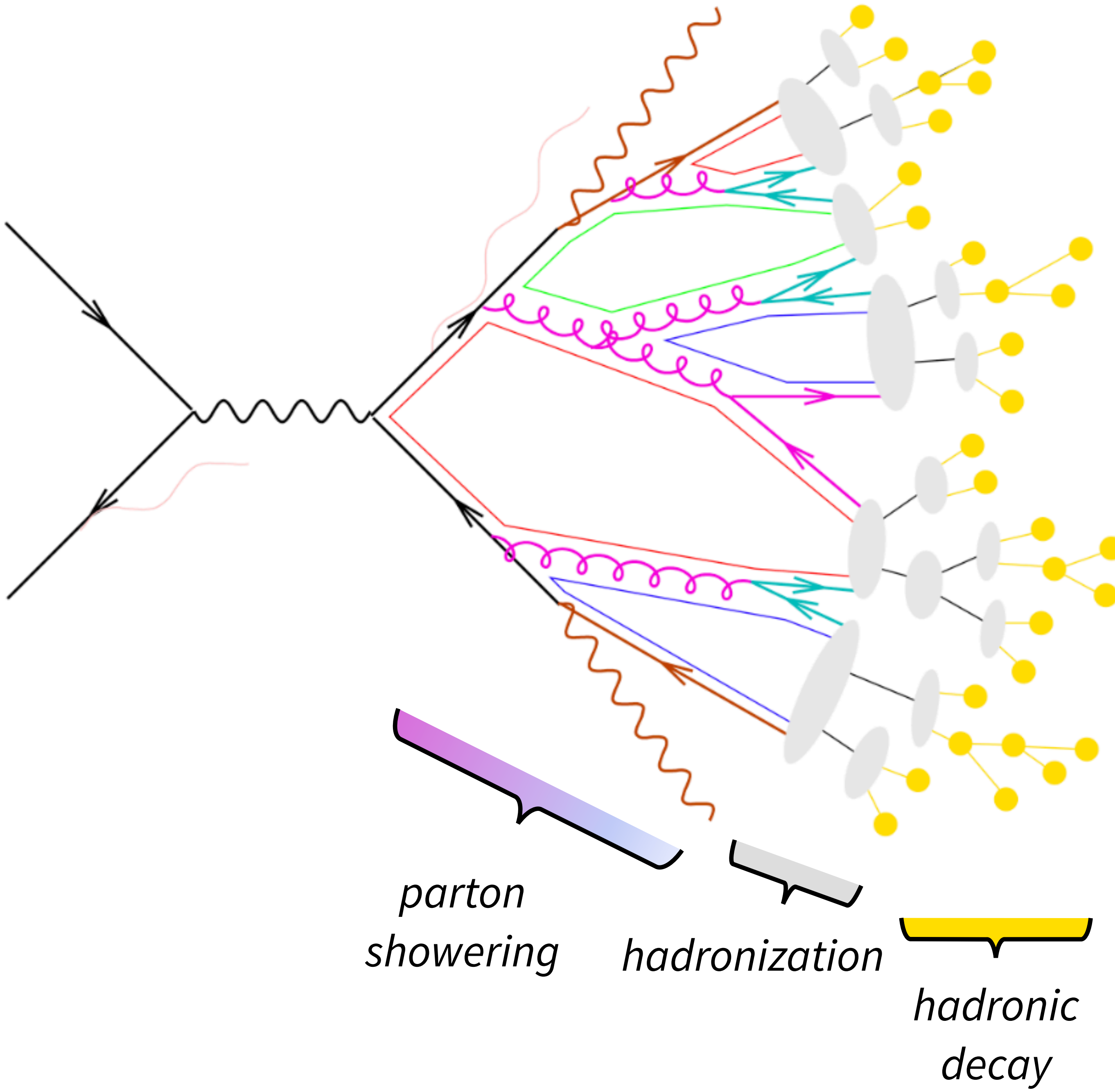
Cutting-edge jet networks and their impact on LHC hadronic-channel searches

Congqiao Li (李聪乔), *Peking University*

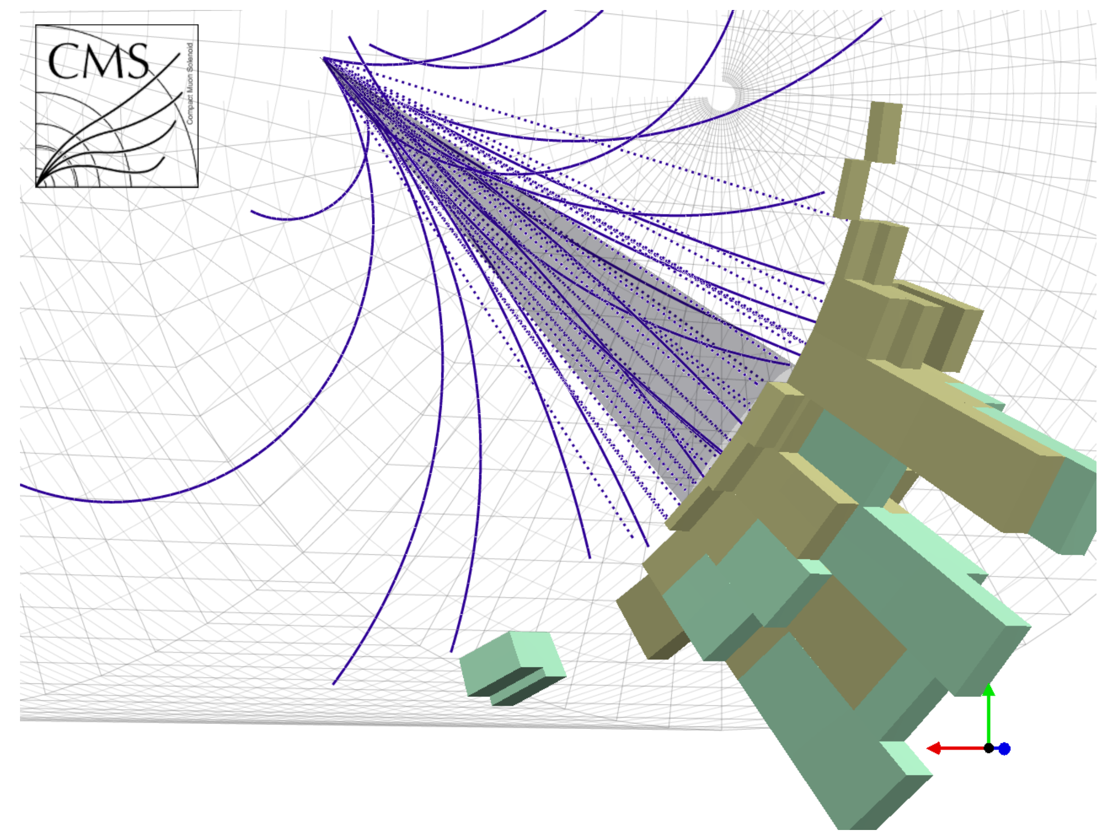
27th Mini-workshop on the frontier of LHC · Zhuhai
20 January, 2024

Jets in hadron colliders

Jets are collinear sprays of particles initiated by quark/gluons



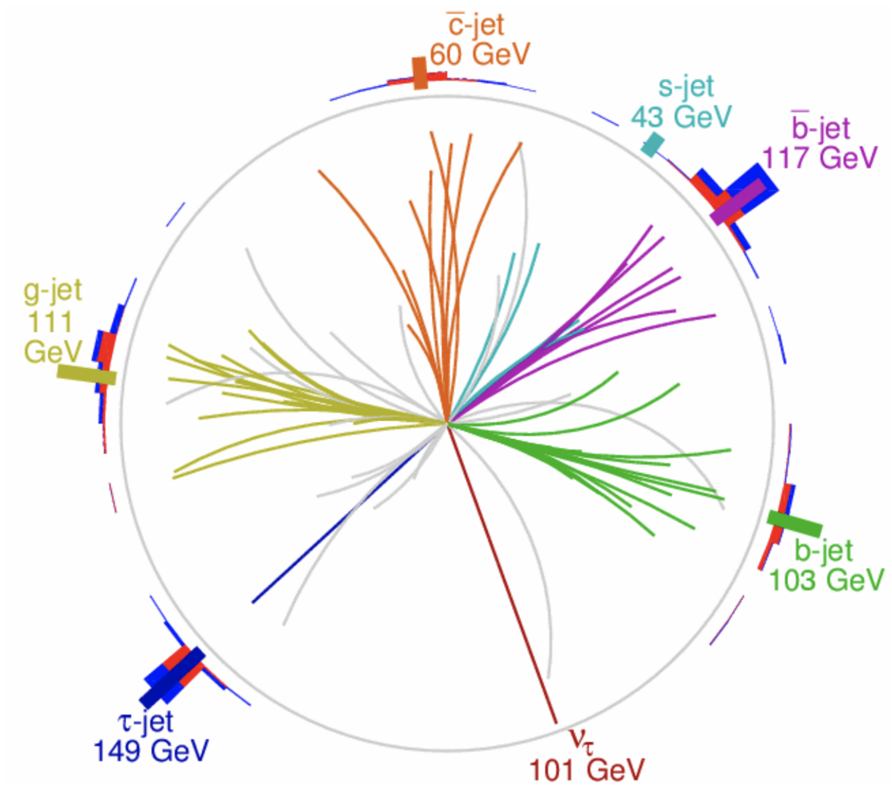
⇒ stable hadrons



raw data from tracker & calorimeter
→ reconstruct to particle records
(in CMS: particle-flow candidates)

Jets in CMS physics measurements/search

- ➔ Jets are clustered from CMS particle-flow candidates
- ➔ Clustering radius controls jet cone size
- ➔ Many jet analyze techniques:
  - ❖ identify the origin (jet tagging): b/c/light? composed W->qq/H->bb/t->bqq?
  - ❖ reveal the substructure
    - *N*-subjettiness
    - energy correctors
- ➔ Construction of variables either rule-based/machine-learning-based
  - ❖ to make selections
  - ❖ as intermediate observables for multivariate analysis
  - ❖ …

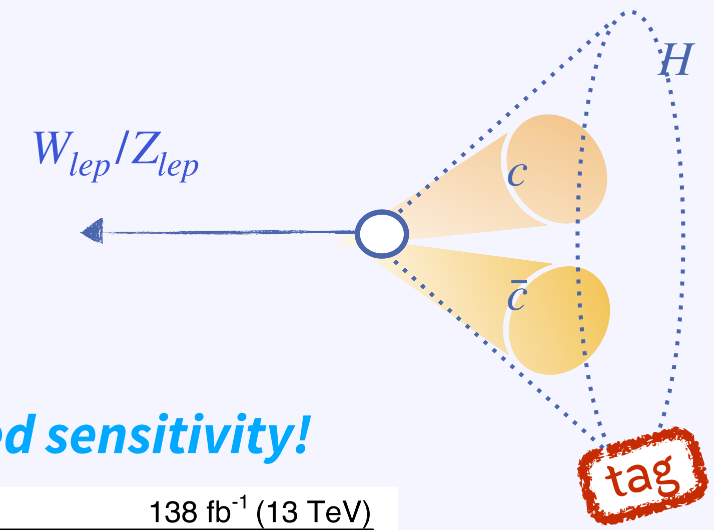


Inspiration from CMS results

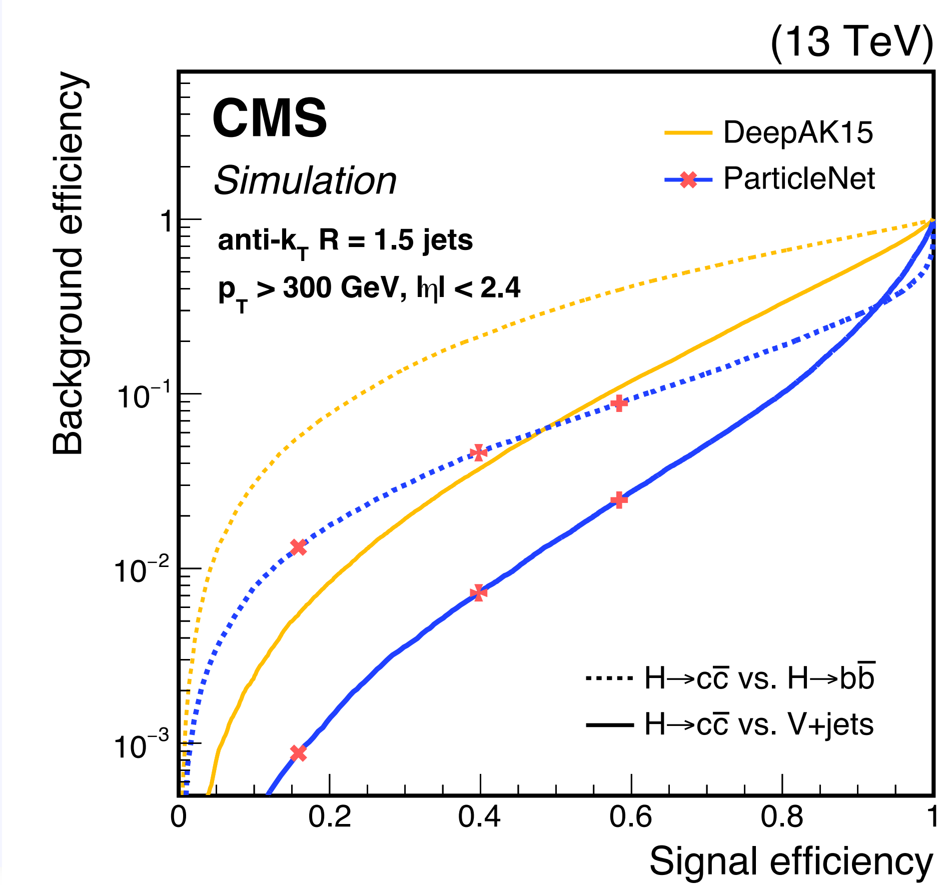
PRL 131 (2023) 061801

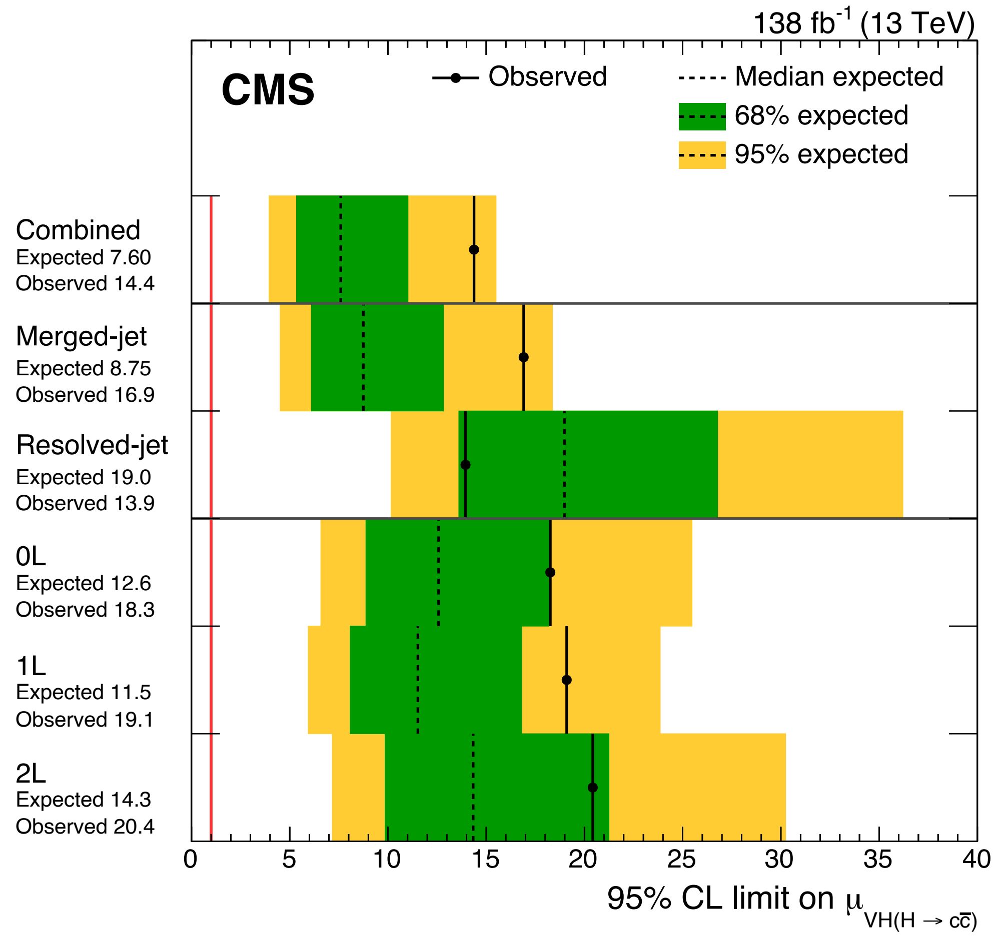
➔ CMS measures κ_c via VH(→cc̄) production mode, including the merged-jet topology

