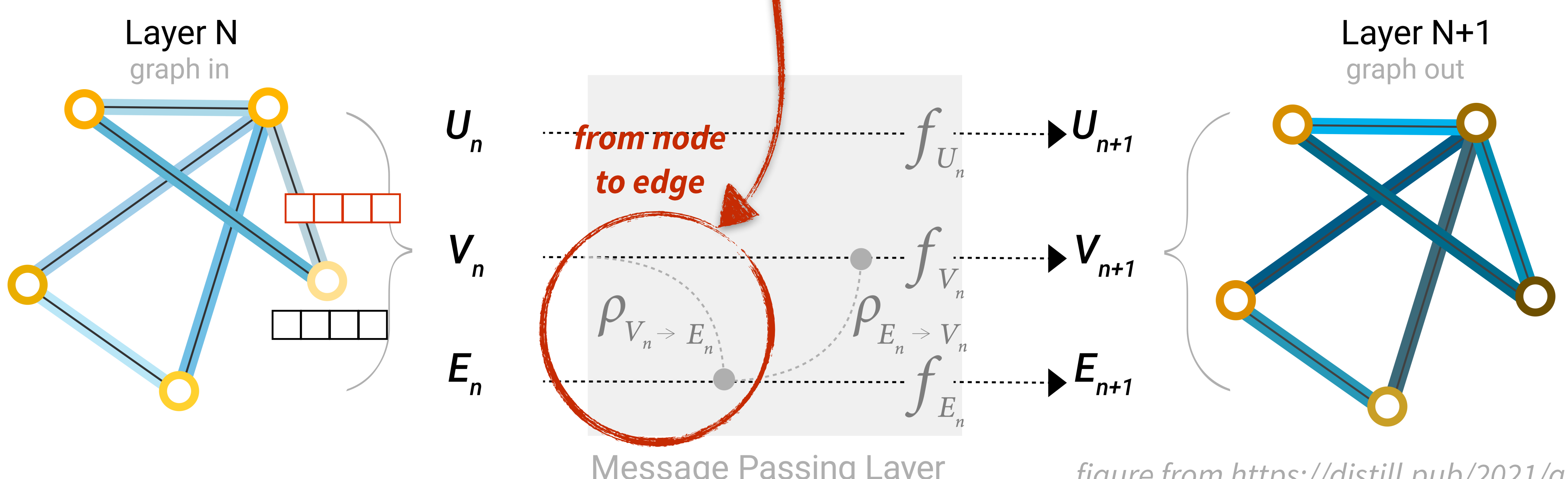
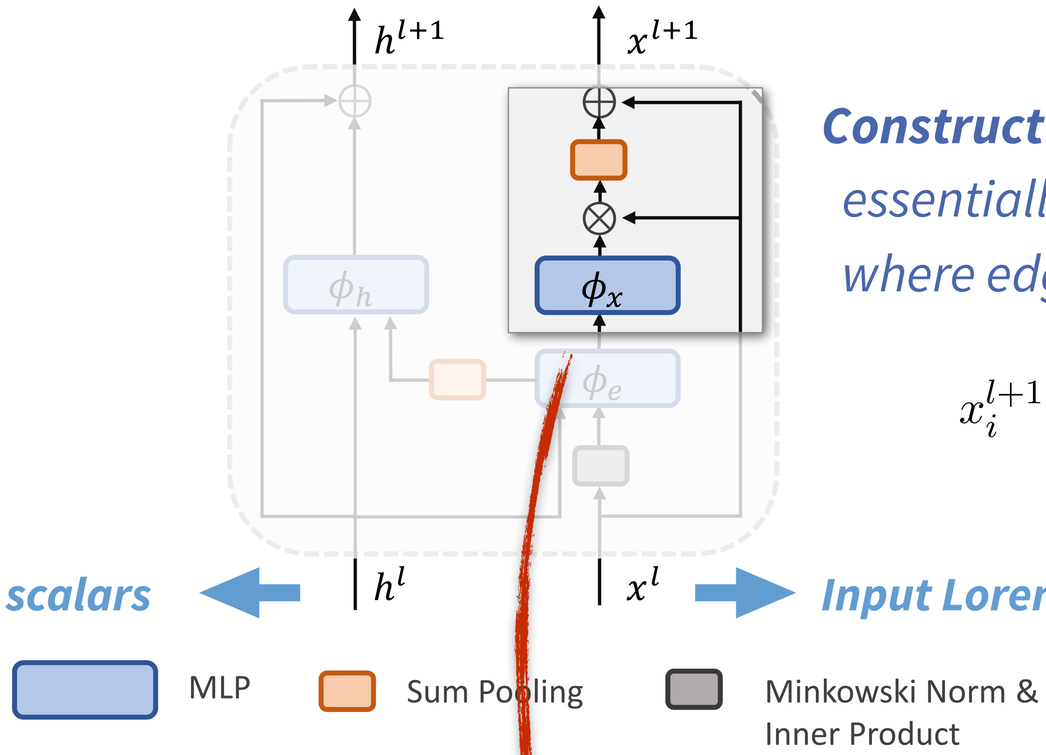



*figure from https://distill.pub/2021/gnn-intro/*

# LorentzNet architecture




Lorentz Group Equivariant Block (LGEB)

***Construct node features (vector):***

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

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

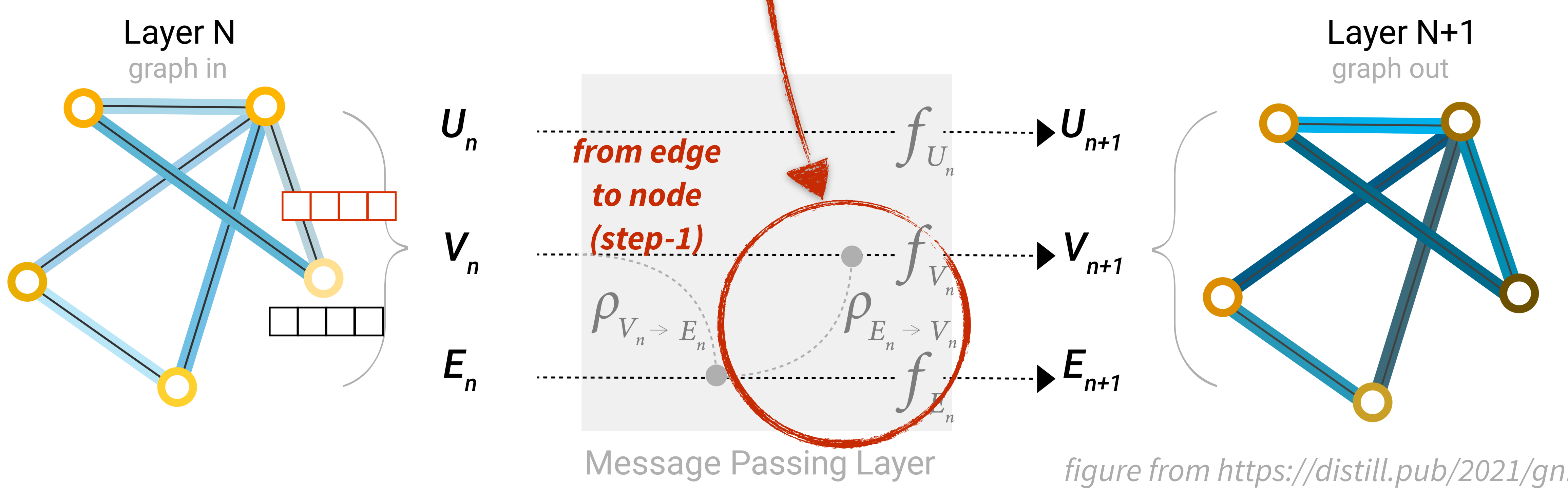
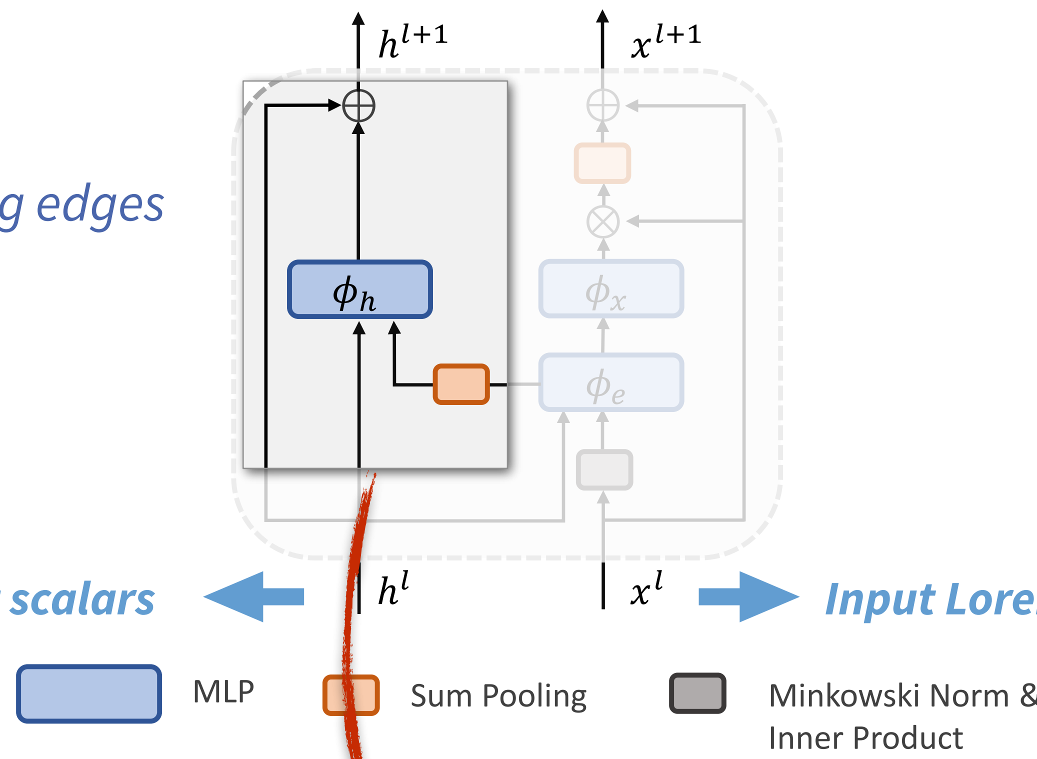



*figure from https://distill.pub/2021/gnn-intro/*

# LorentzNet architecture

***Construct node features (scalar):***
*attentive pooling on all connecting edges*

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




**Lorentz Group Equivariant Block (LGEB)**

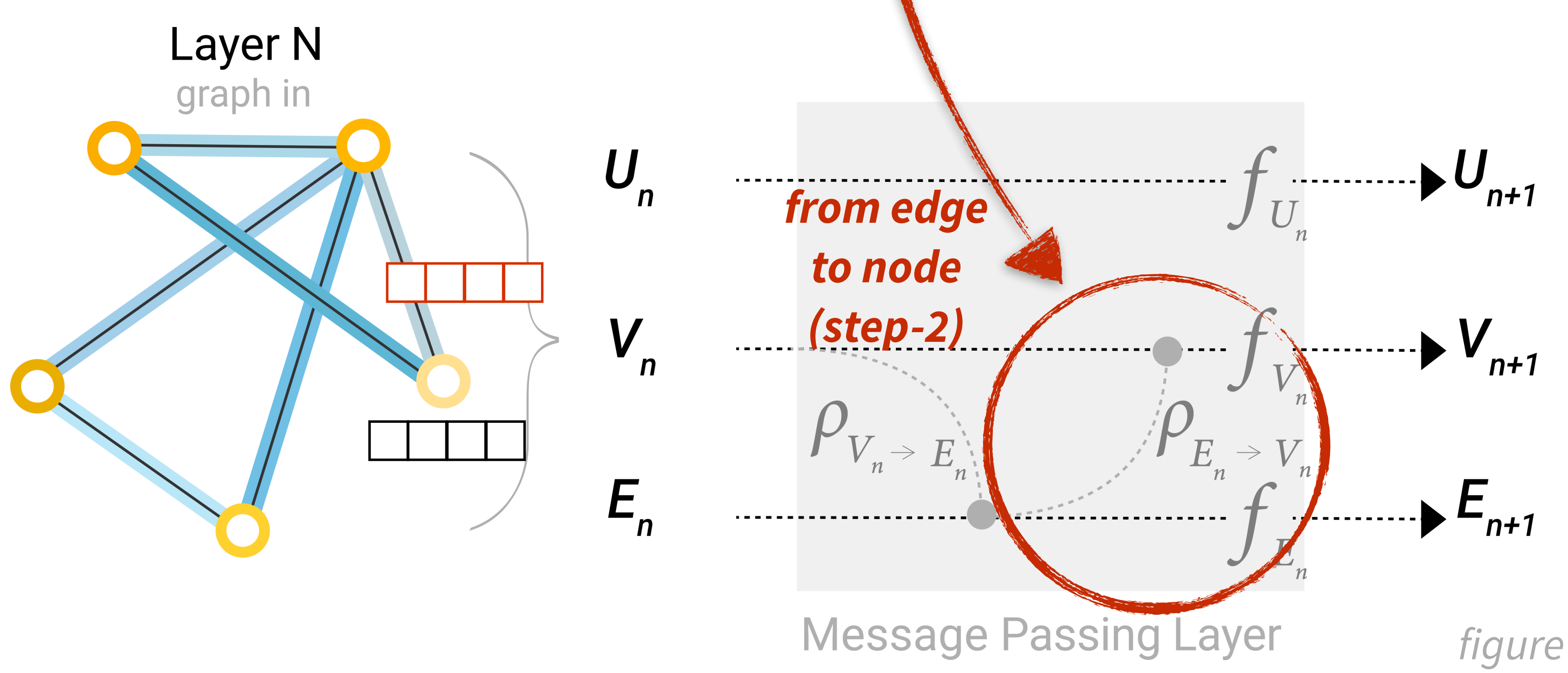
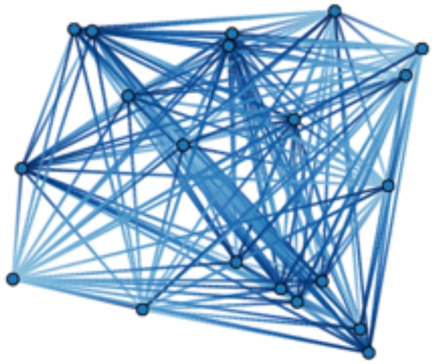