- in merged-jet topology: leveraging advanced jet neural network (ParticleNet) to identify H→cc̄ jets and reconstruct H mass
- obtain the most stringent direct limit (95% C.L.) on κ_c:
  $1.1 < |\kappa_c| < 5.5$
  - ATLAS results: $|\kappa_c| < 8.5$ [EPJC 82 (2022) 717]



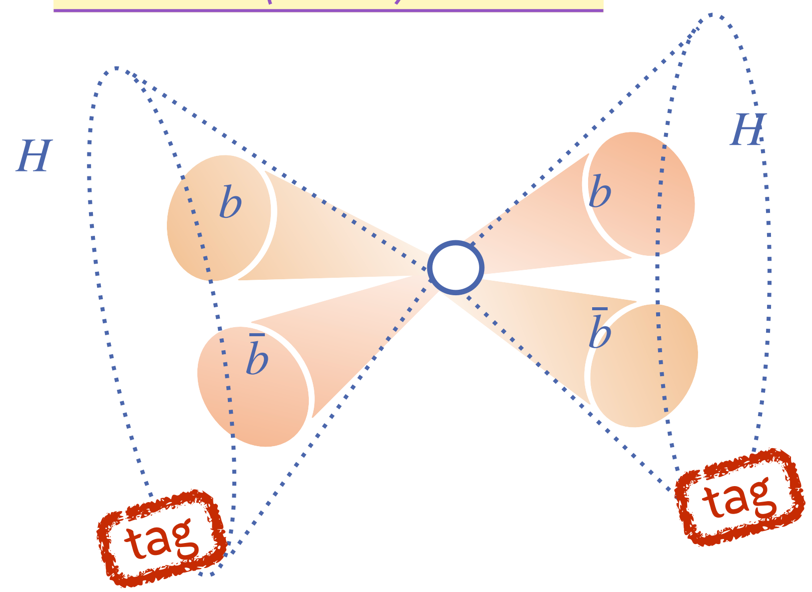
largely improved sensitivity!





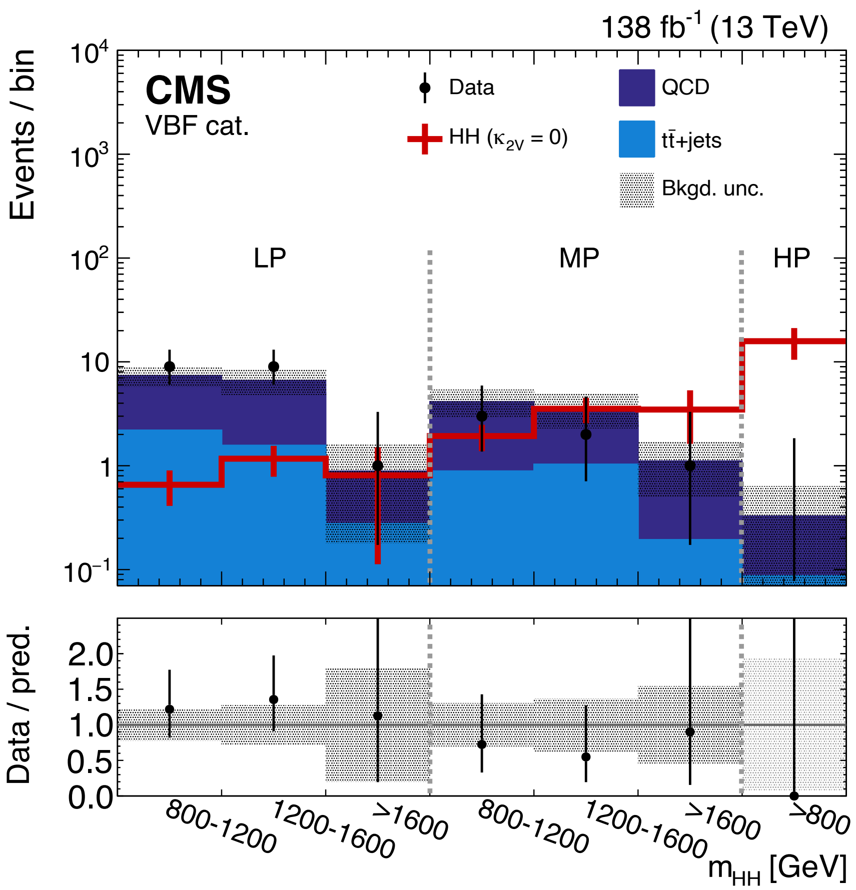
Inspiration from CMS results

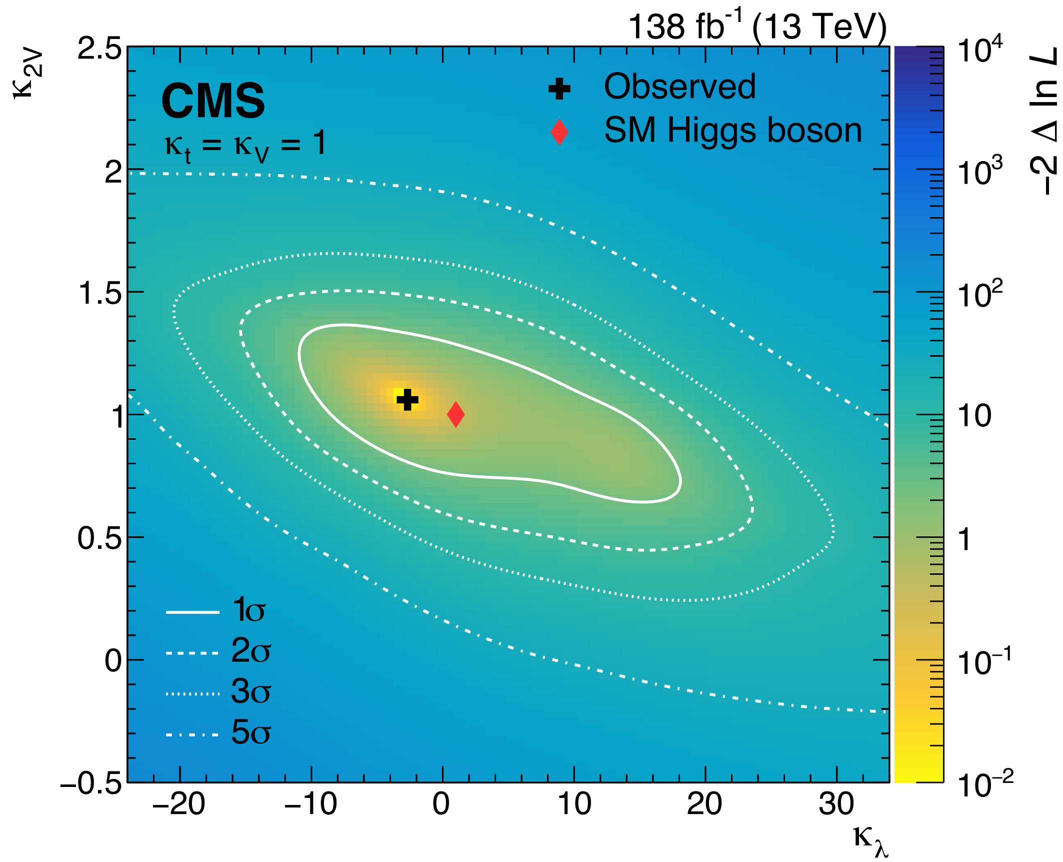
PRL 131 (2023) 041803



➔ Higgs self-coupling & quartic VVHH coupling (κ_{2V}) measured via HH→4b channel

- novel boosted-jet phase space explored by CMS
- advanced NN (ParticleNet) for H→$b\bar{b}$ jet identification and mass regression
- ***first time*** excluding $\kappa_{2V} = 0$ (by 6.3 σ)





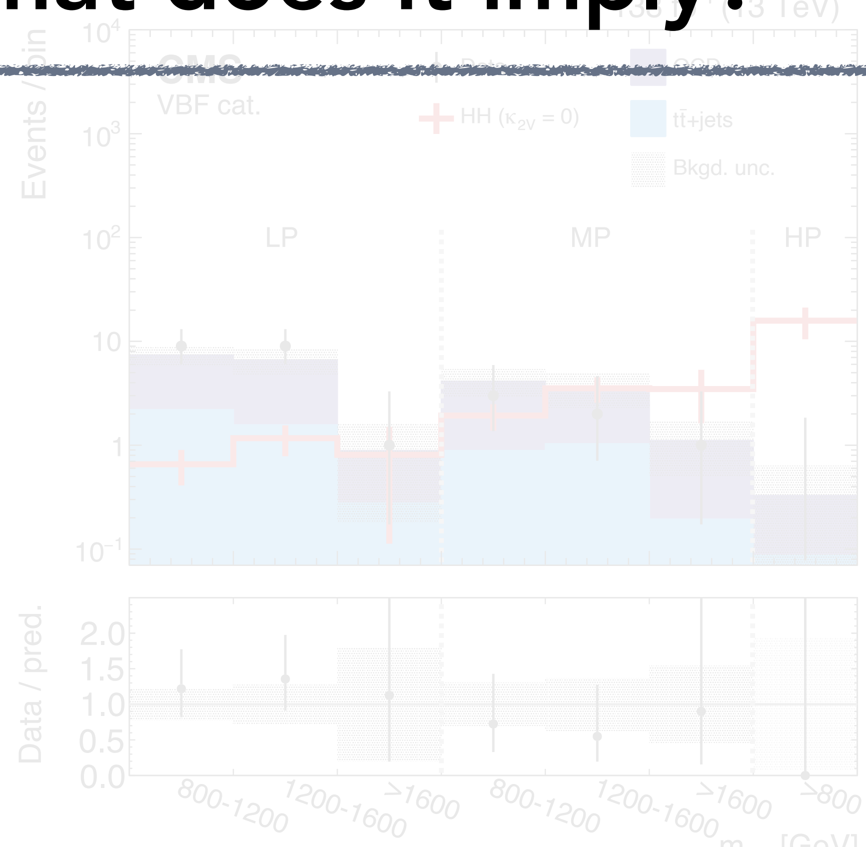
Inspiration from CMS results

PRL 131 (2023) 041803

- Higgs self-coupling & quartic VVHH coupling (κ_{2V}) measured via HH→4b channel
  - novel boosted-jet phase space explored by CMS
  - advanced NN (ParticleNet) for H→$b\bar{b}$ jet identification and mass regression

An upgrade of jet NN → *x2* sensitivity improvement

What does it imply?



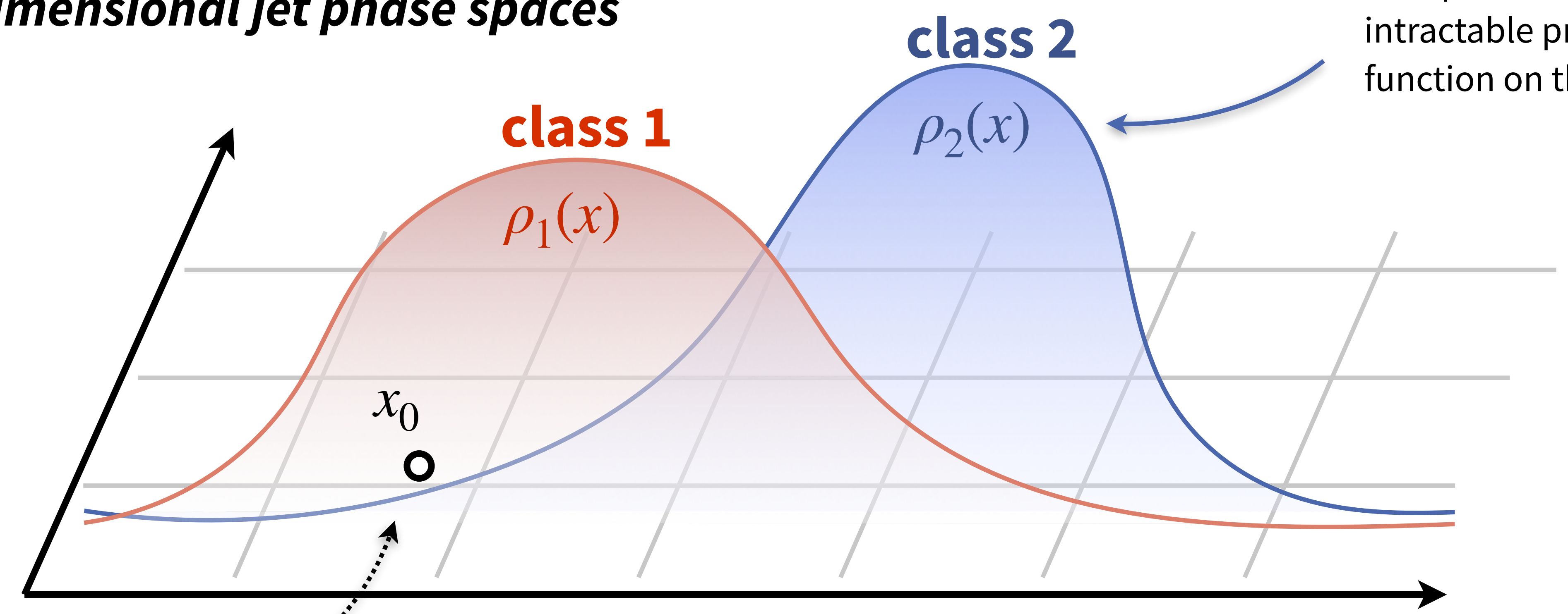
Logic behind the scene

- "Purify" signal events from the vast backgrounds are key to ~all CMS measurements
- A fundamental problem arise:
  - given our current detector capability and data collected, where is that upper bound for purifying the signal?
- Formulate the problem in statistics: likelihood ratio is always the best discriminant
  - but.. it is so intractable (the data format is too complex, and the dimension is high)
  - rule-based methods have helped us thus far (since ~1960, beginning of our collider experiment journeys), but we don't know how far is it to the ideal "upper bound"
  - these results remind us: there is much to improve in hadronic channels!