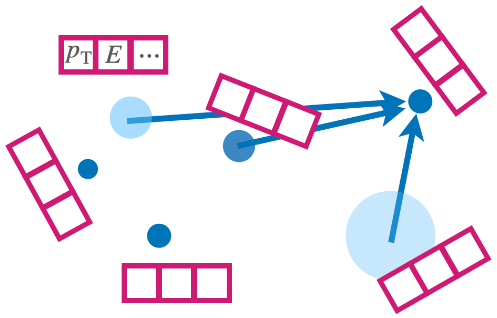

*figure from https://distill.pub/2021/gnn-intro/*

# 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



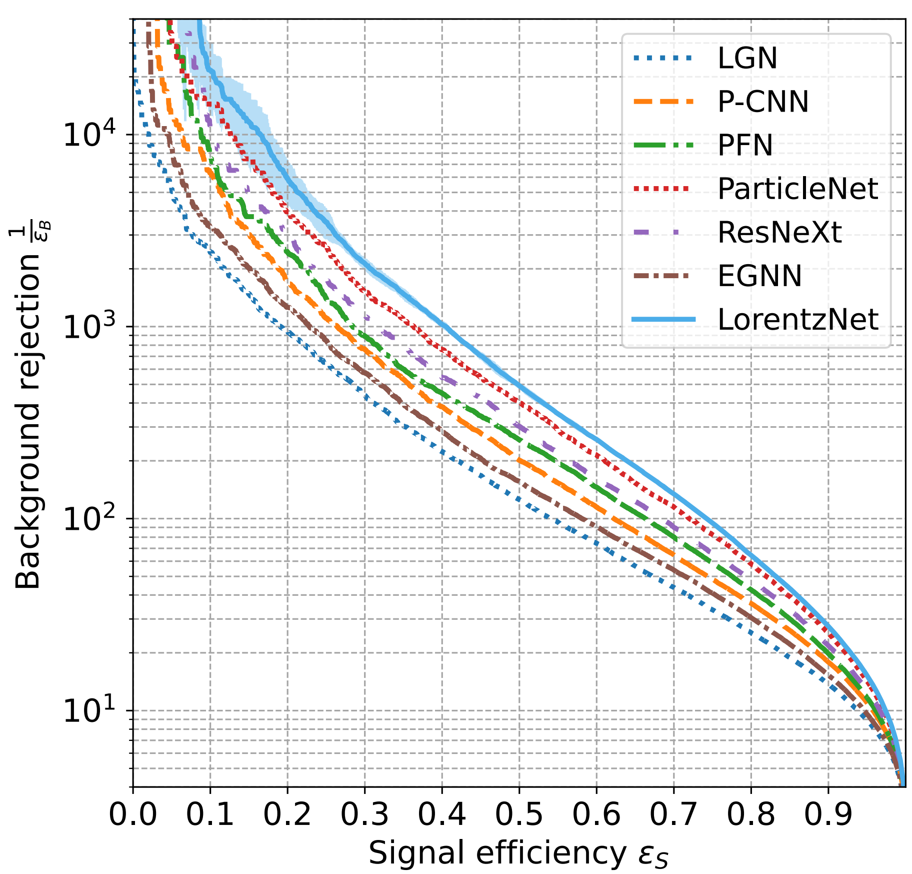



- ***Fully Lorentz invariant/equivariant***
  - nodes can be grouped by either Lorentz scalars or vectors

# Performance

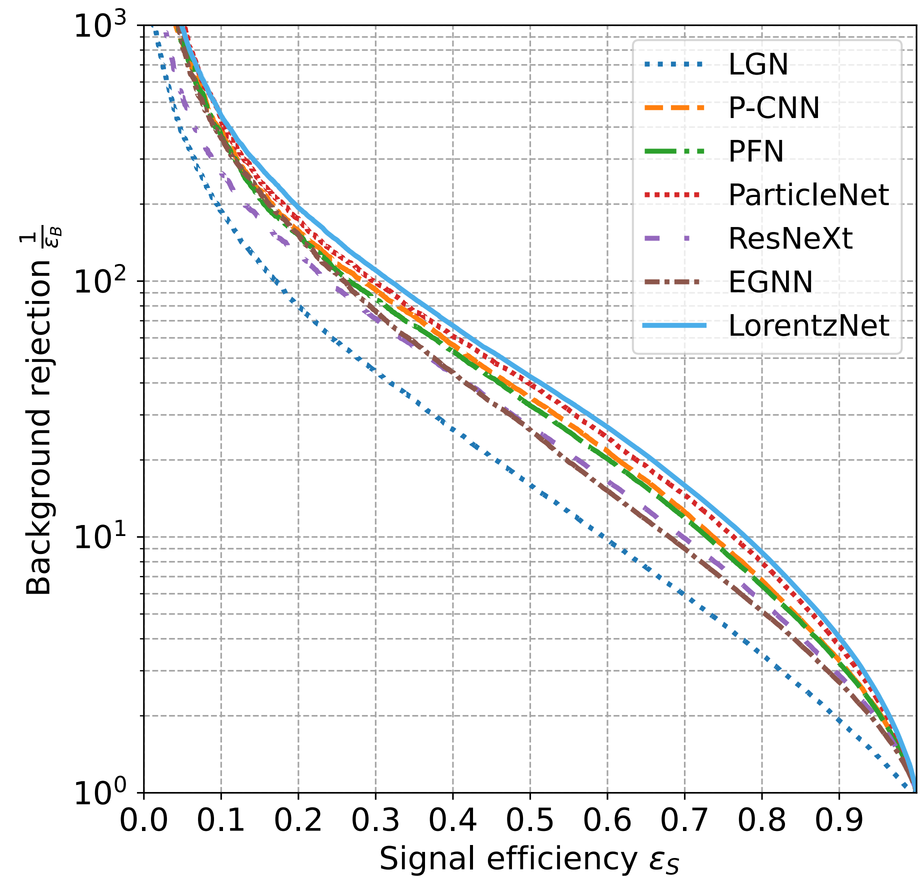
### *Top tagging benchmark* [SciPost Phys. 7 (2019) 014]

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




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

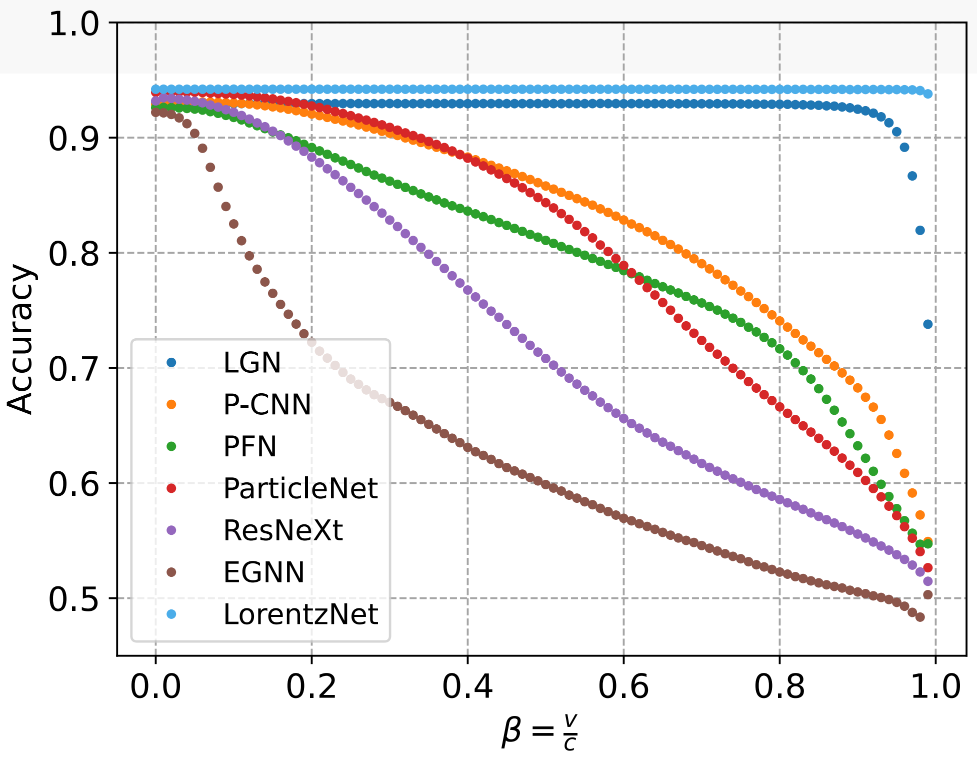
| Model | Accuracy | AUC | $1/\varepsilon_B$ ($\varepsilon_S = 0.5$) | $1/\varepsilon_B$ ($\varepsilon_S = 0.3$) |
|---|---|---|---|---|
| ResNeXt | 0.821 | 0.8960 | 30.9 | 80.8 |
| P-CNN | 0.827 | 0.9002 | 34.7 | 91.0 |
| PFN | – | 0.9005 | $34.7 \pm 0.4$ | – |
| ParticleNet | 0.840 | 0.9116 | $39.8 \pm 0.2$ | $98.6 \pm 1.3$ |
| EGNN | 0.803 | 0.8806 | $26.3 \pm 0.3$ | $76.6 \pm 0.5$ |
| 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 | $77 \pm 4$ | $199 \pm 14$ |
| | LorentzNet | **0.929** | **0.9793** | $\mathbf{176 \pm 14}$ | $\mathbf{562 \pm 72}$ |
| 1% | ParticleNet | 0.919 | 0.9734 | $103 \pm 5$ | $287 \pm 19$ |
| | LorentzNet | **0.932** | **0.9812** | $\mathbf{209 \pm 5}$ | $\mathbf{697 \pm 58}$ |
| 5% | ParticleNet | 0.931 | 0.9807 | $195 \pm 4$ | $609 \pm 35$ |
| | 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 | $290 \pm 30$ | $1105 \pm 59$ |
| LorentzNet | ✓ | **0.942** | **0.9868** | $\mathbf{498 \pm 18}$ | $\mathbf{2195 \pm 173}$ |

# Conclusion on LorentzNet

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
- code and model available in: *https://github.com/sdogsq/LorentzNet-release*

# Conclusion on LorentzNet

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
  - code and model available in: https://github.com/sdogsq/LorentzNet-release

- We would also like to ask
  - why LorentzNet outperforms many networks proposed after ParticleNet
  - can we dig deeper to extract the key component in LorentzNet?
    can it be applied to other networks as well?
- We will use experiments to confirm that one key component of the gain is Lorentz-symmetry preservation