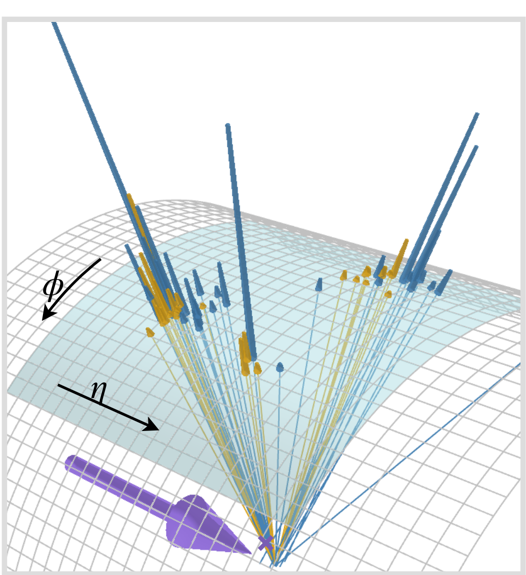
Deep, advanced NN as closest solution

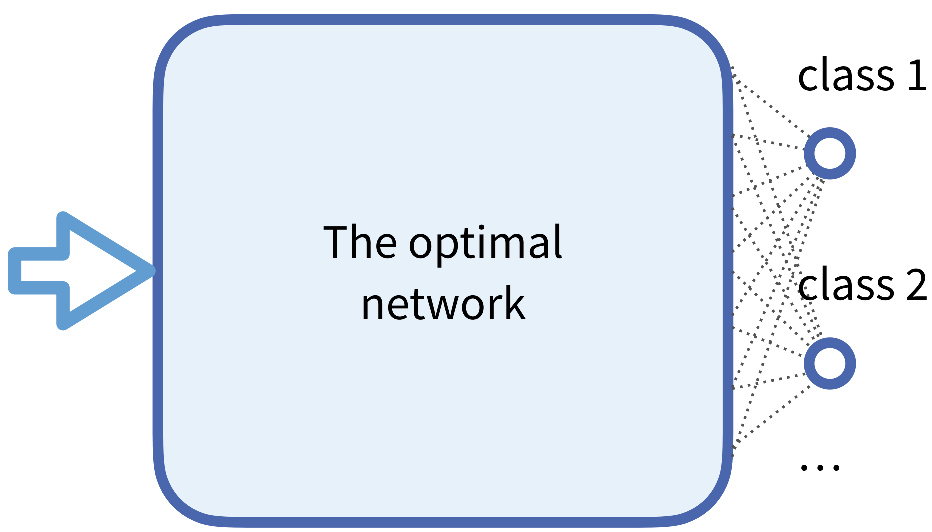
High-dimensional jet phase spaces

To represent the complex, intractable probably density function on that high-dim space



a jet

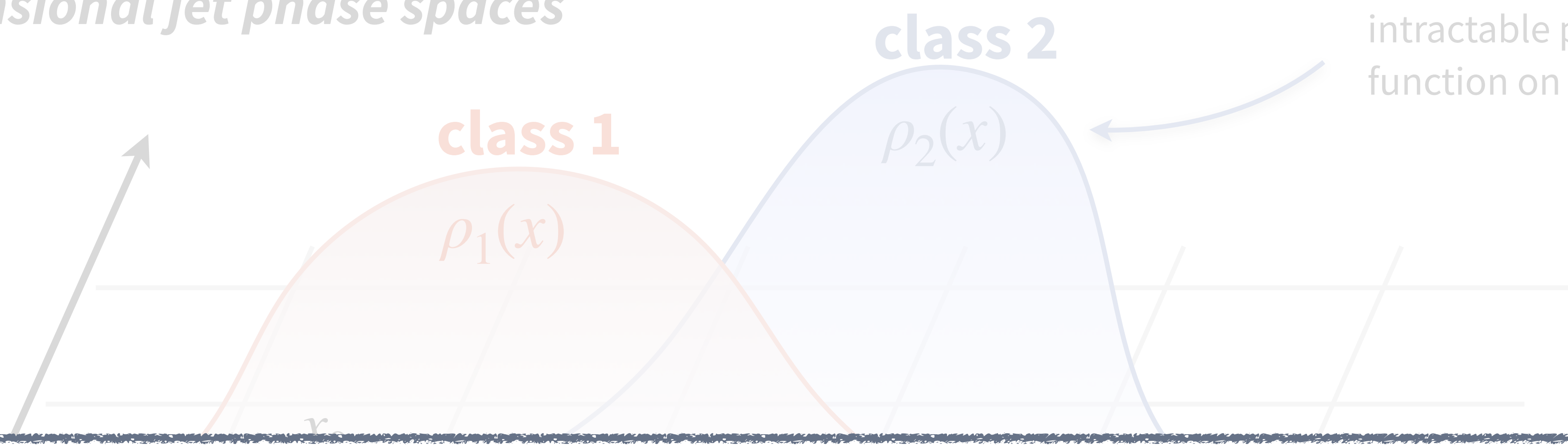




- The ideally optimal classifier network (trained with minimised cross-entropy loss) results in
  $p_1 : p_2 : \ldots = \rho_1(x_0) : \rho_2(x_0) : \ldots$
- It is a direct estimation of ρs
- The **network capacity** decides how close the estimation is

Deep, advanced NN as closest solution

High-dimensional jet phase spaces



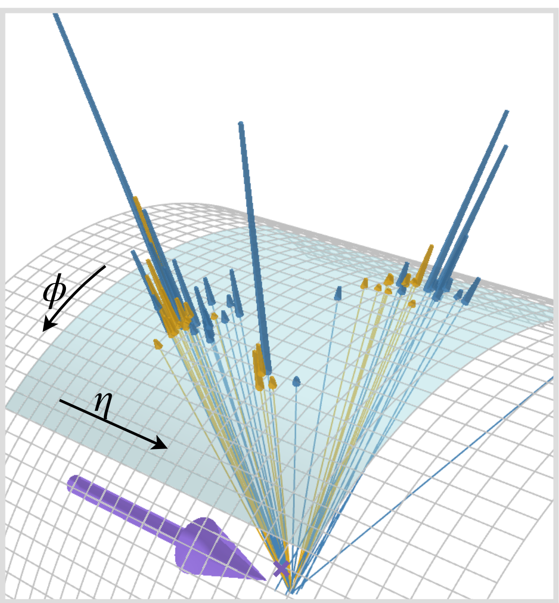
We don't know about *ρ*s (it's too complex & intractable)
We also cannot interpret the NN (NN is a black box)
But we know NN estimates *ρ* in our well-defined training strategy

a jet

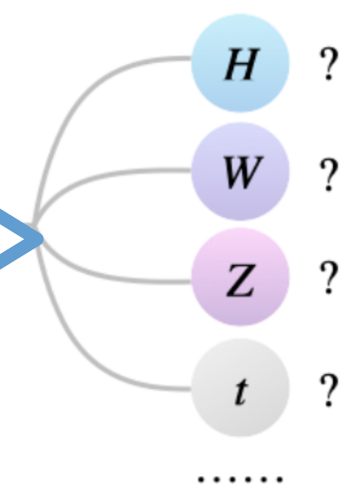
- The ideally optimal classifier network (trained with minimised cross-entropy loss) results in $p_1 : p_2 : \ldots = \rho_1(x_0) : \rho_2(x_0) : \ldots$
- It is a direct estimation of ρs
- The **network capacity** decides how close the estimation is

How to design a most performant jet NN?

- ➔ We have to ask: How to design a most performant jet NN?
- ➔ This is a highly physics-ML interdisciplinary subject
  - ❖ The following slides: cherry-pick some recent advancements



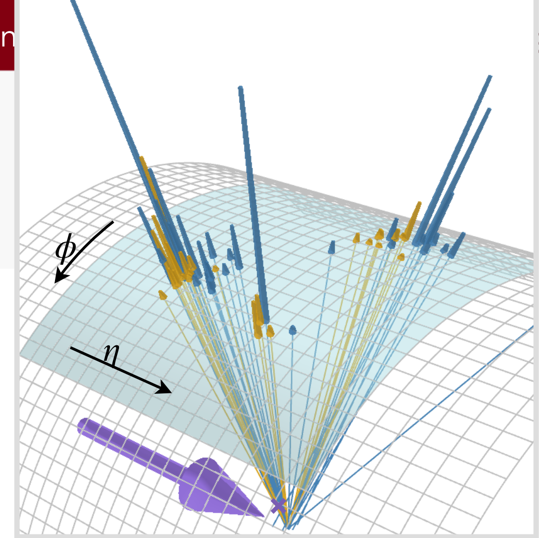
original of the jet?



other jet properties?

research objective:
design a most performant jet NN

Jet representation



earlier approach

better approach

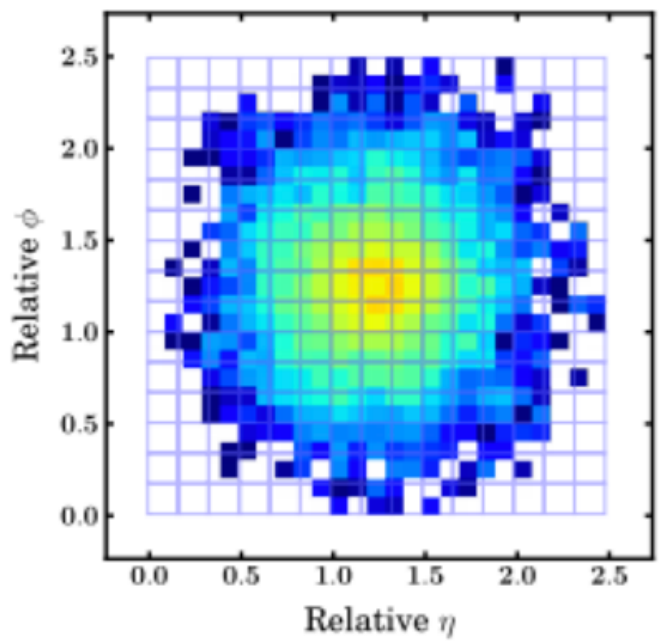
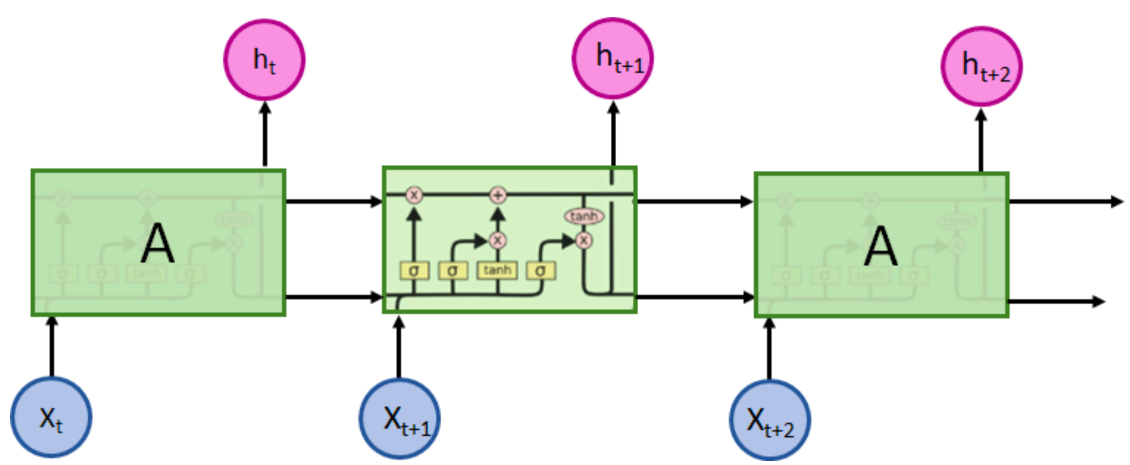
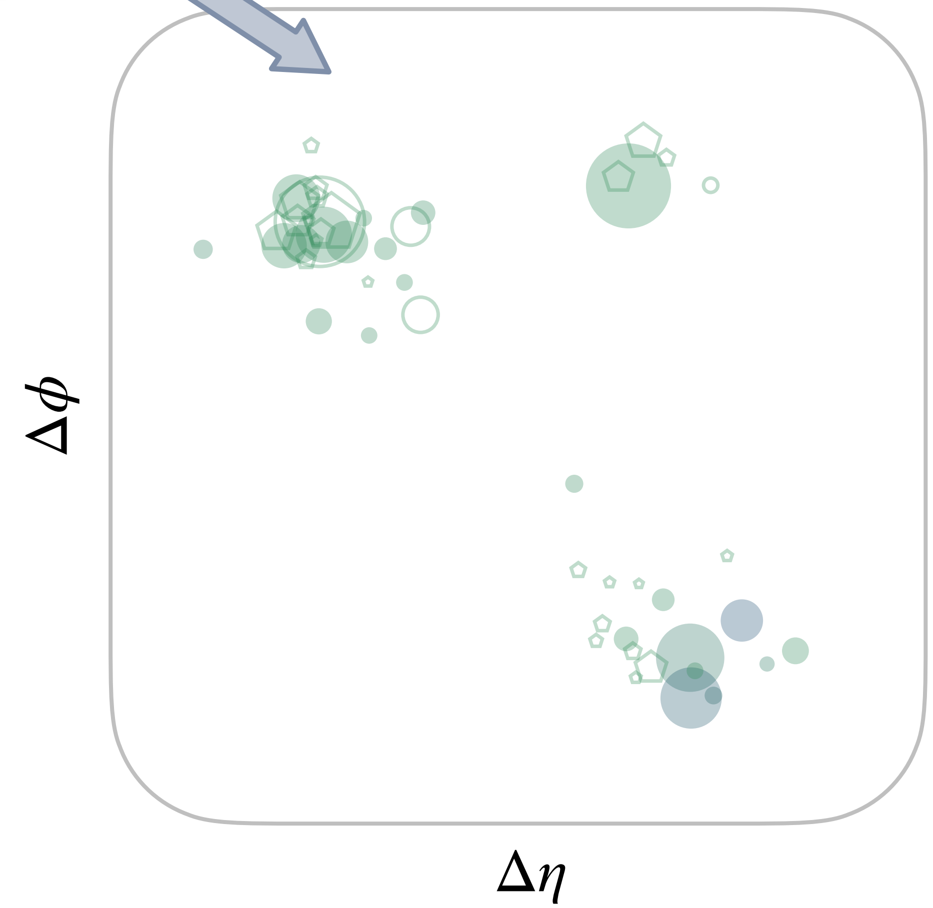