# Part III: Lorentz-symmetric design

# LorentzNet performance on JetClass

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

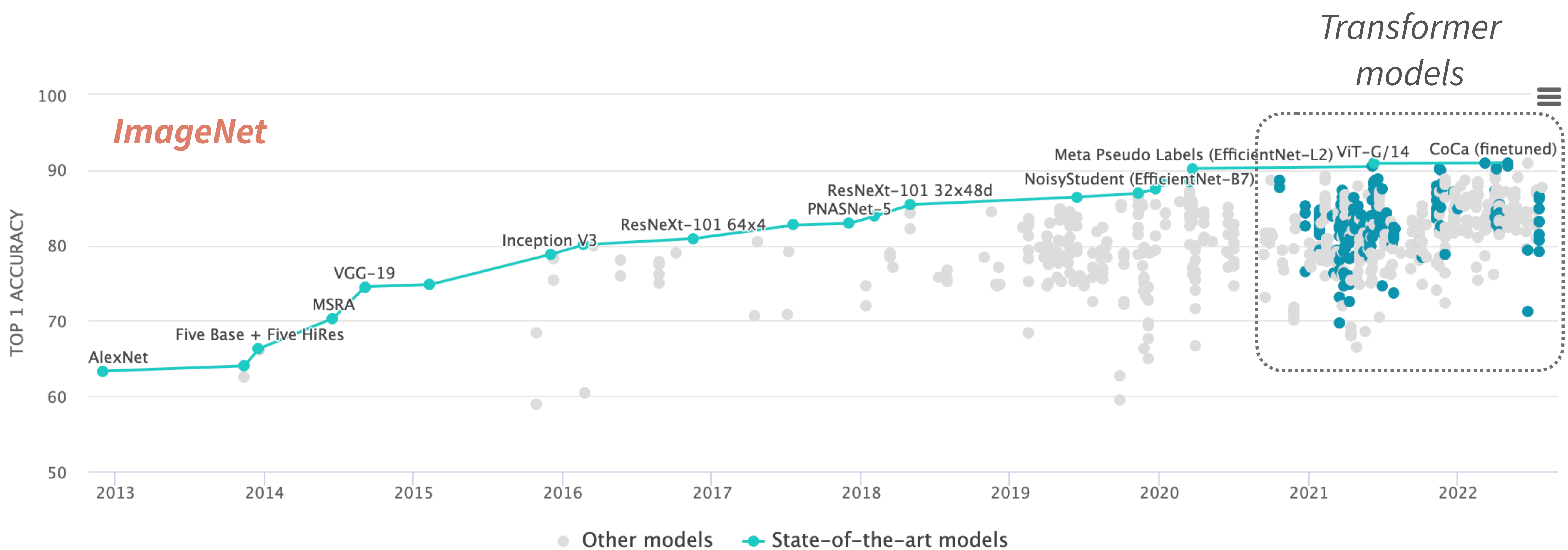
| | All classes | | $H \to b\bar{b}$ | $H \to c\bar{c}$ | $H \to gg$ | $H \to 4q$ | $H \to \ell\nu qq'$ | $t \to bqq'$ | $t \to b\ell\nu$ | $W \to qq'$ | $Z \to q\bar{q}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | AUC | $\text{Rej}_{50\%}$ | $\text{Rej}_{50\%}$ | $\text{Rej}_{50\%}$ | $\text{Rej}_{50\%}$ | $\text{Rej}_{99\%}$ | $\text{Rej}_{50\%}$ | $\text{Rej}_{99.5\%}$ | $\text{Rej}_{50\%}$ | $\text{Rej}_{50\%}$ |
| 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 |
| 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 |

- ➔ A new benchmark, *JetClass* is proposed in arXiv:2202.03772, consists of 100M jets, ~100x larger then previous benchmarks
- ➔ *LorentzNet* performs much better than *ParticleNet*, slightly worse than the most advanced model: *ParT* (discuss in p.51)
  - ❖ note that for #params LorentzNet < ParticleNet, and << ParT
  - ❖ this prove that **LorentzNet is a very efficient model**
- ➔ We may want to understand the core of such efficiency

| | 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"

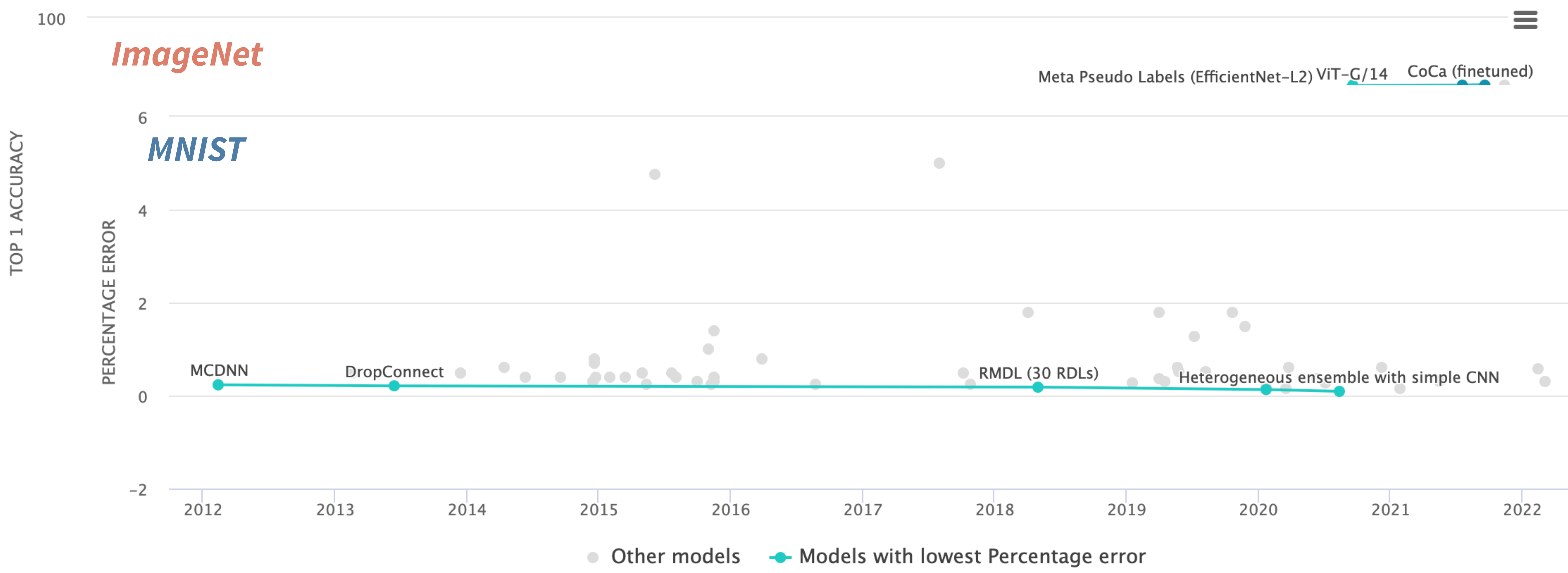
- ➔ Lessons from image classification from Computer Vision
  - ❖ Training on ImageNet and its extension (224x224 pixel image classification)
    - Transformer models have led the performance, since the first application in 2020




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

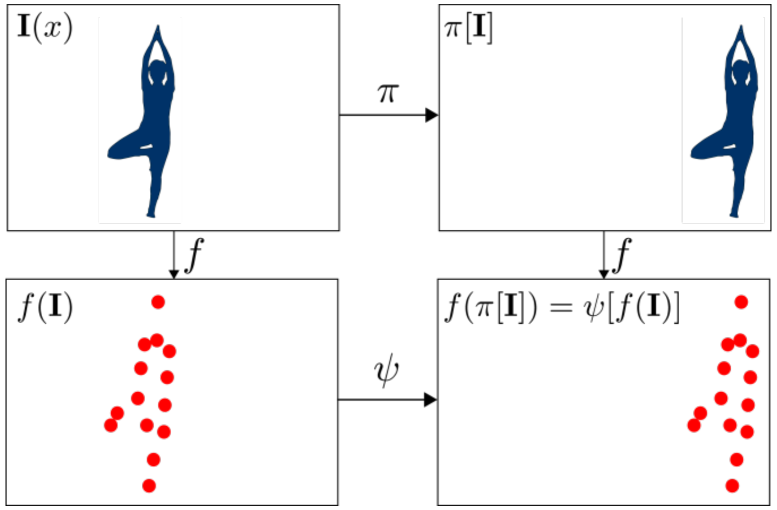
# Model, data size and “inductive bias”

- 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



# Interpret Lorentz-symmetry as an inductive bias

→ Goal of our new study:

- we want to confirm that Lorentz-symmetry preservation the "inductive bias" for jet physics to boost the network performance
- even better if we isolate "a patch" from LorentzNet, which can be applied to a wider range of networks

→ Our experiments

- devise multiple choices of additional 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*

# Interpret Lorentz-symmetry as an inductive bias

- Goal of our new study:
  - we want to confirm that Lorentz-symmetry preservation the "inductive bias" for jet physics to boost the network performance
  - even better if we isolate "a patch" from LorentzNet, which can be applied to a wider range of networks
- Our experiments
  - devise multiple choices of additional features, which are ***invariant to some or all Lorentz transformations***
  - want to see if this affects network performance as we expect

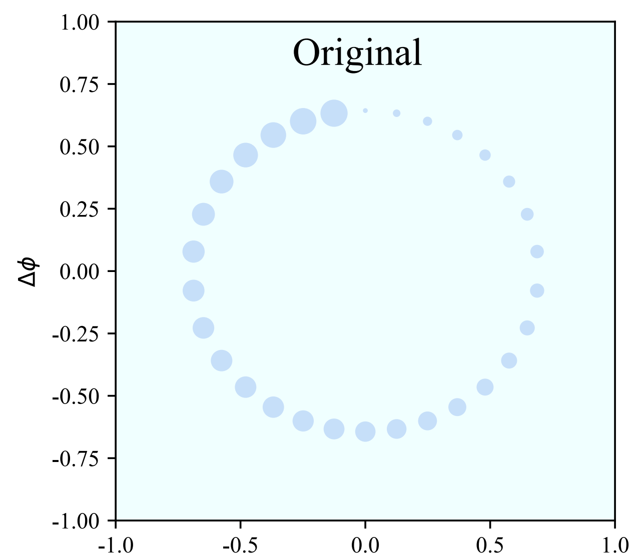
*First of all, we need to find a good way to categorize the possible Lorentz transformations acted on the jet*

C.Li *et al.* arXiv:2208.07814

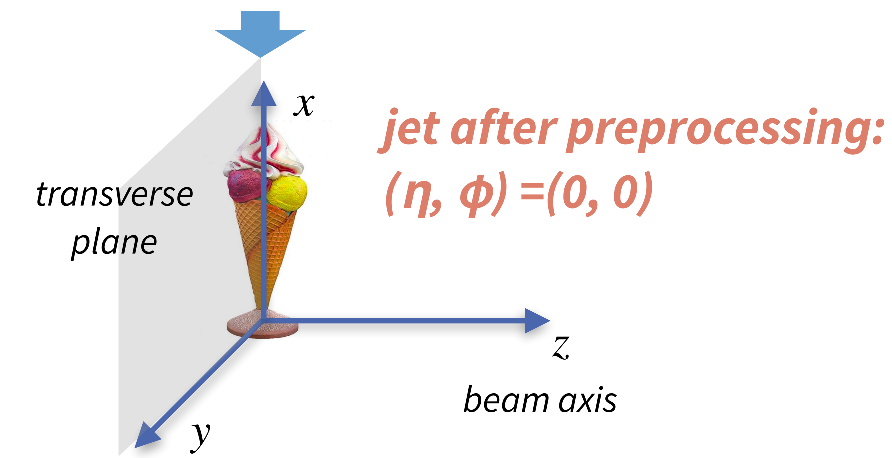
# Lorentz transformations

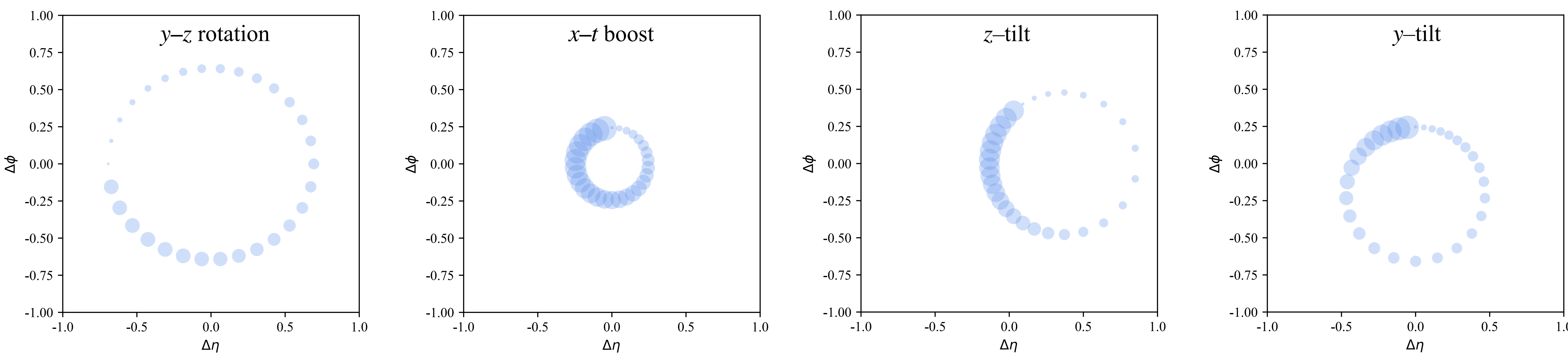
➔ By HEP convention, a jet is represented on Δη-Δϕ plane w.r.t. its axis

- equivalent as: apply a boost on *z*-axis → then a rotation on *x*-*y* plane (transverse plane) → now jet points to the *x*-axis
- after the conventional preprocessing, we have ***four additional DoFs*** for Lorentz transformation!




**≈ *Viewing the jet from x-direction***




≈ $\eta$-$\phi$ rotation

$z$-boost + $x$-$z$ rotation

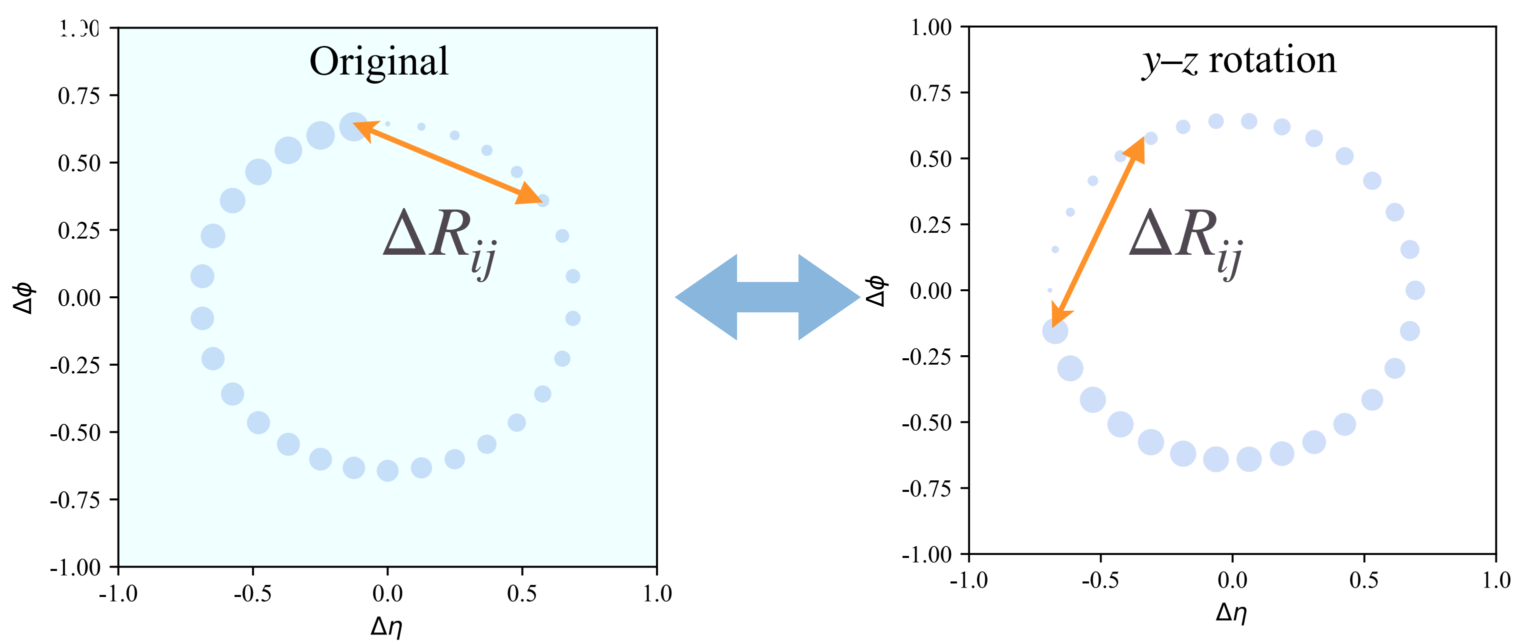
$y$-boost + $x$-$y$ rotation

# 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 ($y_{y,z} \sim o(1)$) |
|---|---|---|---|---|---|---|
| $m_{ij}^2$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $\Delta R_{ij}$ | ✓ | ✓ | ✓ | | | |
| $\Delta R_{ij}(p_{\mathrm{T},i} + p_{\mathrm{T},j})$ | ✓ | ✓ | ✓ | ✓ | | |
| $E_{ij}$ (ablation study) | | ✓ | ✓ | | | |

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

- pairwise mass: invariant under all transformations
- pairwise $\Delta R_{ij}$: approx. invariant under y-z rotation (≈ η-ϕ rotation)
- 

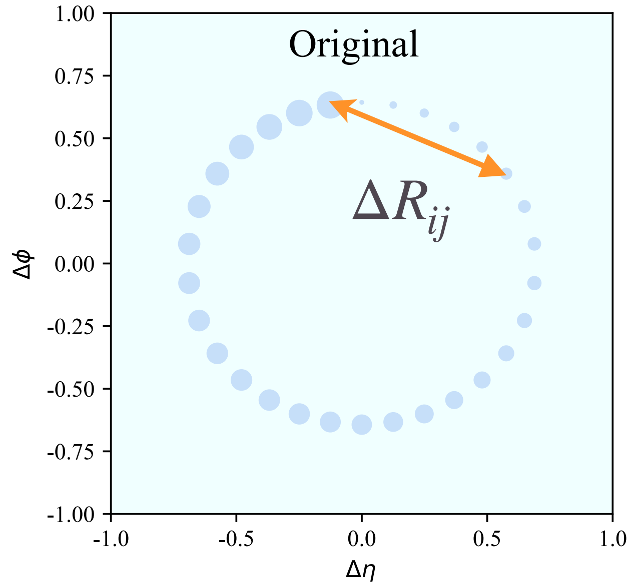


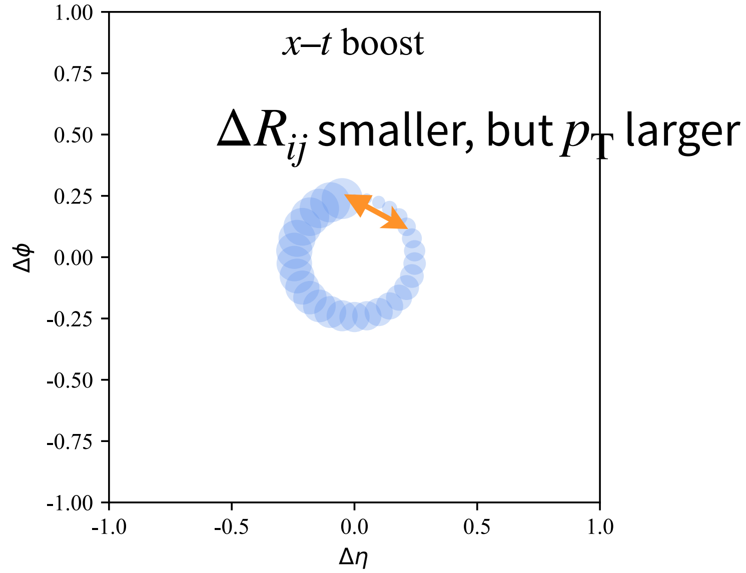
# Variable design

| 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 ($y_{y,z} \sim o(1)$) |
|---|---|---|---|---|---|---|
| $m_{ij}^2$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $\Delta R_{ij}$ | ✓ | ✓ | ✓ | | | |
| $\Delta R_{ij}(p_{\mathrm{T},i} + p_{\mathrm{T},j})$ | ✓ | ✓ | ✓ | ✓ | | |
| $E_{ij}$ (ablation study) | | ✓ | ✓ | | | |

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

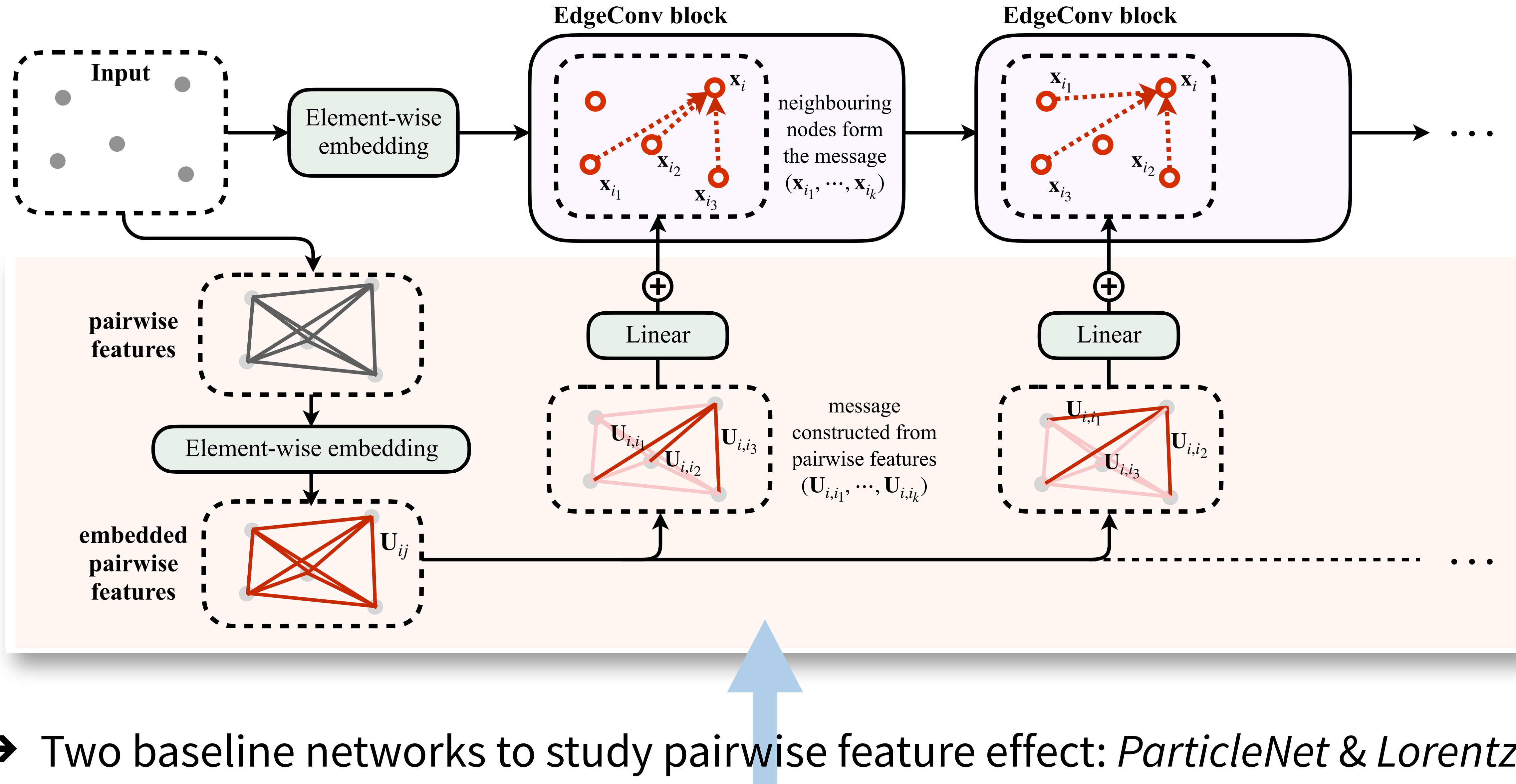
- pairwise mass: invariant under all transformations
- pairwise $\Delta R_{ij}$: approx. invariant under y-z rotation (≈ η-ϕ rotation)
- manually construct variable $\Delta R_{ij}(p_{\mathrm{T},i} + p_{\mathrm{T},j})$: can prove that it is also approx. invariant under x-boost