Image:
has information loss,
brings data sparsity

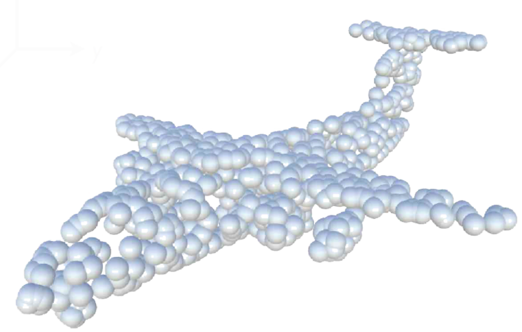
Sequence:
introduces artificial order



particle 1 particle 2 particle 3



Point cloud (set): a more natural way to represent jet

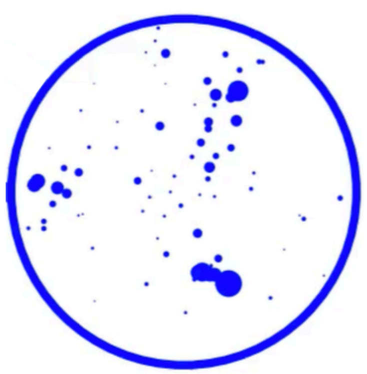


good analogue to point-cloud tasks
"graph-based netowrks"

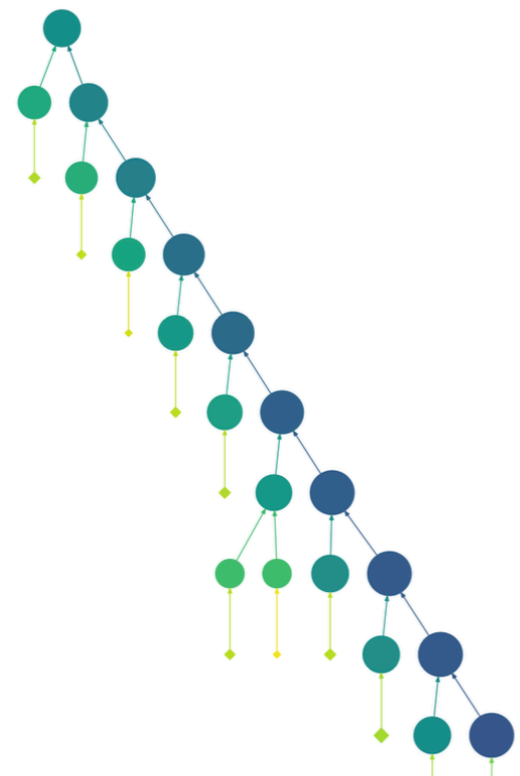
"Graph" neural networks

- ➔ View input particles as a set/graph
  - ❖ guarantee the *permutational invariance* of input particles
- ➔ The **edges** of graph: enable communication between pairs of particles

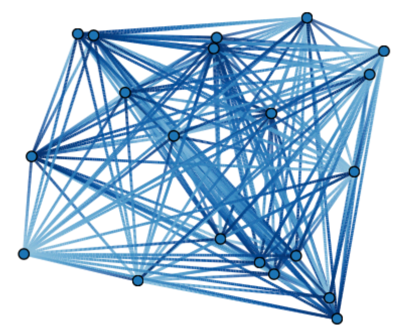
Set: no edges



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

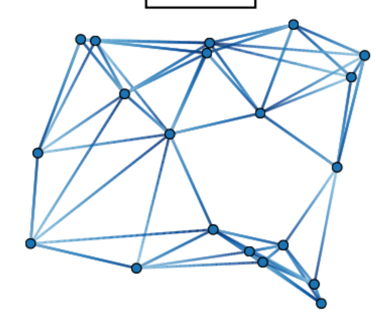


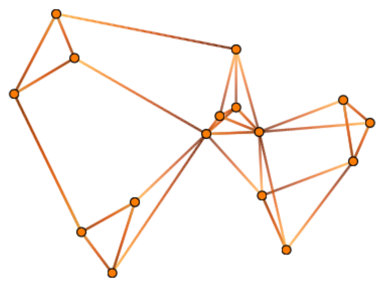
Fully connected graph
- *i.e., connect each node to all other nodes*



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

static





(dynamically) learned

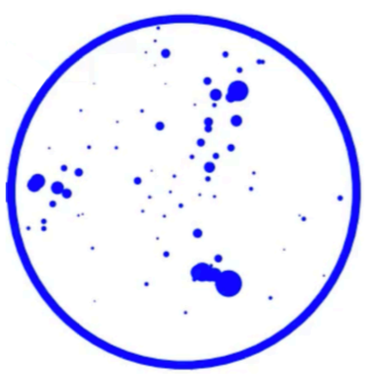
[image from link]

"Graph" neural networks

- ➔ View input particles as a set/graph
  - ❖ guarantee the *permutational in*
- ➔ The **edges** of graph: enable con … f particles

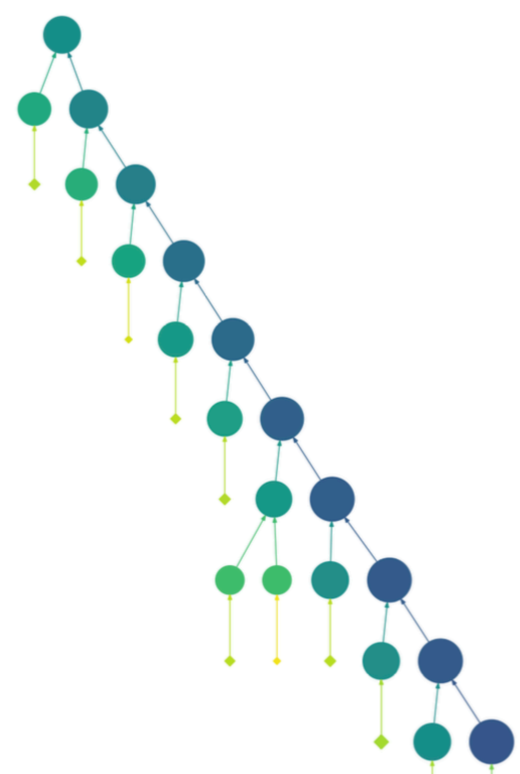
LorentzNet: S. Gong et al. JHEP 07 (2022) 030
ParT: H. Qu et al. arXiv:2202.03772, ICML 2022
CPT : S. Qiu et al. PRD 107 (2023) 11, 114029
HMPNet : F. Ma et al. PRD 108 (2023) 7, 072007

Set: no edges



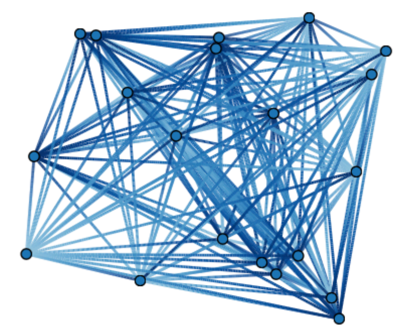
PFN/EFN: P. Komiske et al. JHEP 01 (2019) 121

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



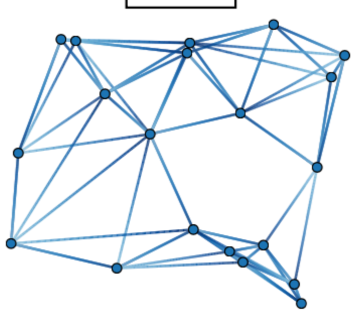
LundNet: F. Dreyer et al. JHEP 03 (2021) 052

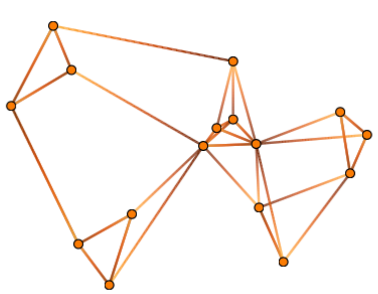
Fully connected graph
- *i.e., connect each node to all other nodes*



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

static

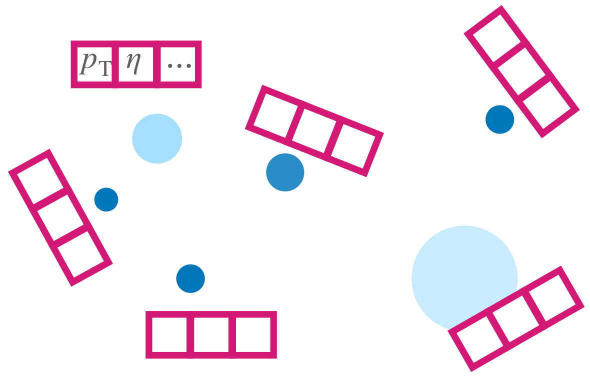




(dynamically) learned

ParticleNet: H.Qu et al. PRD 101, 056019 (2020)
ABCNet: V. Mikuni et al. EPJC 2020; 135(6): 463

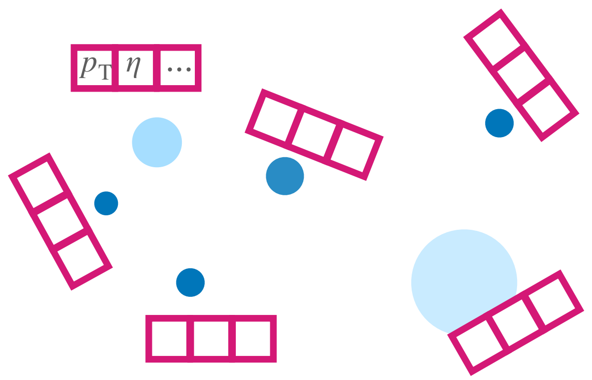
"Graph" processing prototype

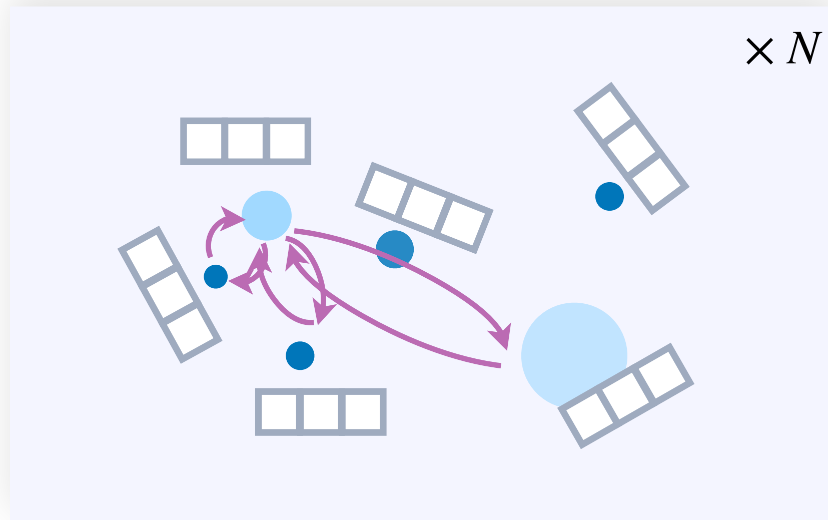


Input jet with per-particle features

- four-momentum; or equiv. (E, p_T, η, ϕ)
- reconstructed particle ID
- track displacement (for charged particle)
- …

“Graph” processing prototype



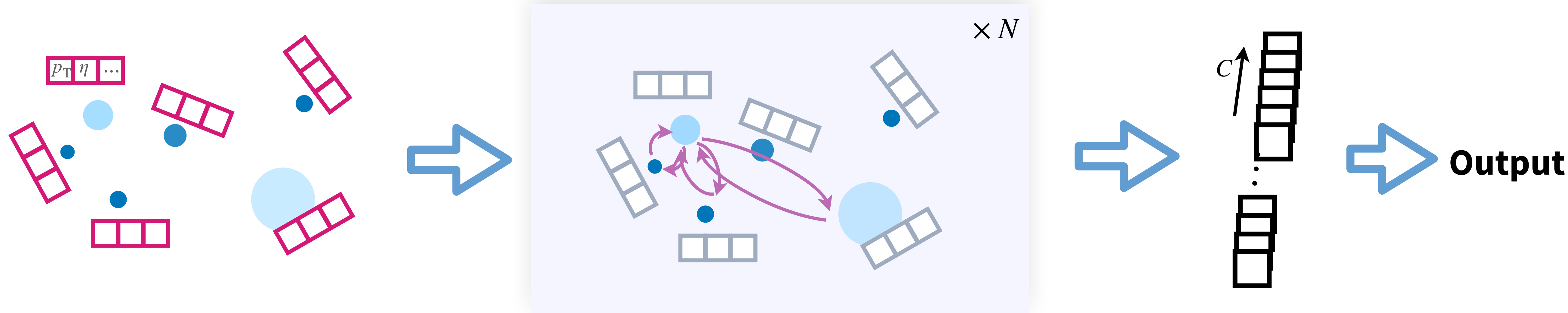


Input jet with per-particle features

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

Massage passing layers
“communicate between particles”

“Graph” processing prototype



Input jet with per-particle features

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

Massage passing layers
“communicate between particles”

Feature pooling
“summarize all particles into one”

The full process is invariant to the permutation of particles

Engineering with graphs

- ✦ Preferred full connected
- ✦ Average information by weights
- ✦ “Multi” over “one”
- ✦ Pairwise features help

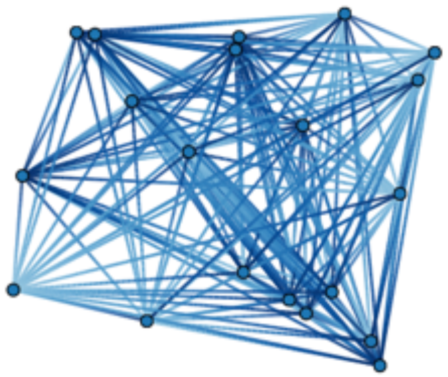
Reference:
ABCNet: V. Mikuni et al. EPJC 2020; 135(6): 463
LGN: A. Bogatskiy et al. arXiv: 2006.04780, ICML 2020
ParticleNeXt: H. Qu. Talk@ML4Jets2021
LundNet: F. Dreyer et al. JHEP 03 (2021) 052
PCN: C. Shimmin. arXiv:2107.02908
LorentzNet: S. Gong et al. JHEP 07 (2022) 030
PCT@hep: V.Mikuni et al. 2021 MLST 2 035027
ParT: H. Qu et al. arXiv:2202.03772, ICML 2022
CPT : S. Qiu et al. PRD 107 (2023) 11, 114029
HMPNet : F. Ma et al. PRD 108 (2023) 7, 072007
PELICAN: A.Bogatskiy et al. arXiv:2211.00454

Engineering with graphs

✦ **Preferred full connected**

✦ Average information by weights

✦ “Multi” over “one”