# 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 *k*NN 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)

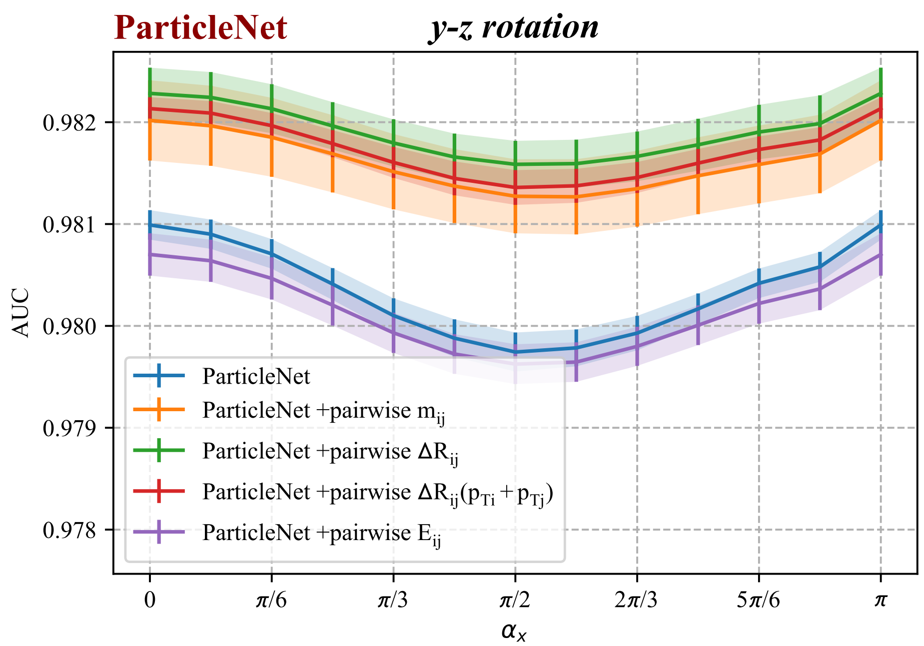
# Performance for adding pairwise features

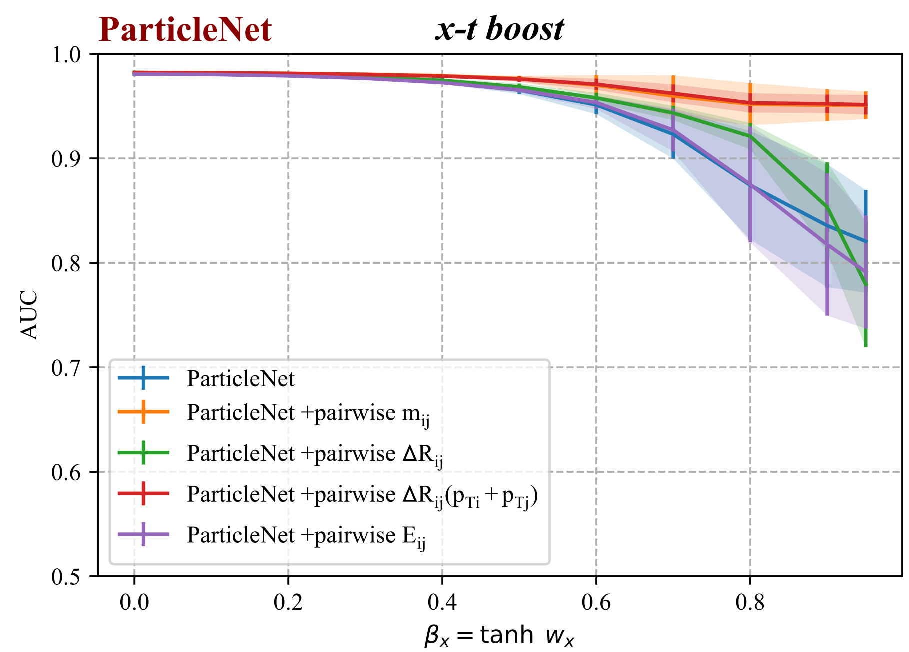
***Training on 60k top tagging dataset*** (smaller dataset manifest the power of inductive bias)

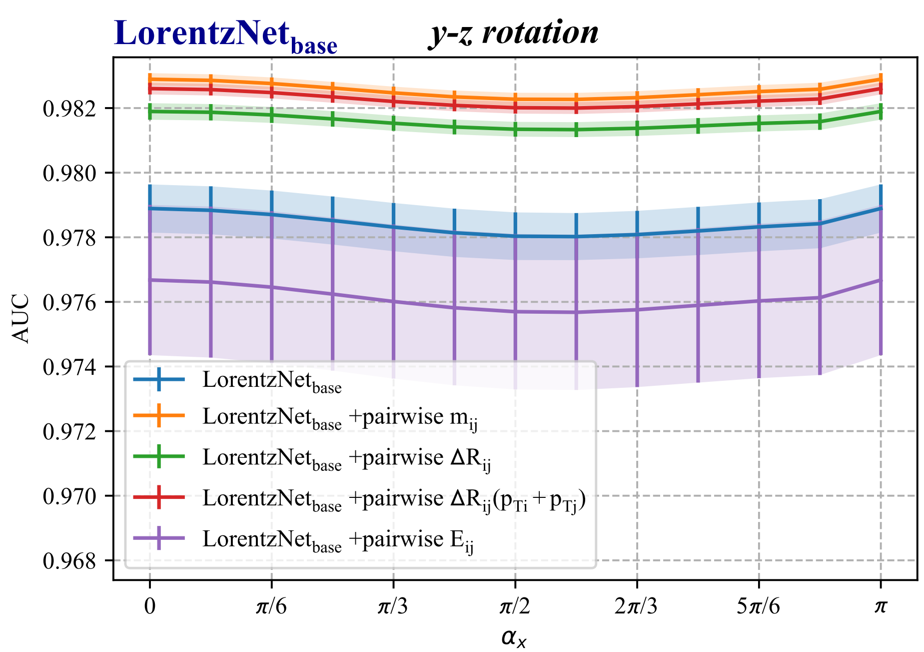
| Base model | Variation | Accuracy | AUC | $1/\epsilon_B$ ($\epsilon_S = 50\%$) | $1/\epsilon_B$ ($\epsilon_S = 30\%$) |
|---|---|---|---|---|---|
| ParticleNet | — | 0.9310(3) | 0.9810(2) | $198 \pm 7$ | $640 \pm 29$ |
| | +pairwise: $m_{ij}$ | 0.9334(8) | 0.9820(4) | $222 \pm 13$ | $722 \pm 52$ |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

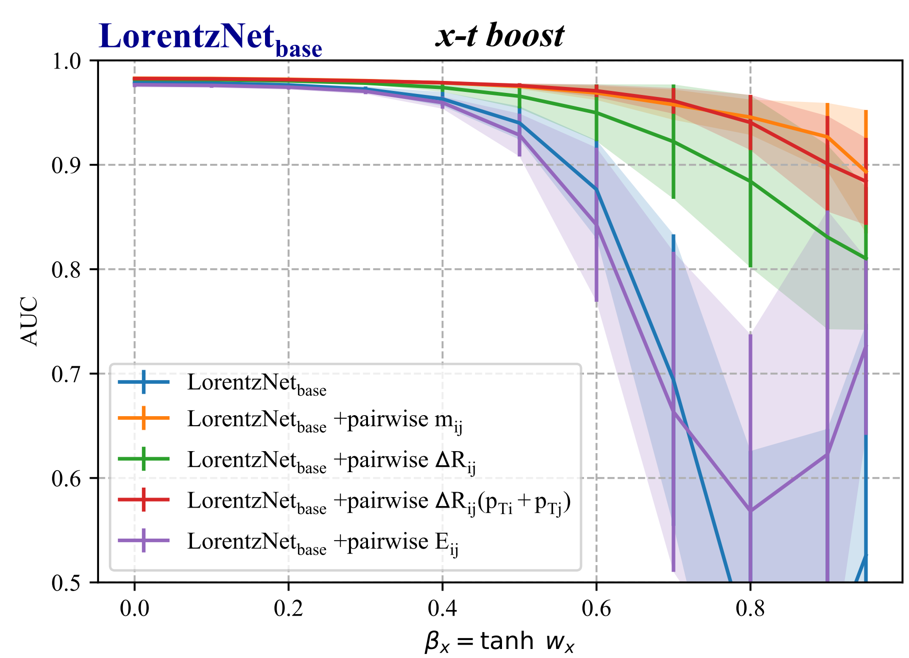
***better compared to baselines***

# Performance for adding pairwise features



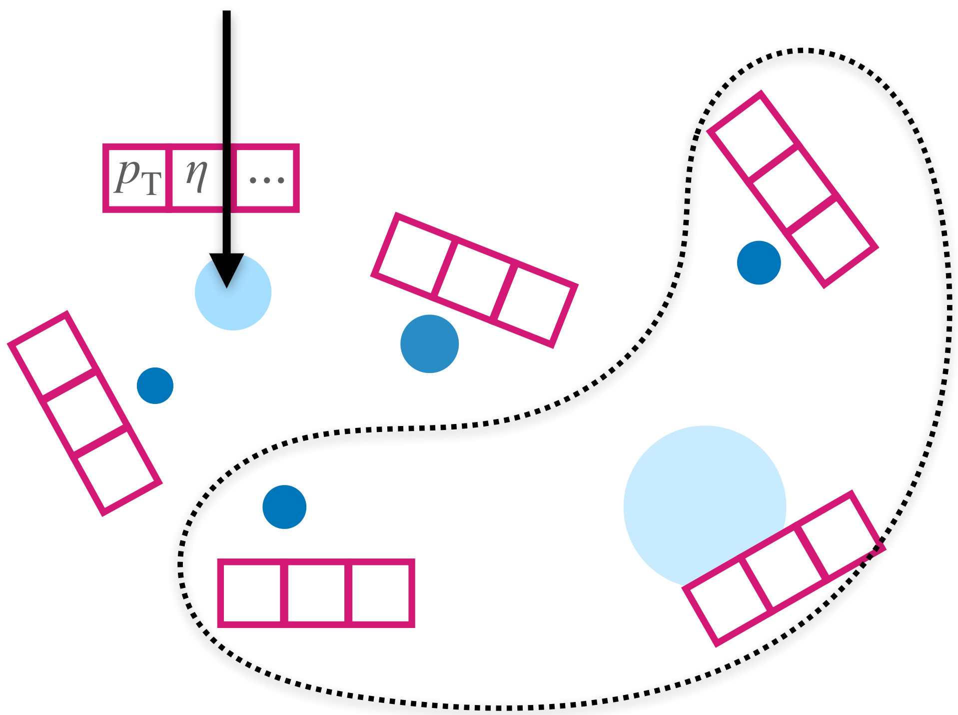




- ***Injecting ΔR to the network*** *→ more robust to y-z rotation*
- ***Injecting $\Delta R(p_{Ti}+p_{Tj})$ or mass*** *→ more robust to y-z rotation and now also the x-boost*

# A step towards a general solution

- ➔ Pairwise features have limitations
  - ❖ only applicable to GNN networks which intrinsically build “edges”
- ➔ Upgrade to node-wise features
  - ❖ “mass features” carried per node, not edge between nodes

*(1) for each node $i$*




*(2) find a* ***friend group*** $G_i$*: composed of k nodes* $i_m\,(m=1,\cdots,k)$ *having largest* $p_i^\mu p_{i_m\mu}$

this is a Lorentz invariant choice

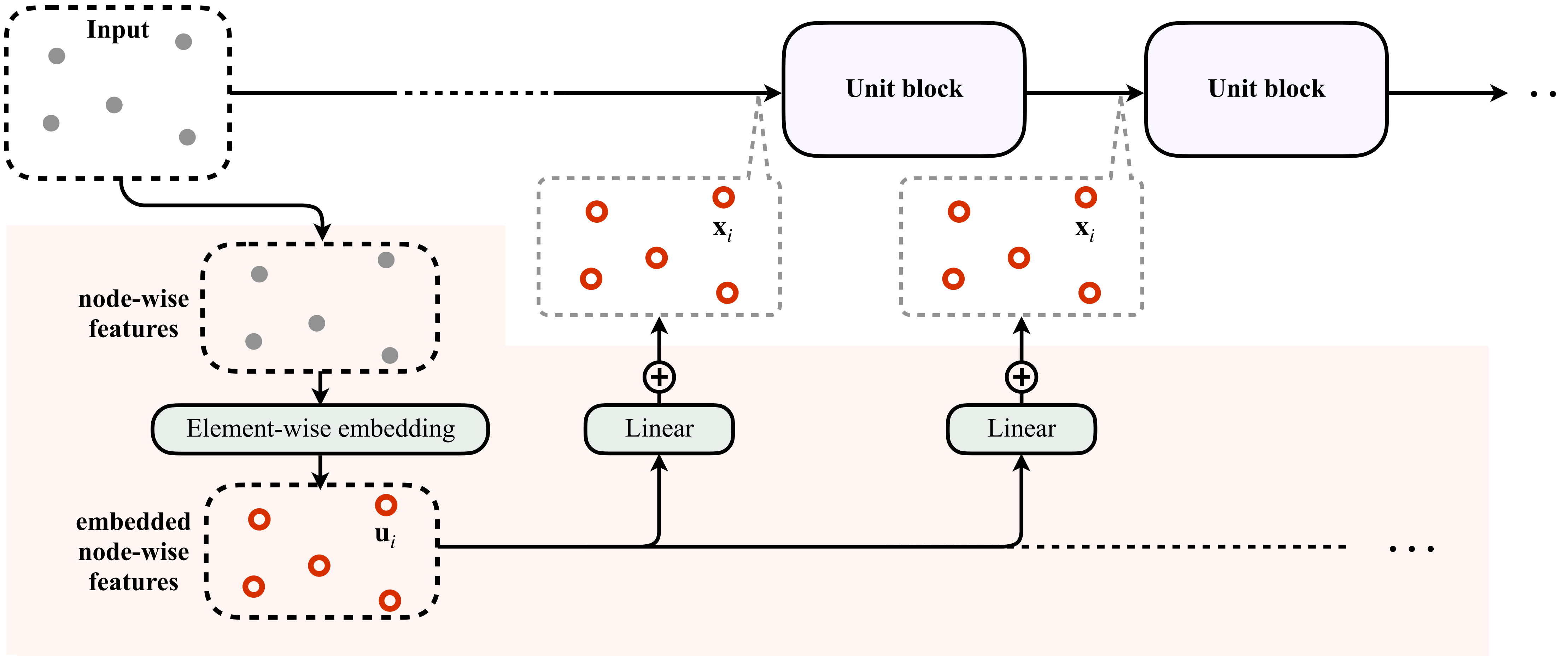
*(3) calculate mass*

$$m_{G_i}^2 = \Big(\sum_{j\in G_i} p_j\Big)^2 \approx 2\sum_{j,k\in G_i}^{j<k} p_j^\mu p_{k\mu}$$