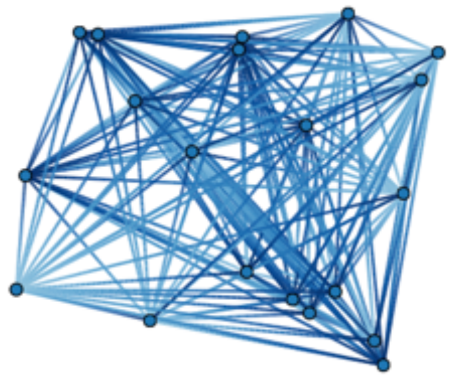
In terms of performance:

fully-connected graph
> edges from nearest neighbours
> no edge

Engineering with graphs

✦ **Preferred full connected**

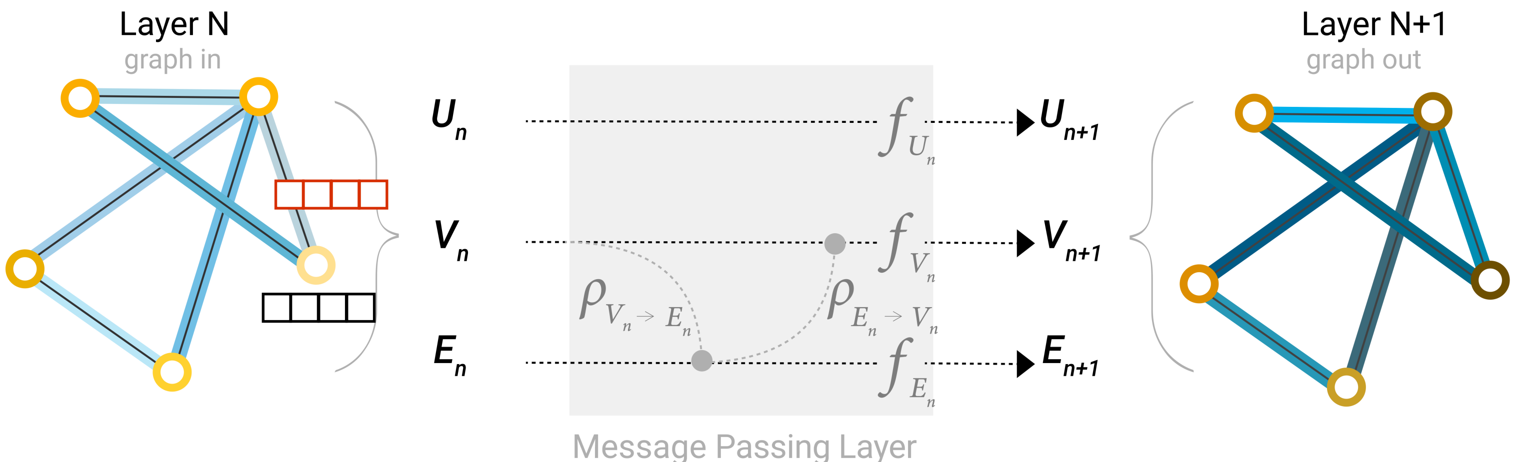
✦ Average information by weights



In terms of performance:

fully-connected graph
> edges from nearest neighbours
> no edge

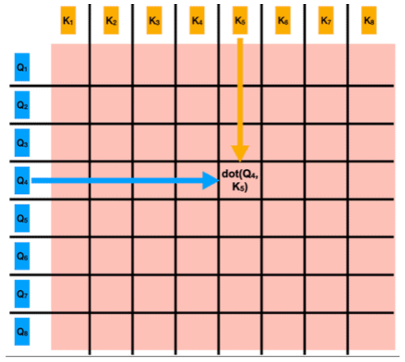
Typical "graph neural network"



Node -> edge -> node
(build edge between all pairs)

Typical "Transformer network"

particles as "query"



particles as "key"

"Fast communication" between all pairs via dot-product

Engineering with graphs

✦ Preferred full connected

✦ **Average information by weights**

when aggregating features among all particles, using ***average/max pooling*** *losses more information;* ***assigning learnable weights to particles*** *usually works better*

$$z_i = \mathrm{mean}_j(e_{ij})$$

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

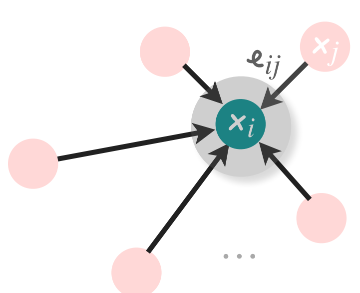
ParticleNeXt: H. Qu. Talk@ML4Jets2021

Engineering with graphs

✦ Preferred full connected

✦ **Average information by weights**

ParticleNet



$z_i = mean_j(e_{ij})$

att

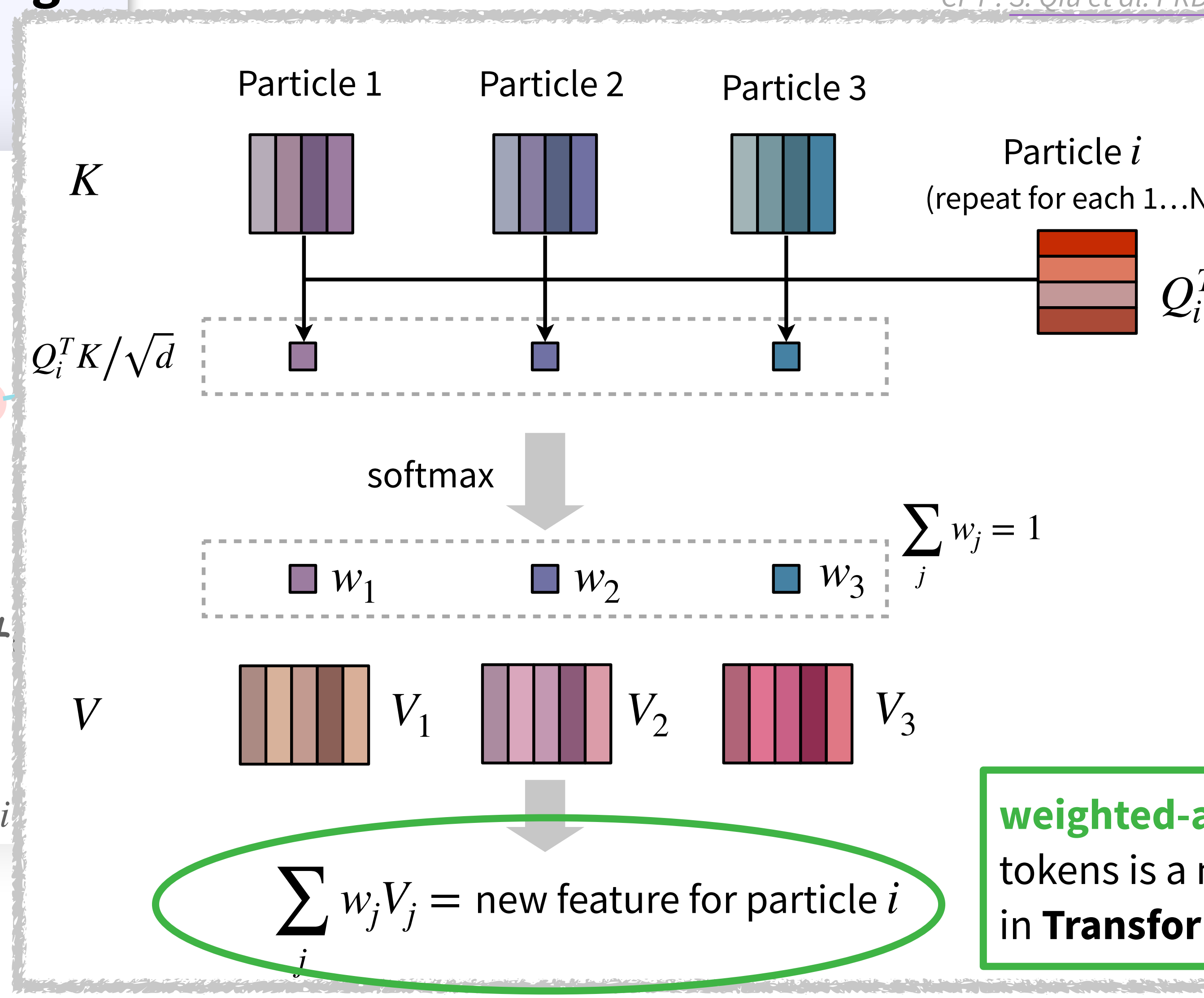
$w_{ij} =$

z_i

ParticleNeXt: H. Qu. Talk@ML4Jets2021

*when aggregating features among all particles, using **average/max pooling** losses more information; **assigning learnable weights to particles** usually works better*

PCT@hep: V.Mikuni et al. 2021 MLST 2 035027
ParT: H. Qu et al. arXiv:2202.03772, ICML 2022
CPT : S. Qiu et al. PRD 107 (2023) 11, 114029

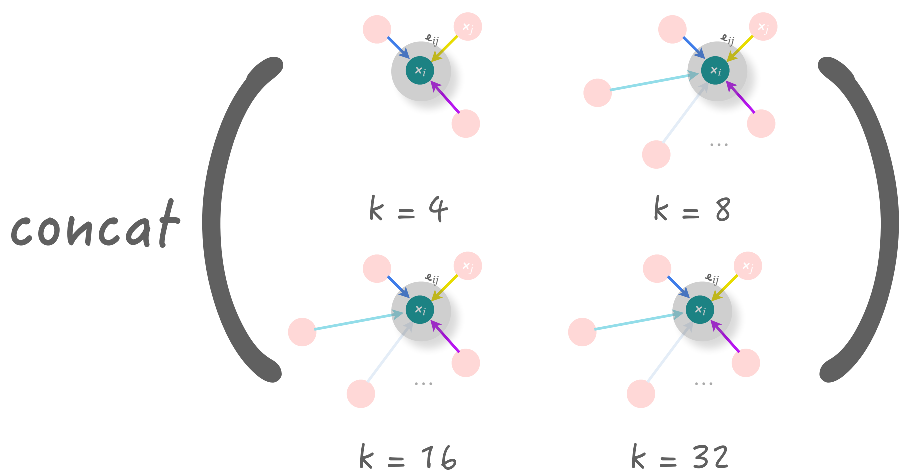


weighted-average over tokens is a native feature in **Transformers**

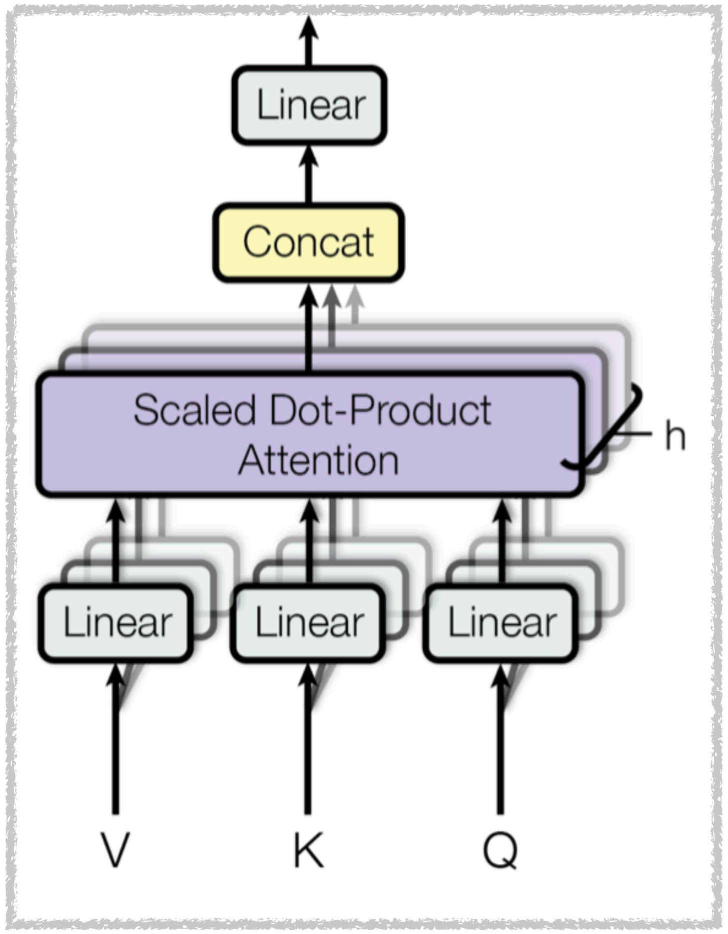
Engineering with graphs

- ✦ Preferred full connected
- ✦ Average information by weights
- ✦ **“Multi” over “one”**
- ✦ Pairwise features help

ParticleNeXt: H. Qu. Talk@ML4Jets2021



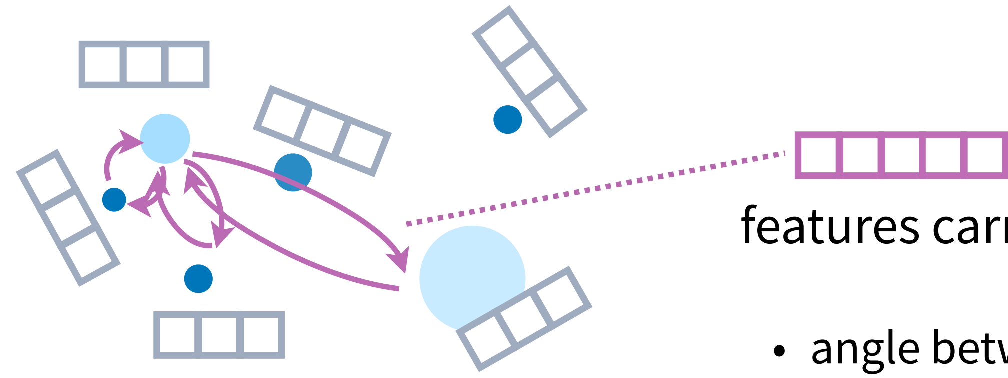
Multi-scale communications between neighbours



Classical Transformers enable “multi-head” attention

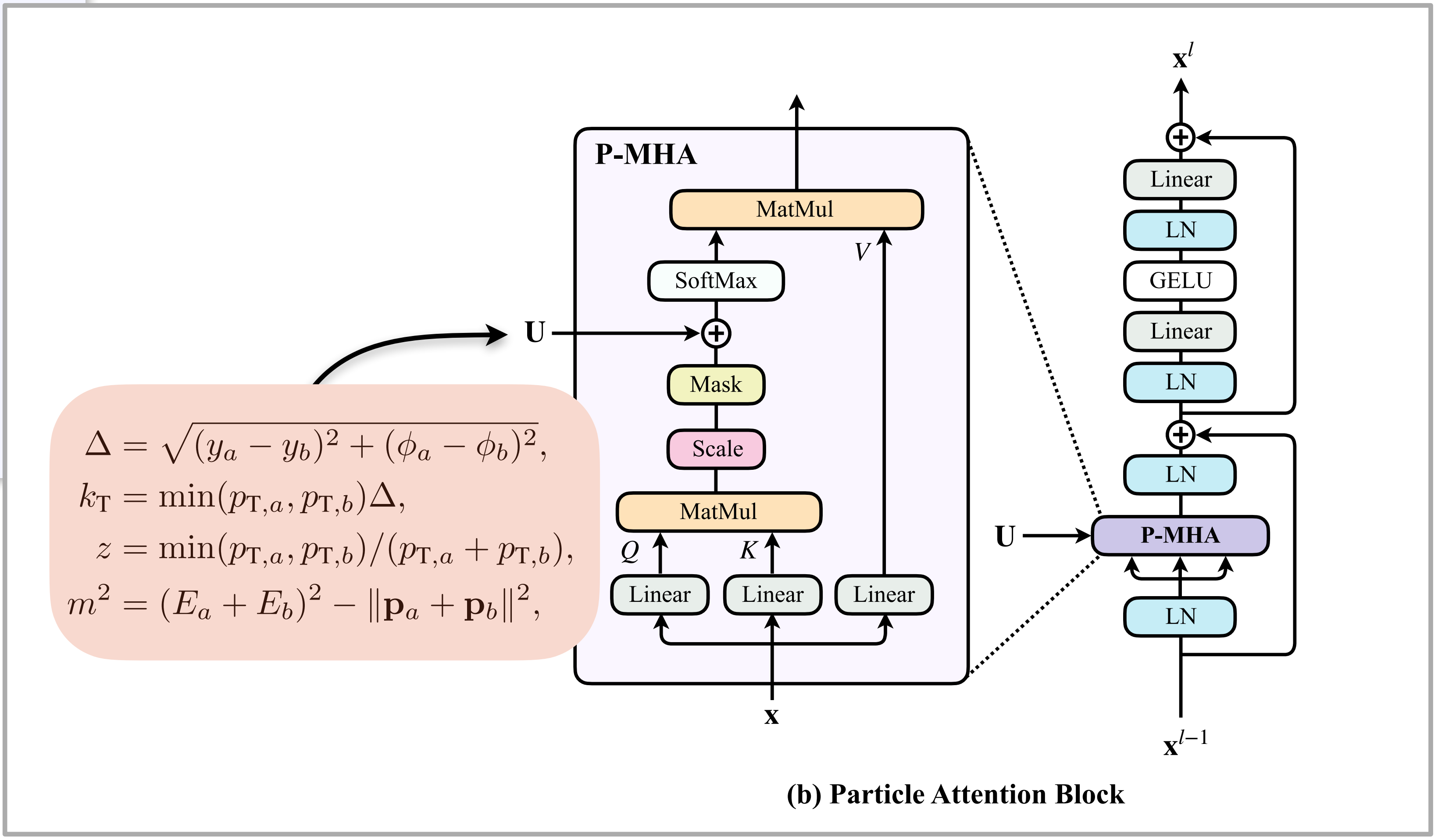
Engineering with graphs

- ✦ Preferred full connected
- ✦ Average information by weights
- ✦ “Multi” over “one”
- ✦ **Pairwise features help**



features carried on “edges”

- angle between particles
- pairwise invariant mass
- track distance, …



(b) Particle Attention Block

ParticleNeXt: H. Qu. Talk@ML4Jets2021
ParT: H. Qu et al. arXiv:2202.03772, ICML 2022

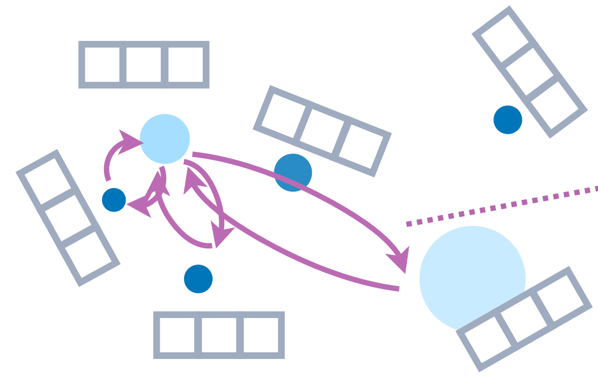
Engineering with graphs

✦ Preferred full connected

✦ Average information by weights

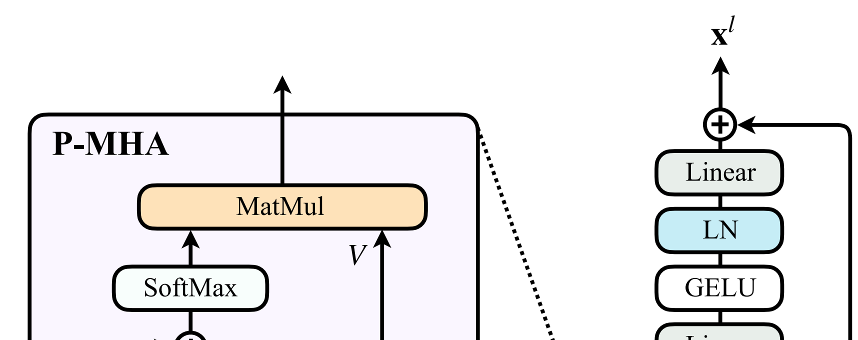
✦ "Multi" over "one"

✦ **Pairwise features help**

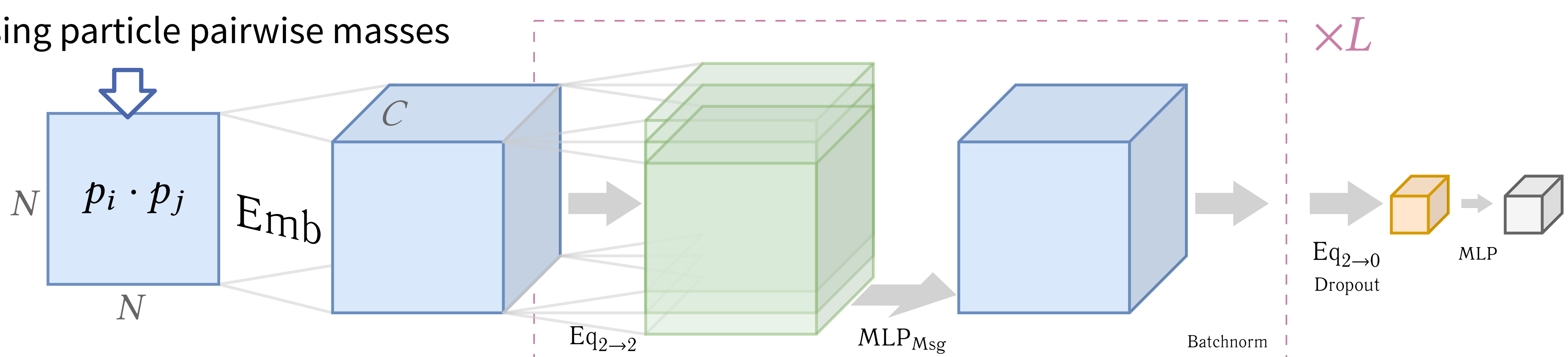


features carried on "edges"

- angle between particles
- pairwise invariant mass
- track distance, …

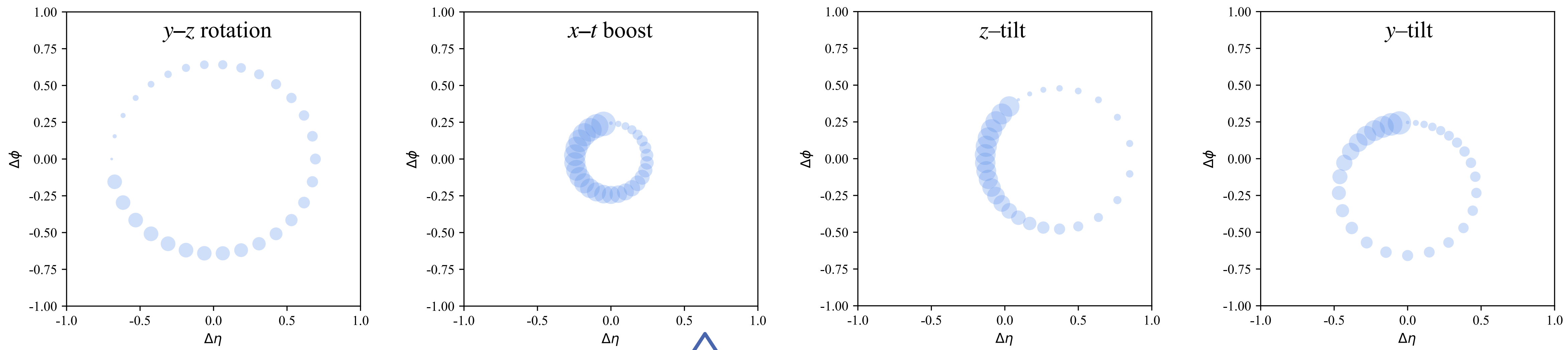


merely using particle pairwise masses

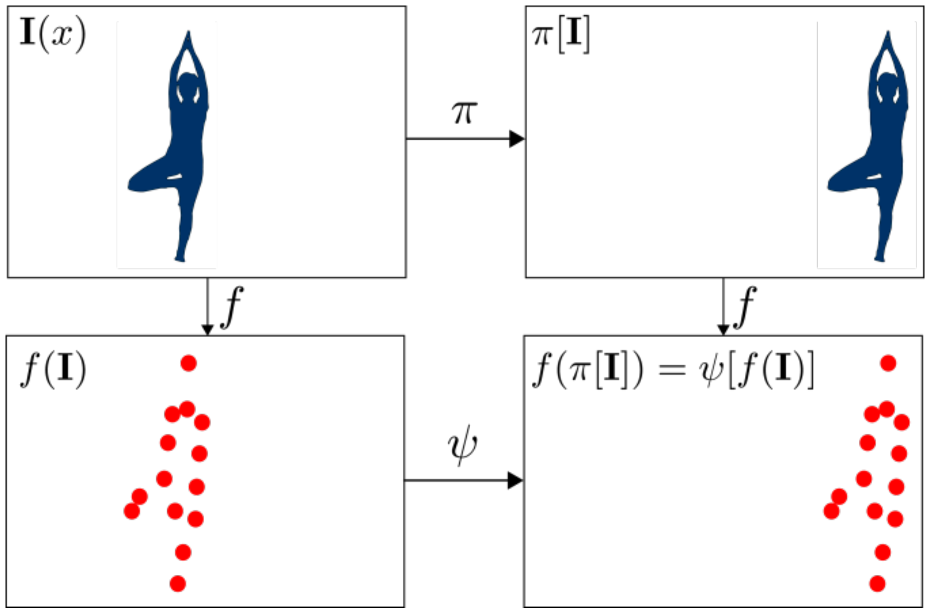


PELICAN: A.Bogatskiy et al. arXiv:2211.00454; arXiv:2310.16121, NeurIPS 2023 Workshop

Symmetries and inductive biases



Hint the network:
our input jet property usually do not change, if all/some of its particle undergoes Lorentz transformations



(an analogy from CNN for vision tasks)

➔ Symmetries can usally be used as inductive bias

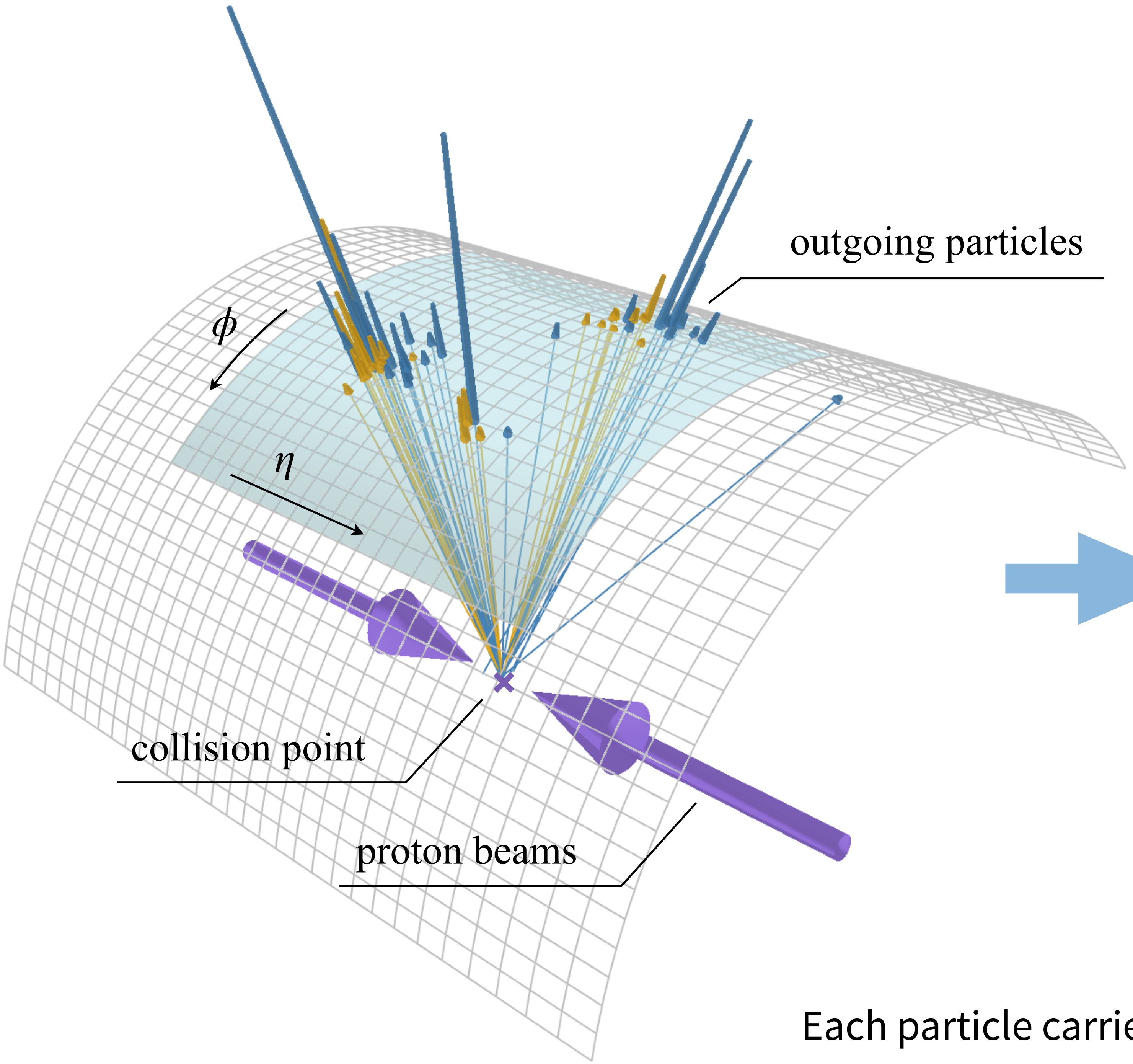
➔ Jets have symmetries under permutations & Lorentz transformations