is essentially the pre-determined linear combination of all pairwise masses

# General patch structure design for node-wise features

***Any baseline network***




➔ Baseline networks can be any network that treats jet as a point cloud

➔ Integrate new node-wise features layer-by-layer

- unit block is $\Phi(x)$ function for PFN, EdgeConv for ParticleNet, and LEGB for LorentzNet

# Performance for adding node-wise features

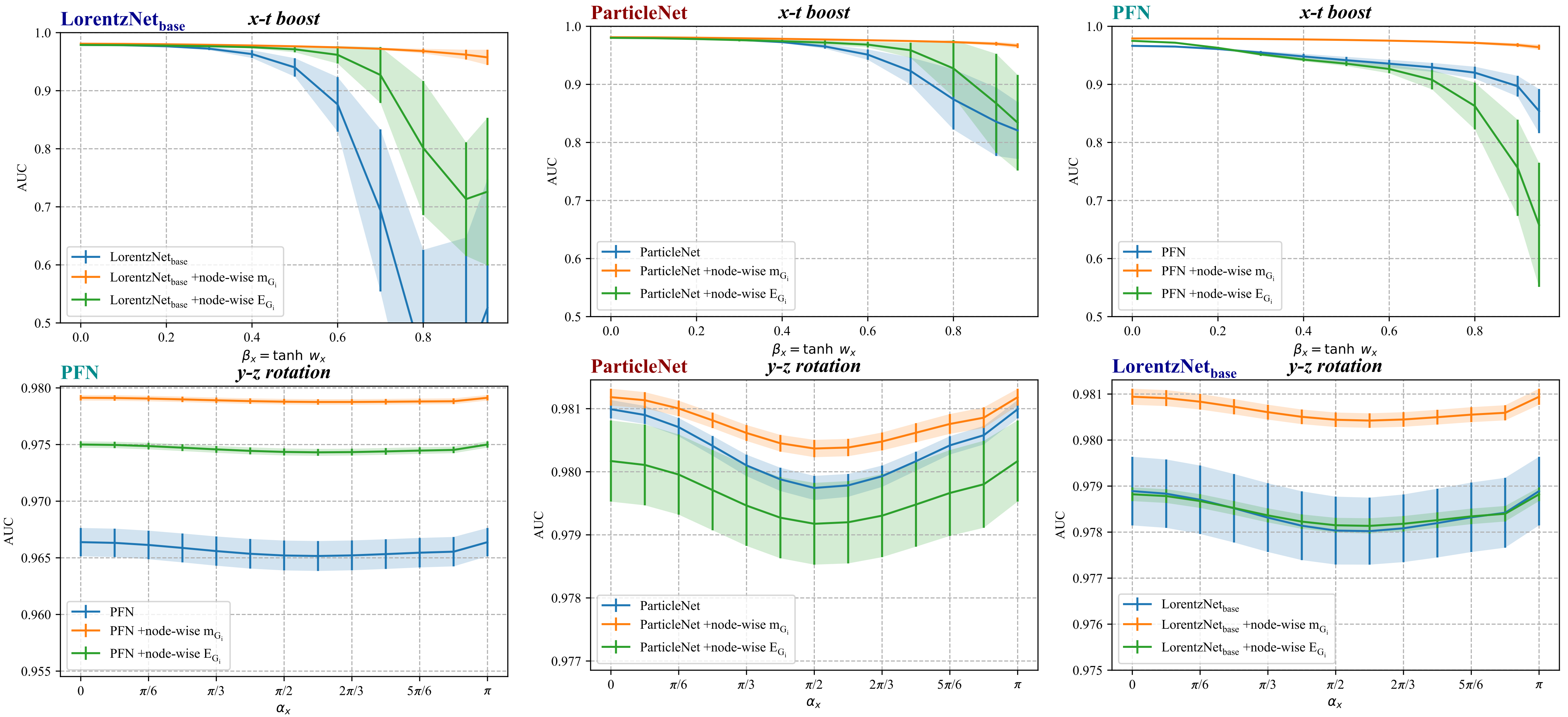
| Base model | Variation | Accuracy | AUC | $1/\epsilon_B$ ($\epsilon_S = 50\%$) | $1/\epsilon_B$ ($\epsilon_S = 30\%$) |
|---|---|---|---|---|---|
| PFN | — | 0.9104(12) | 0.9664(13) | $67 \pm 5$ | $198 \pm 21$ |
| | +node-wise: $m_{G_i}$ | **0.9281(4)** | **0.9791(2)** | $\mathbf{184 \pm 5}$ | $\mathbf{714 \pm 50}$ |
| | +node-wise: $E_{G_i}$ | 0.9207(4) | 0.9750(3) | $125 \pm 3$ | $378 \pm 19$ |
| ParticleNet | — | 0.9310(3) | 0.9810(2) | $198 \pm 7$ | $640 \pm 29$ |
| | +node-wise: $m_{G_i}$ | **0.9313(3)** | **0.9812(1)** | $\mathbf{222 \pm 5}$ | $\mathbf{800 \pm 40}$ |
| | +node-wise: $E_{G_i}$ | 0.9300(12) | 0.9802(6) | $183 \pm 12$ | $572 \pm 47$ |
| LorentzNet$_{\text{base}}$ | — | 0.9276(12) | 0.9789(7) | $172 \pm 13$ | $581 \pm 53$ |
| | +node-wise: $m_{G_i}$ | **0.9306(3)** | **0.9809(2)** | $\mathbf{219 \pm 3}$ | $\mathbf{887 \pm 36}$ |
| | +node-wise: $E_{G_i}$ | 0.9272(3) | 0.9788(1) | $171 \pm 2$ | $562 \pm 16$ |

*Adding node-wise mass:*

*(1) improve network performance (especially for PFN)*

*(2) more robust to Lorentz transformations on test data*

*(3) smaller error bars (illustrate more generalization ability)*



# Performance summary

→ What do the above results mean?

- the full network tends to be ***more robust and performant***, when we incorporate Lorentz-symmetry-preserved variables (pairwise/node-wise ones) into the network
- even when we ***introduce a very small patch structure*** invariant under a certain symmetry (the original network is unaffected) helps the network to perform better
  - without need to let the network fully satisfy Lorentz symmetries
  - 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 summary

→ What do the above results mean?

- the full network tends to be ***more robust and performant***, when we incorporate Lorentz-symmetry-preserved variables (pairwise/node-wise ones) into the network
- even when we ***introduce a very small patch structure*** invariant under a certain symmetry (the original network is unaffected) helps the network to perform better
  - without need to let the network fully satisfy Lorentz symmetries
  - invariance property of the small sub- 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 |

- The experiments show that “**pairwise mass**” is the key component in network design
- **We reveal that the underlying logic lies in the Lorentz symmetry preservation**
- We make a successful attempt to understand the interpretability of the network in terms of symmetry preservation

# Part IV: Outlook & Summary

# Discussions

1. We provide two general solutions to improve neural network performance
   - ❖ incorporate pairwise/node-wise features
   - ❖ the node-wise solution is more generalized to be applied
   - ❖ yet "pairwise mass" is still crucial ***if one hopes to achieve state-of-the-art performance*** (as they form a full set of Lorentz scalar basis)
2. We address that the Lorentz-symmetric design ***is already used in the current best models***
   - ❖ *LorentzNet* and *ParT* (discussion in p.51) both inject "pairwise mass" in network design
   - ❖ can explain to some extent their high performance

# Brief intro to ParT

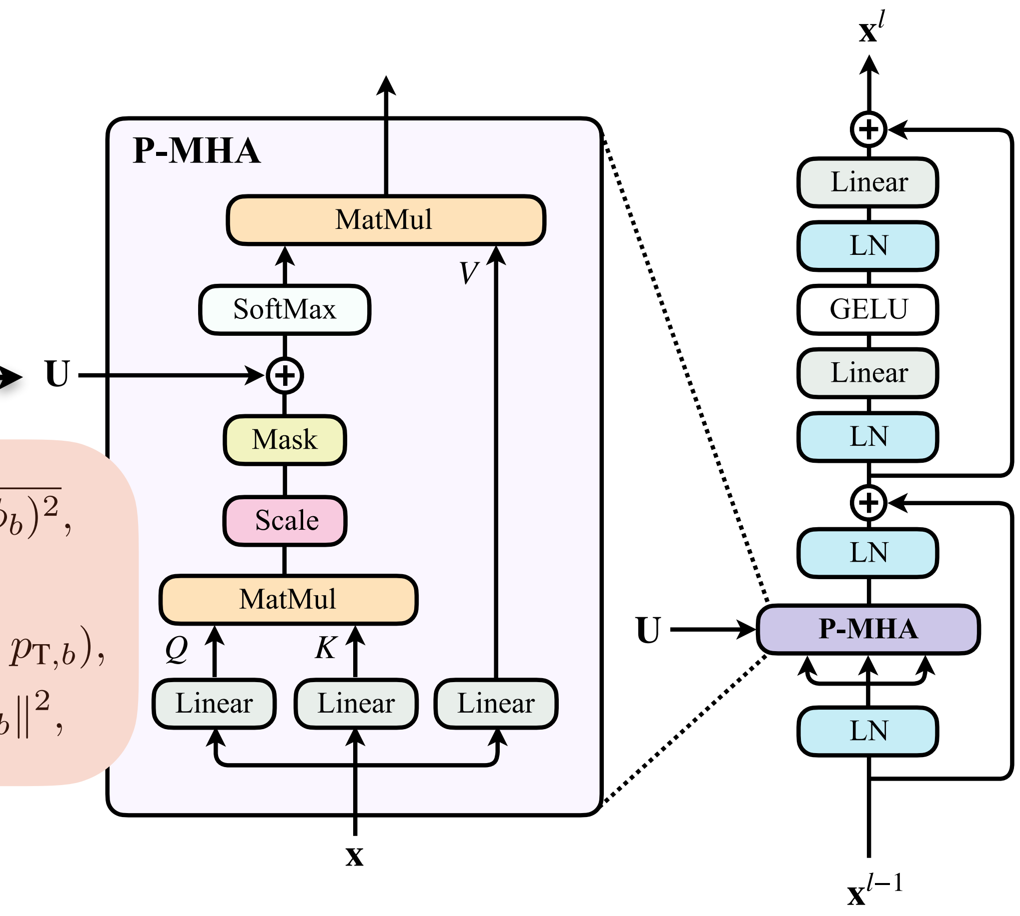
- ➔ Transformer model is the new state-of-the-art architecture introduced in DL community
  - ❖ Language models: BERT, GPT-3…
  - ❖ Computer Vision: ViT, Swin-T
  - ❖ AI for Science: AlphaFold2 for protein structure prediction
- ➔ Transformers architecture
  - ❖ consists only of self-attention blocks
  - ❖ more scalable with large model/data
  - ❖ big model (more parameters) + more training data + affordable computing complexity → better performance