Summary of the talk

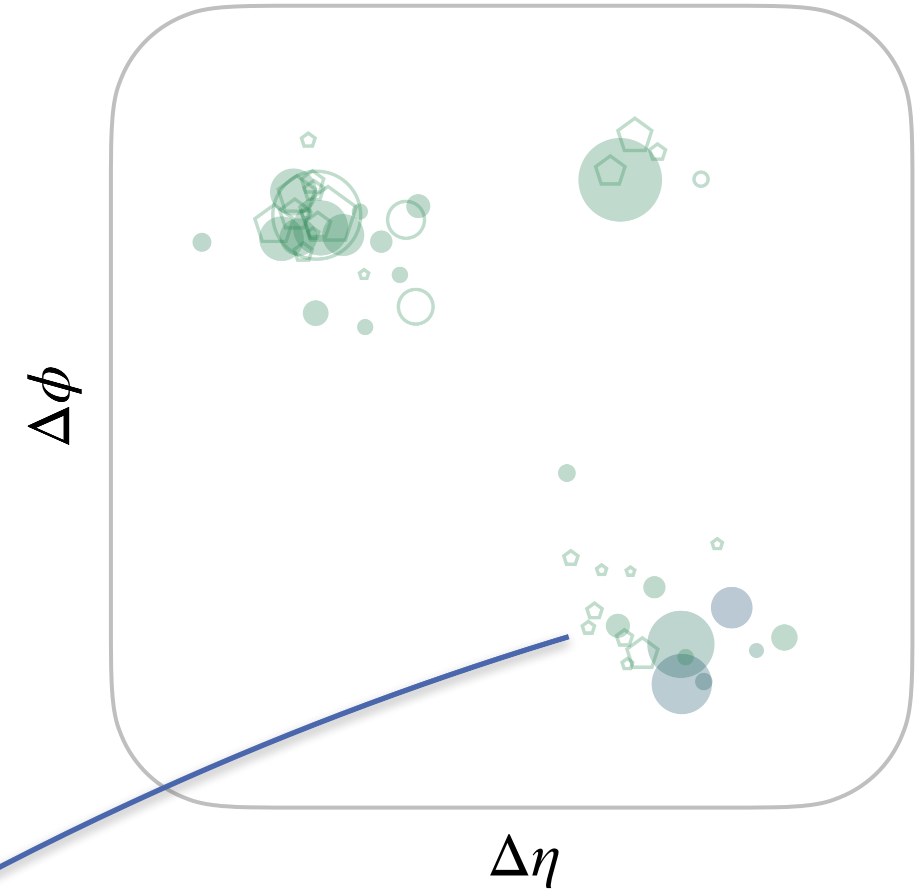
- ➔ From two recent CMS results, we reveal the underlying message to experimentalists
  - ❖ sensitivity in hadronic channels is quite under-explored
  - ❖ formulate the problem from a statistical view: recognize the "upper bound" and understand how close we are now
  - ❖ reveal that the cutting-edge NN approaches provide the closest way by far to reach the target
- ➔ Overview of the progress made in designing advanced jet NN
  - ❖ better jet representations
  - ❖ "graph" building and its engineering experiences
  - ❖ impacts from intrinsic jet symmetries
- ➔ Picturing a promising path of our experimental evolution with advanced AI

Backup

View of a jet



Viewing jets on the η-ϕ space



[image from link]

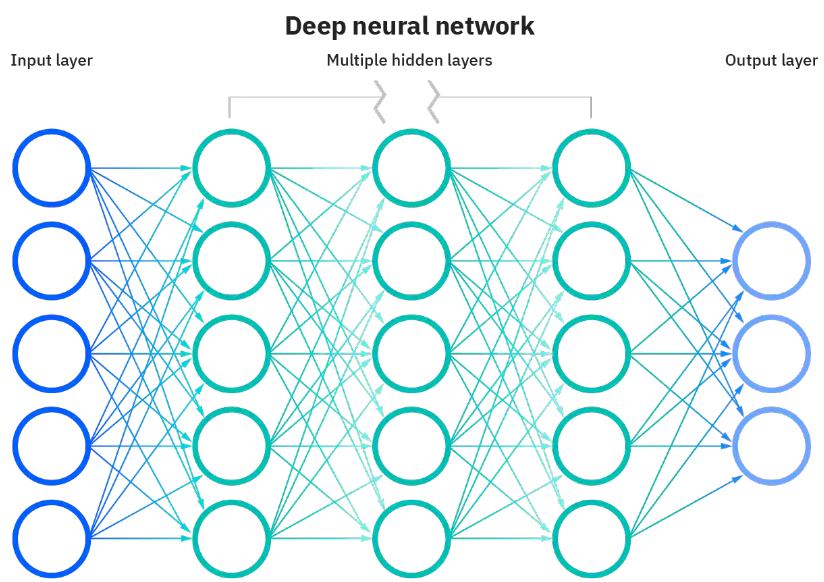
Each particle carries features:

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

** not necessarily exists*

Roadmap of DL model for jet tagging

➔ DL model design draw from experiences in Computer Vision

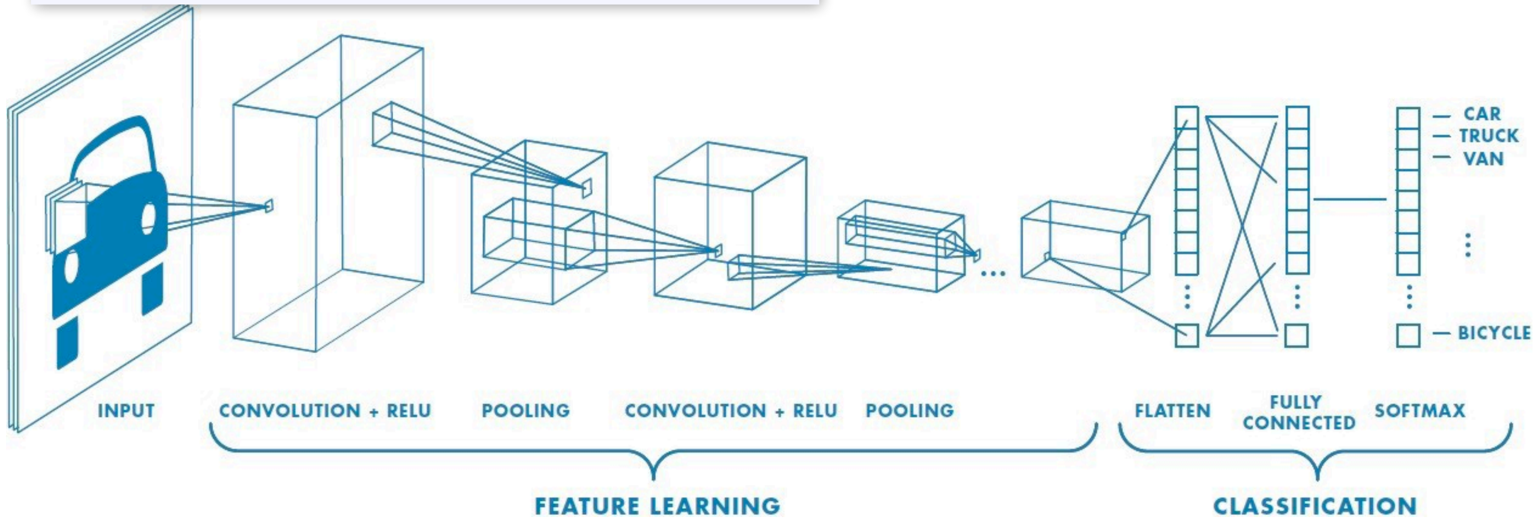


conventional "deep neural network" or multi-layer perceptron (MLP)

process fix-length input data

convolutional neural network (CNN)

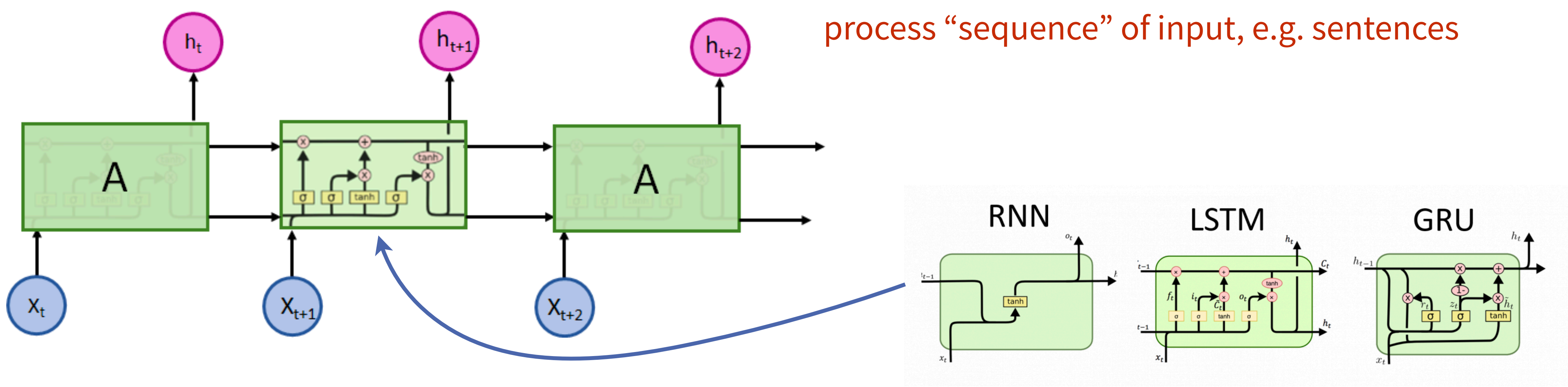
process images



[image from link]

recurrent neural network (RNN) & LSTM

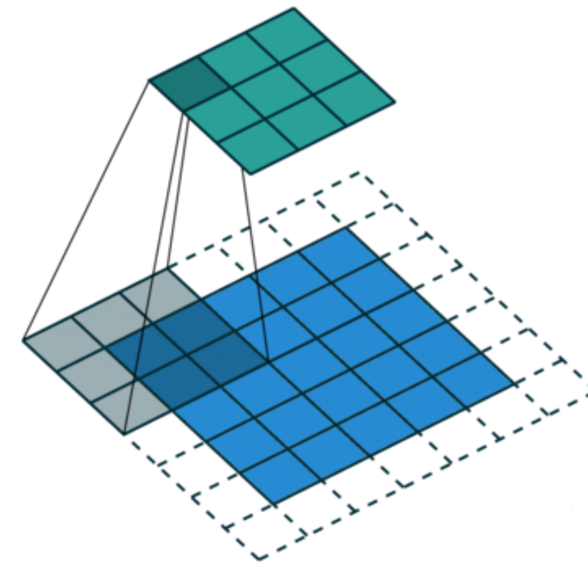
process "sequence" of input, e.g. sentences



Roadmap of DL model for jet tagging

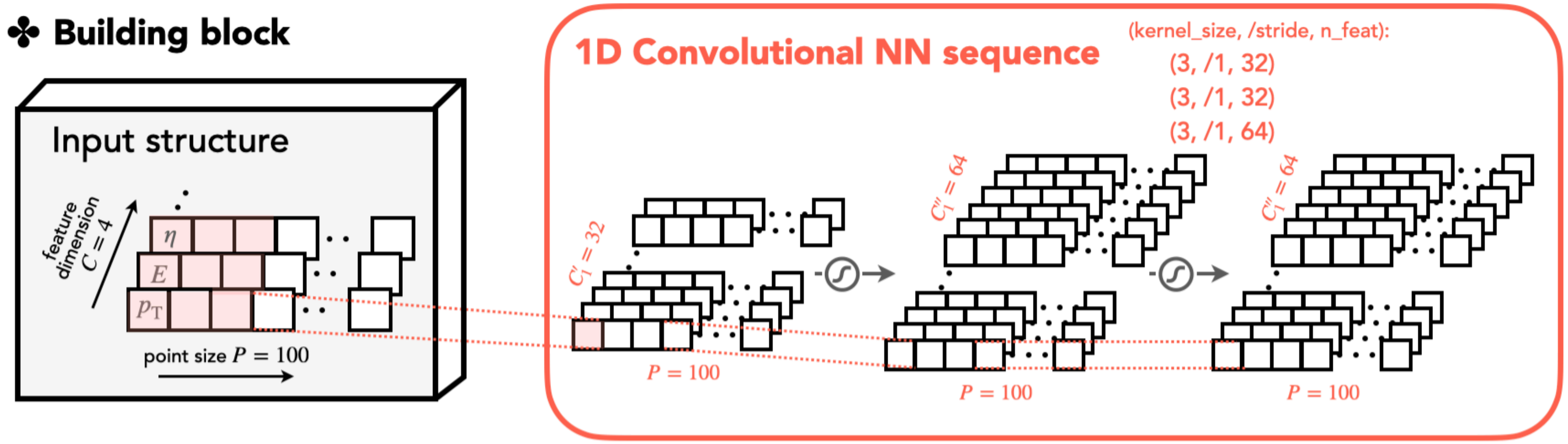
EXAMPLE OF CNN

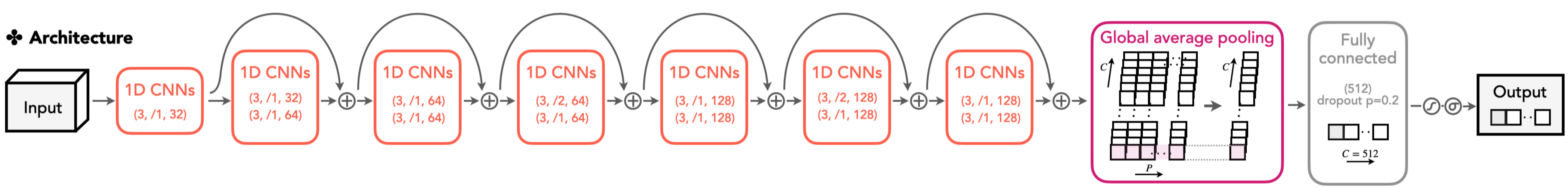
illustration of 2D convolution



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

[CMS, JINST 15 (2020) P06005]



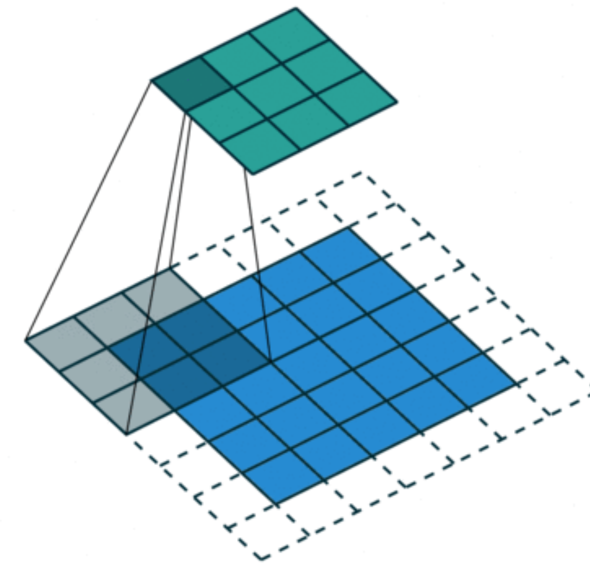


[image from link]