# Brief intro to ParT

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

| | All classes Accuracy | AUC | $H \to b\bar{b}$ Rej$_{50\%}$ | $H \to c\bar{c}$ Rej$_{50\%}$ | $H \to gg$ Rej$_{50\%}$ | $H \to 4q$ Rej$_{50\%}$ | $H \to \ell\nu qq'$ Rej$_{99\%}$ | $t \to bqq'$ Rej$_{50\%}$ | $t \to b\ell\nu$ Rej$_{99.5\%}$ | $W \to qq'$ Rej$_{50\%}$ | $Z \to q\bar{q}$ 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 |
| 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 |

$$\Delta = \sqrt{(y_a - y_b)^2 + (\phi_a - \phi_b)^2},$$
$$k_{\mathrm{T}} = \min(p_{\mathrm{T},a}, p_{\mathrm{T},b})\Delta,$$
$$z = \min(p_{\mathrm{T},a}, p_{\mathrm{T},b})/(p_{\mathrm{T},a} + p_{\mathrm{T},b}),$$
$$m^2 = (E_a + E_b)^2 - \|\mathbf{p}_a + \mathbf{p}_b\|^2,$$




**(b) Particle Attention Block**

| | 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 |

*similar computation complexity with ParticleNet, but more performant than ParticleNet and LorentzNet!*

# Intro to published tool: `Weaver`

- Introducing **`Weaver`**, a streamlined and flexible machine learning R&D framework for HEP applications
- use the below link to
  - try out ***ParT***, ***ParticleNet*** model out-of-the-box
  - play with the ***JetClass*** dataset
    - we invite the community to explore and experiment with this dataset and extend the boundary of deep learning and jet physics even further.
  - or explore previous top tagging & quark/gluon dataset, or any custom ones

[https://github.com/jet-universe/particle_transformer](https://github.com/jet-universe/particle_transformer)

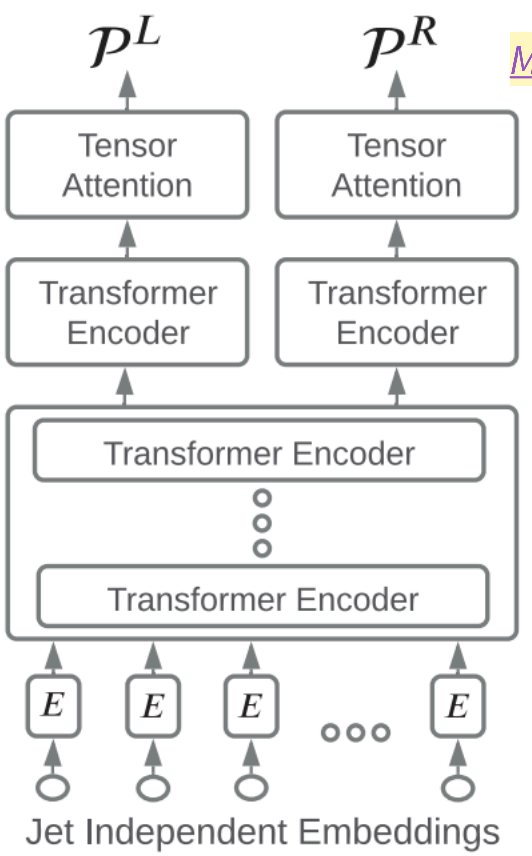
# Discussions

3. Lorentz-symmetric design (incorporating pairwise mass) can be adopted to other scenarios

### *(1) Jet tasks beyond jet tagging*

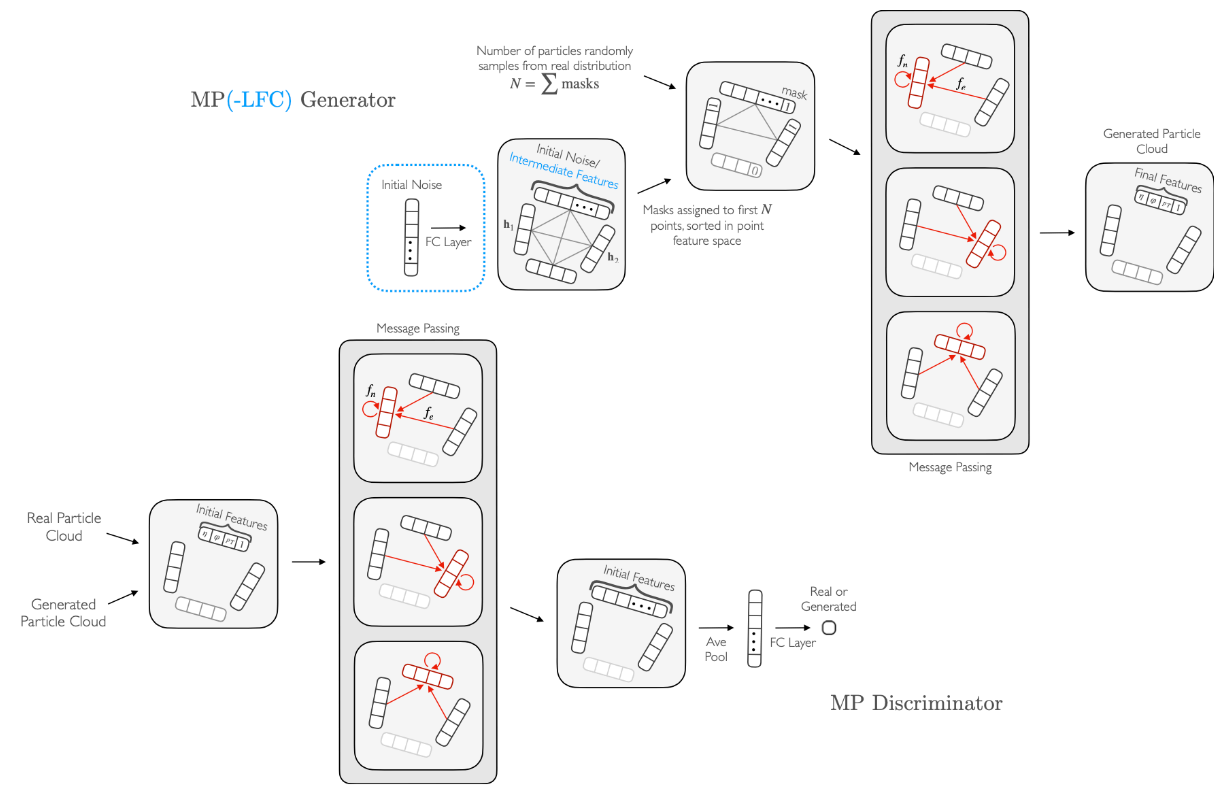
***SPA-Net for jet assignments***

M.Fenton et al. PRD 105, 11200




***MPGAN for generation of jets***

R.Kansal et al. arXiv:2106.11535, Proceeding of 35th NIPS



# Discussions

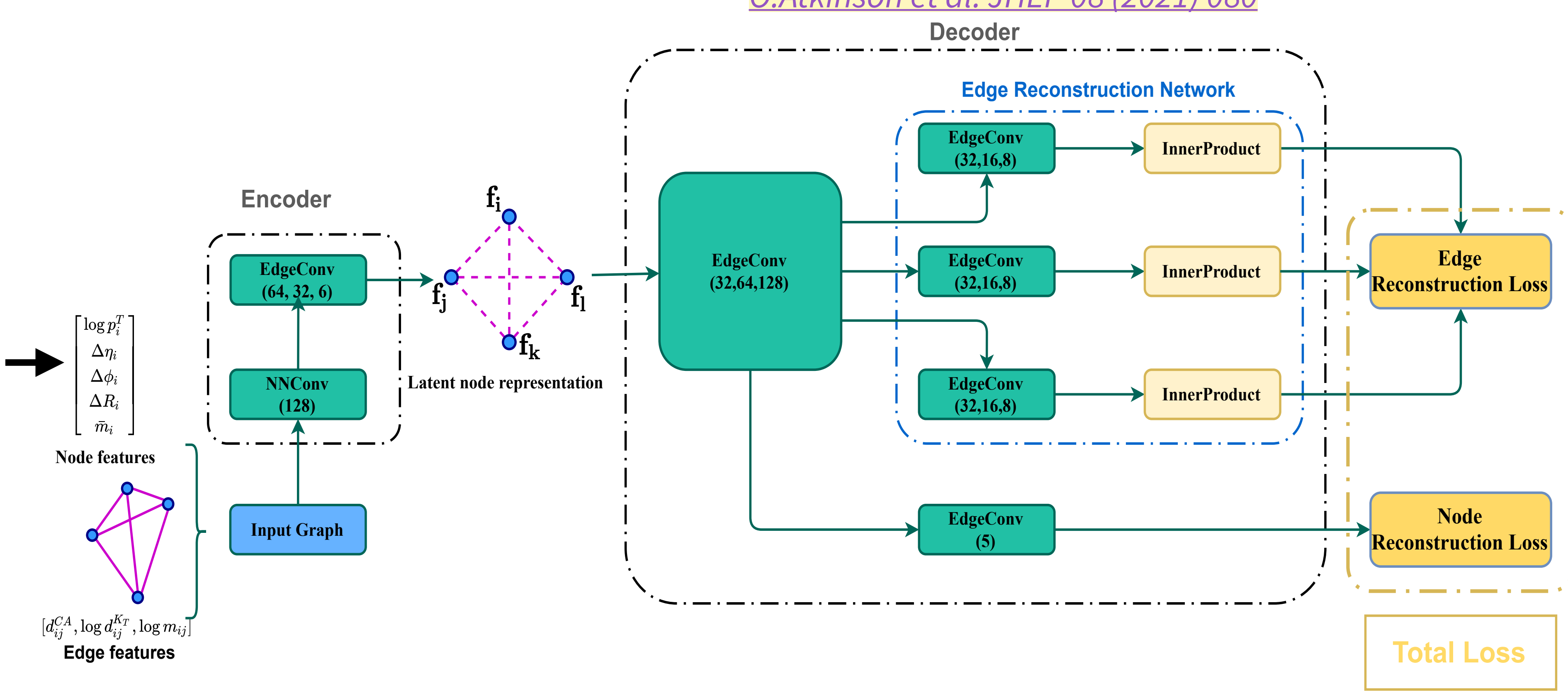
3. Lorentz-symmetric design (incorporating pairwise mass) can be adopted to other scenarios

### *(2) Tasks to process whole collision event*

***Convolutional GNN autoencoder for anomalous detection***

[O.Atkinson et al. JHEP 08 (2021) 080](#)

Recluster R=1.5 jets to R=0.1 microjets and use as inputs



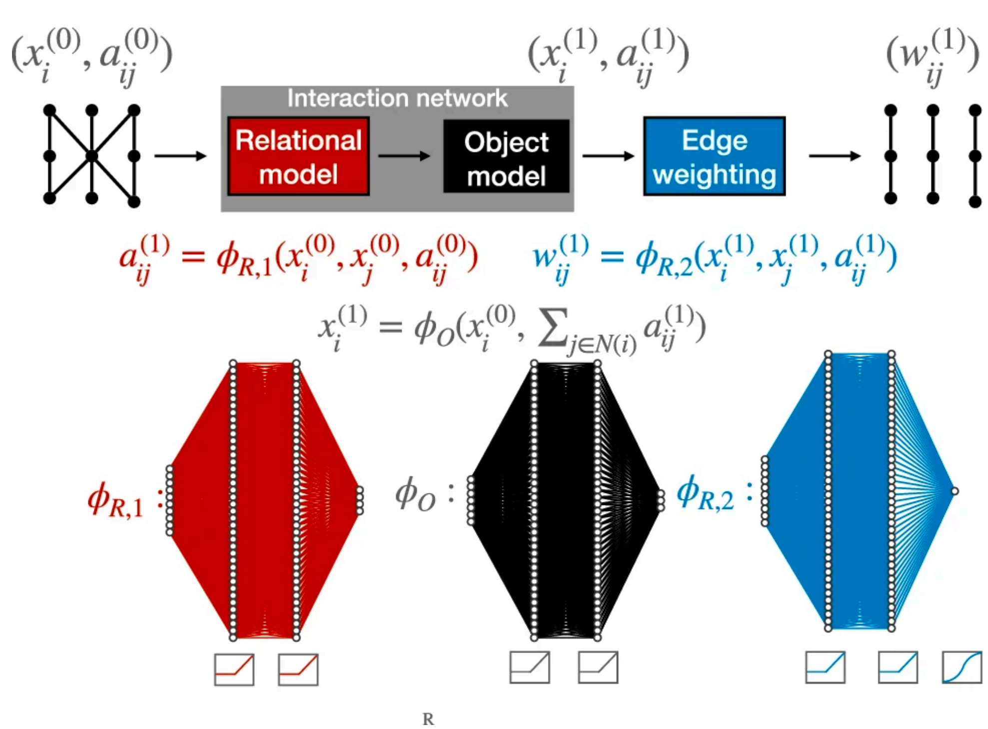
# Discussions

3. Lorentz-symmetric design (incorporating pairwise mass) can be adopted to other scenarios

### *(3) Tasks using primitive data source*

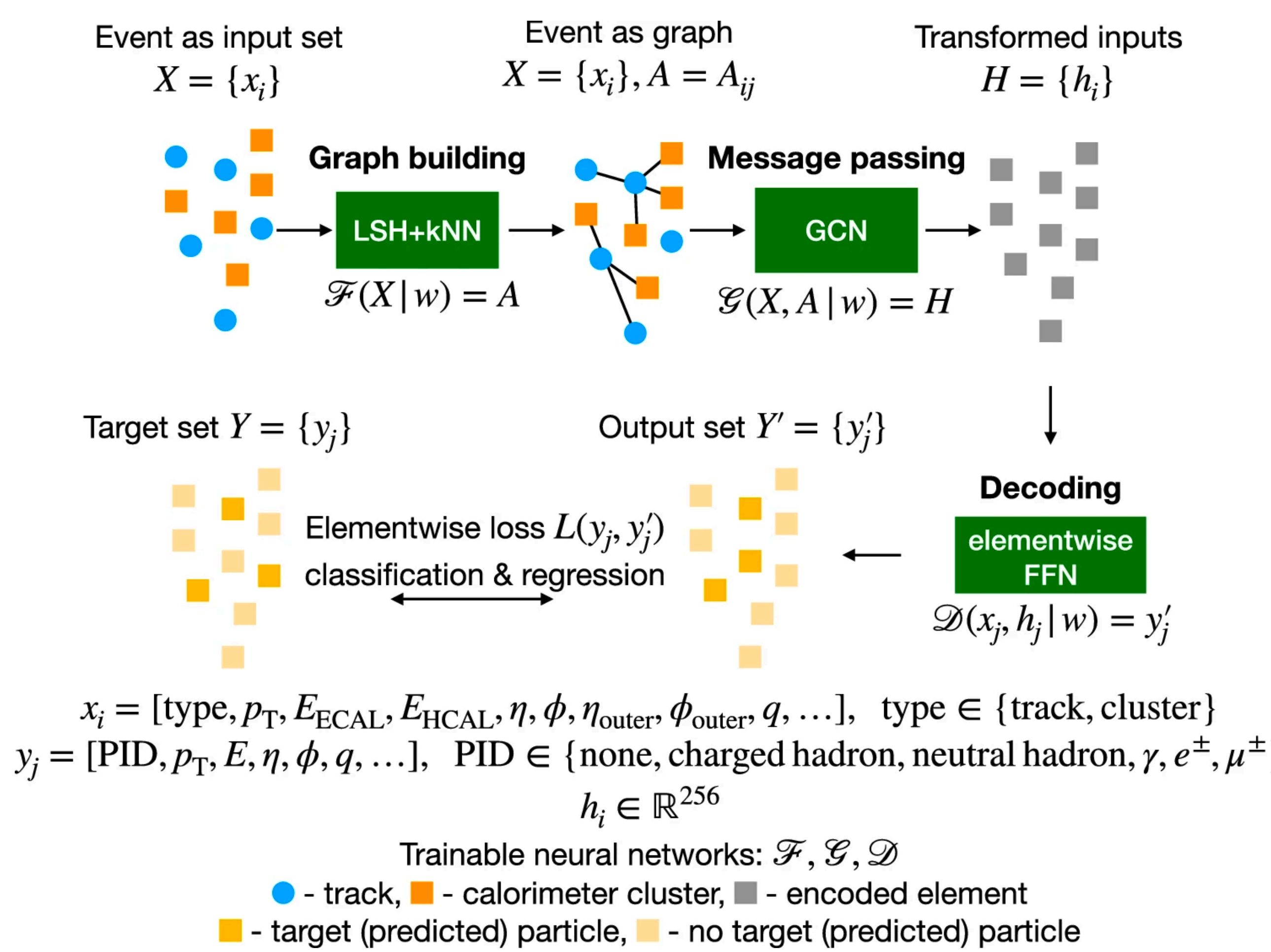
### *GNN for track reconstruction*

G.DeZoort et al. Comput. Softw. Big Sci. 5, 26 (2021)




### *MLPF, using tracks and clusters to reconstruct particle-flow candidates*

J.Pata et al. EPJC 81, 381 (2021)



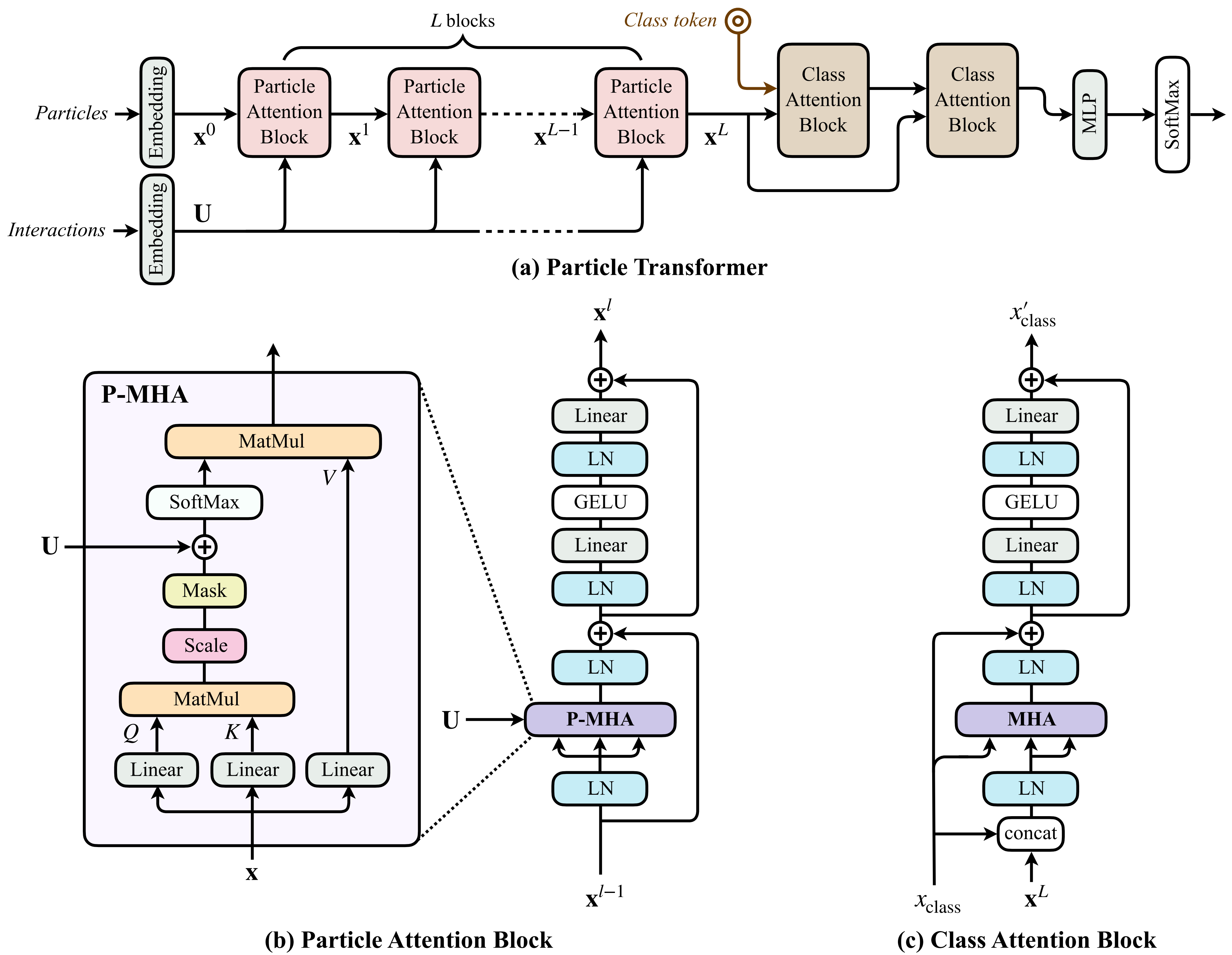
# Overall summary

- In this talk:
  - we recap the evolution of DL application to jet tagging
  - we introduce LorentzNet, a GNN-based network respecting full Lorentz symmetry, which exhibits better performance than previous state-of-the-arts
  - we investigate the core of such enhancement, and discover the role “Lorentz symmetry preservation” plays in networks; we propose two patch structures applicable to a variety of baselines
- Hints to interesting new applications
  - the Lorentz symmetry design as an intrinsic inductive bias in jet physics has a wider range of potential applications

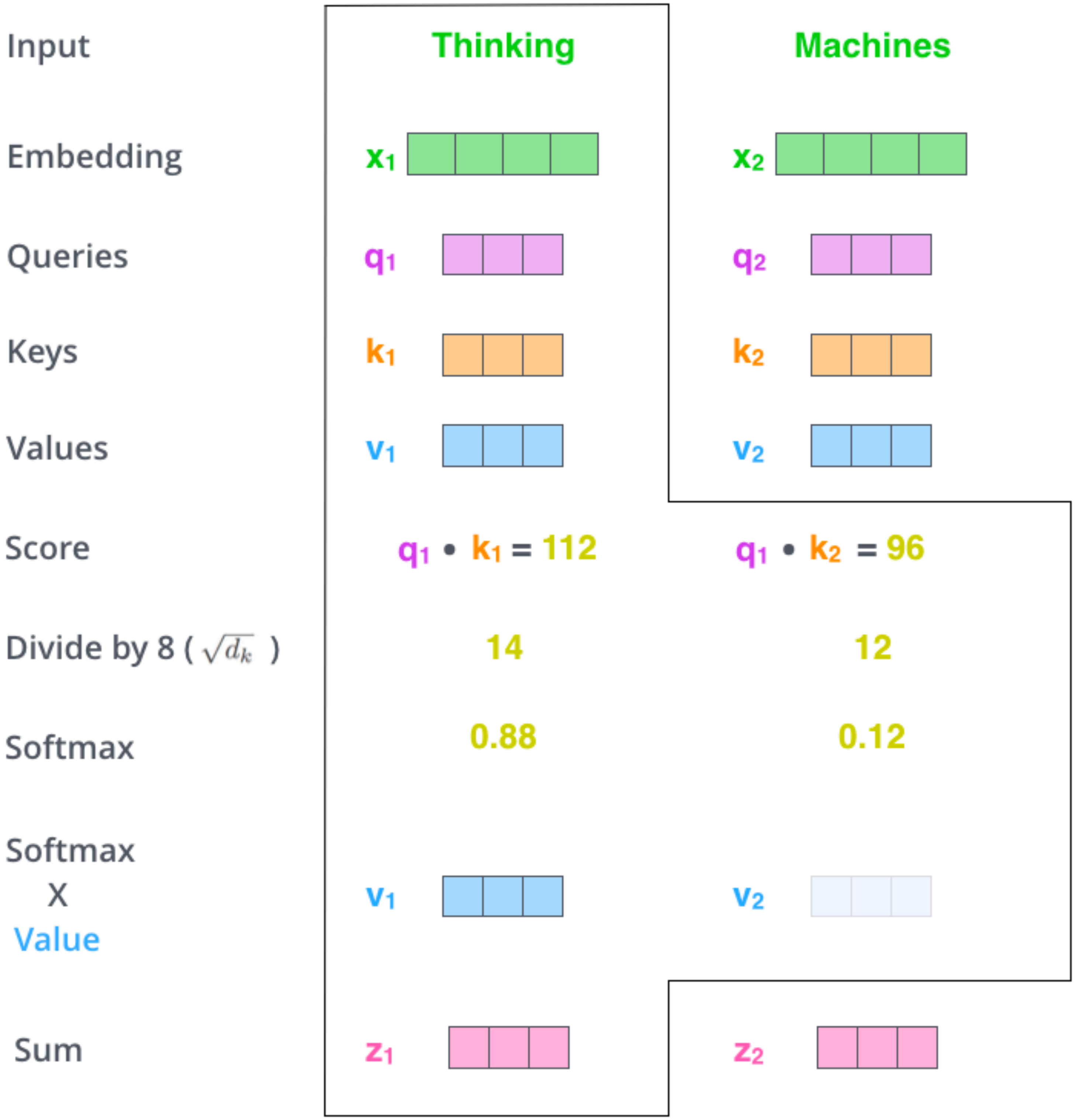
# Backup

# ParT architecture

*H.Qu et al. arXiv:2202.03772, proceedings of 39th ICML, Vol.162*



# Transformer illustration




[image from link]