Roadmap of DL model for jet tagging

EXAMPLE OF CNN

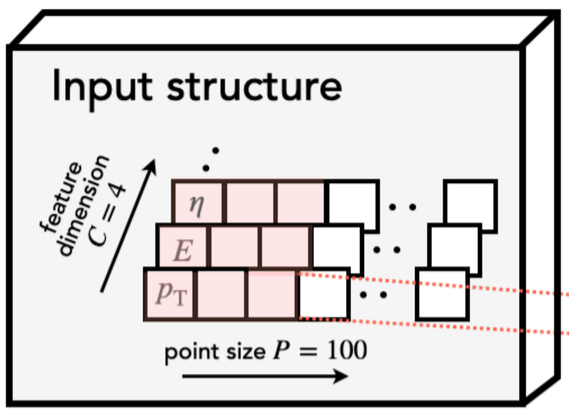
illustration of 2D convolution

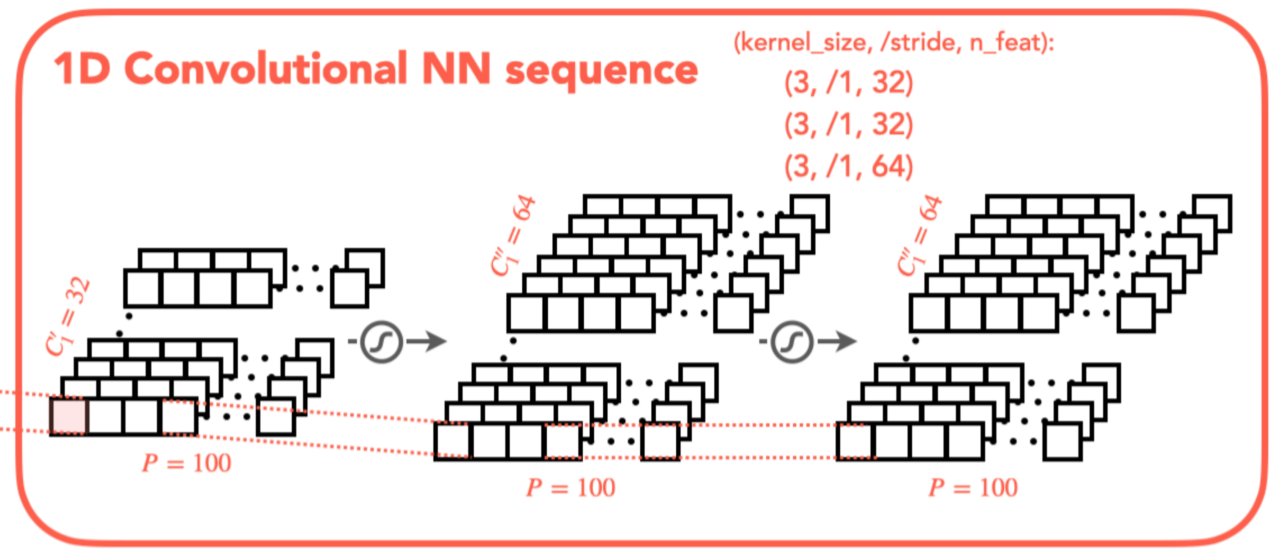


The previous jet tagging model in CMS: **DeepAK8** algorithm

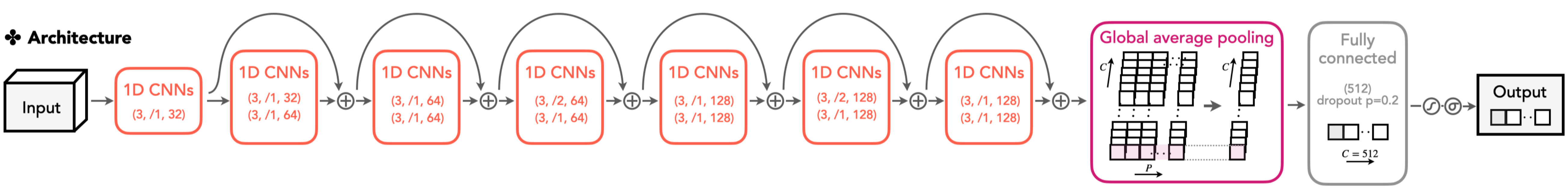
CMS, JINST 15 (2020) P06005

✤ **Building block**





✤ **Architecture**



[image from link]

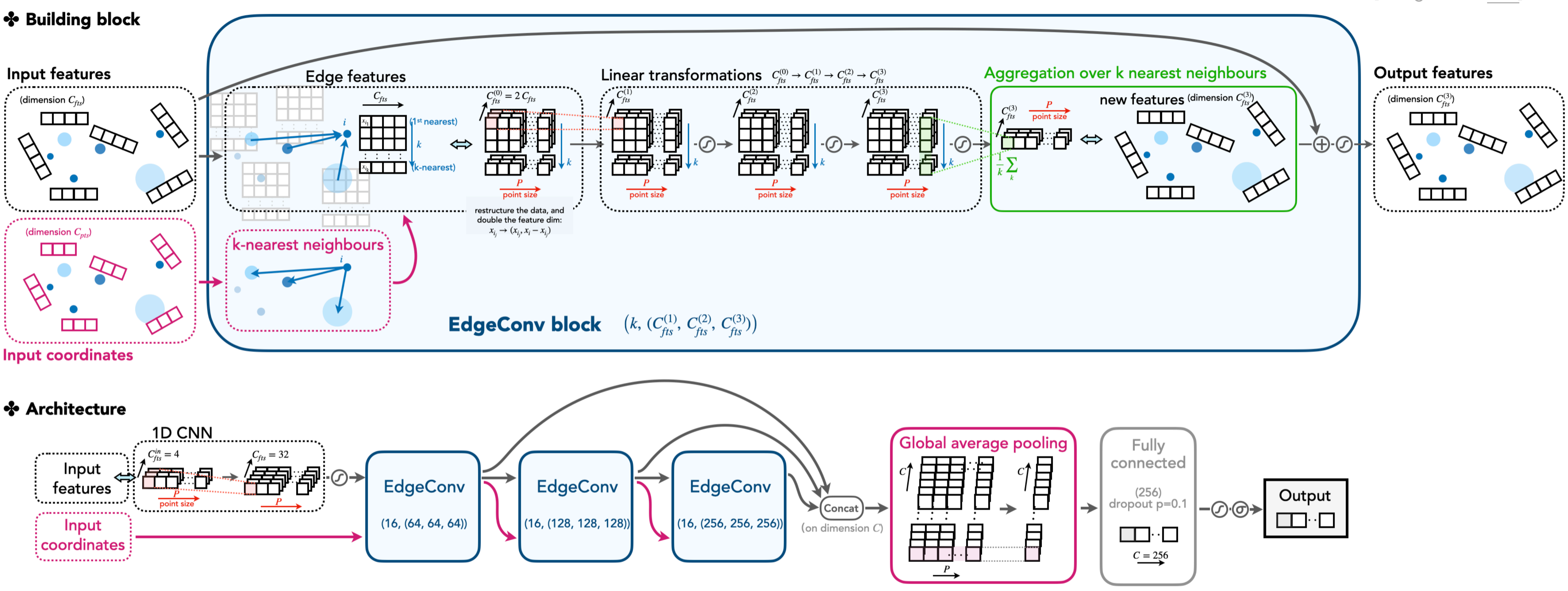
Roadmap of DL model for jet tagging

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]



"Post-ParticleNet" DL studies

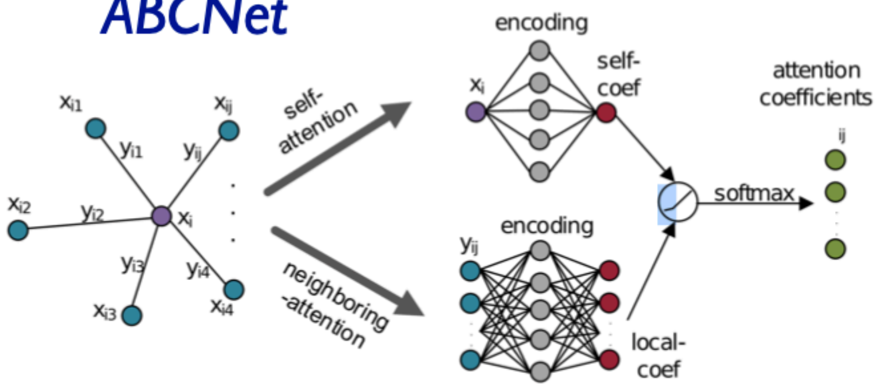
disclaimer: only shows a part of relevant works

➔ 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

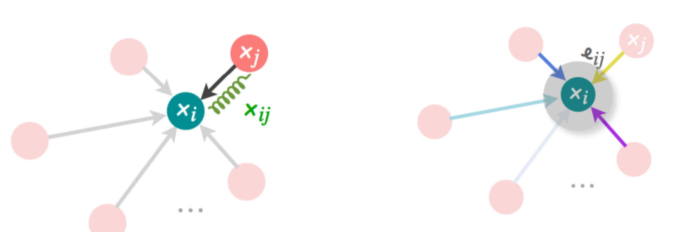


add attention mechanism in addition to graph convolution

ParticleNeXt

H.Qu. Talk@ML4Jets2021

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



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

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

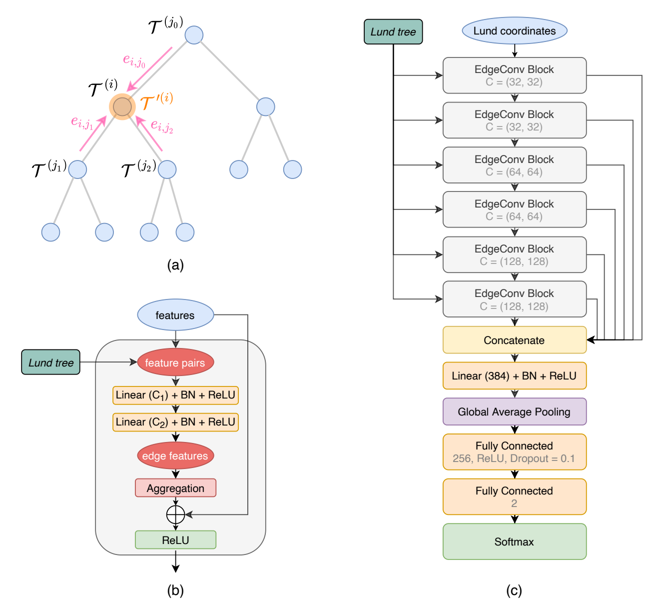
$$w_{ij} = softmax_j(attn_{ij})$$

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

physics-inspired design/modifications

LundNet

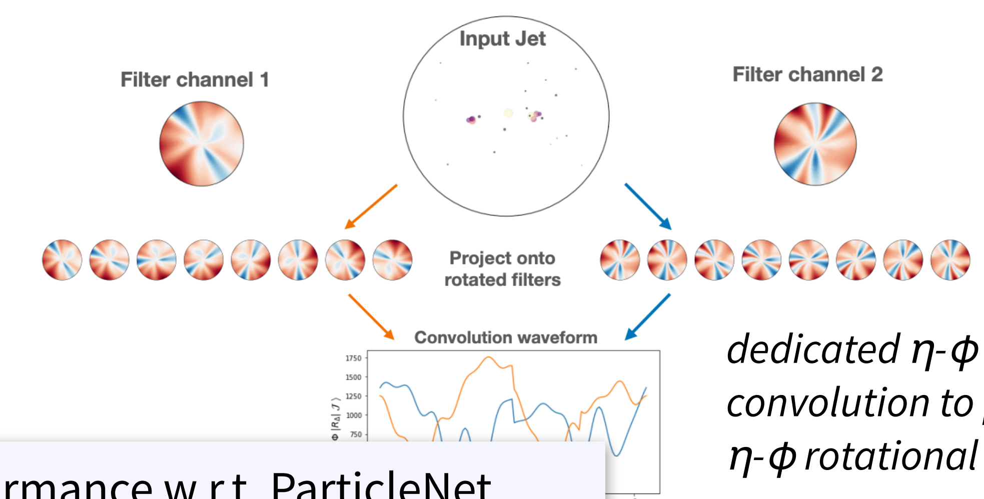
F.Dreyer et al. JHEP 03 (2021) 052



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

C.Shimmin. arXiv:2107.02908

Particle Convolution Network

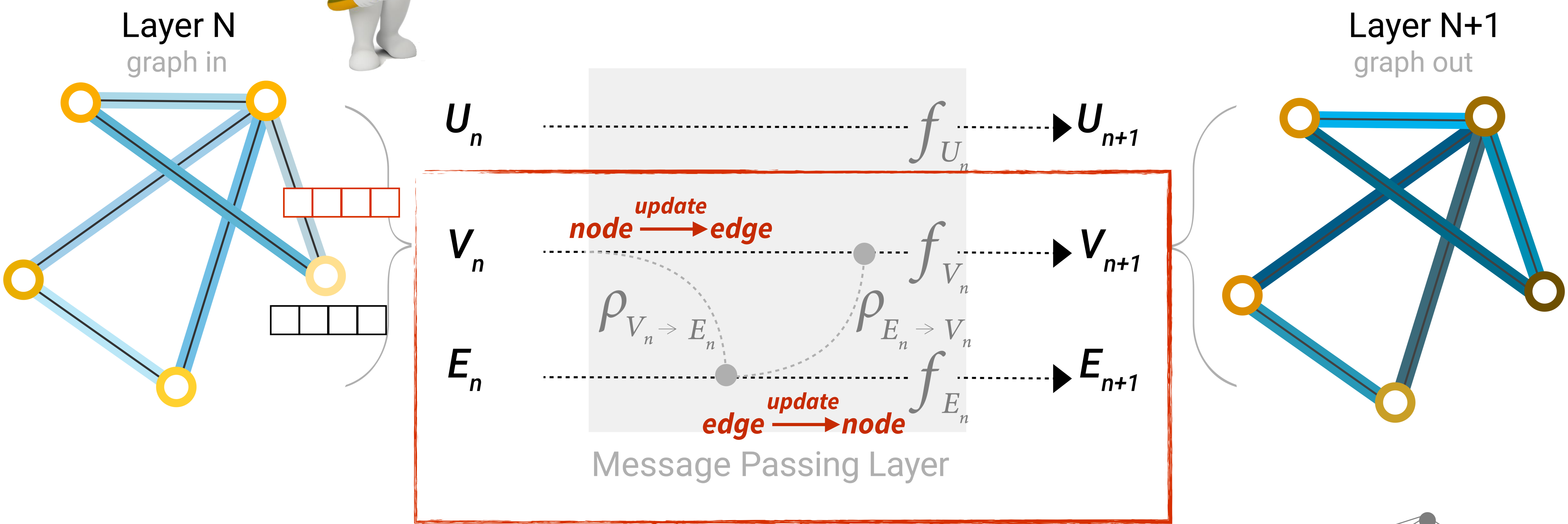


dedicated η-ϕ convolution to preserve η-ϕ rotational symmetry

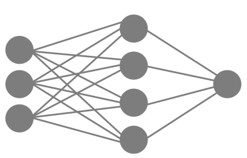
small improvement or comparable performance w.r.t. ParticleNet, evaluated on two mainstream benchmarks

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 ρ

Brief intro to ParT

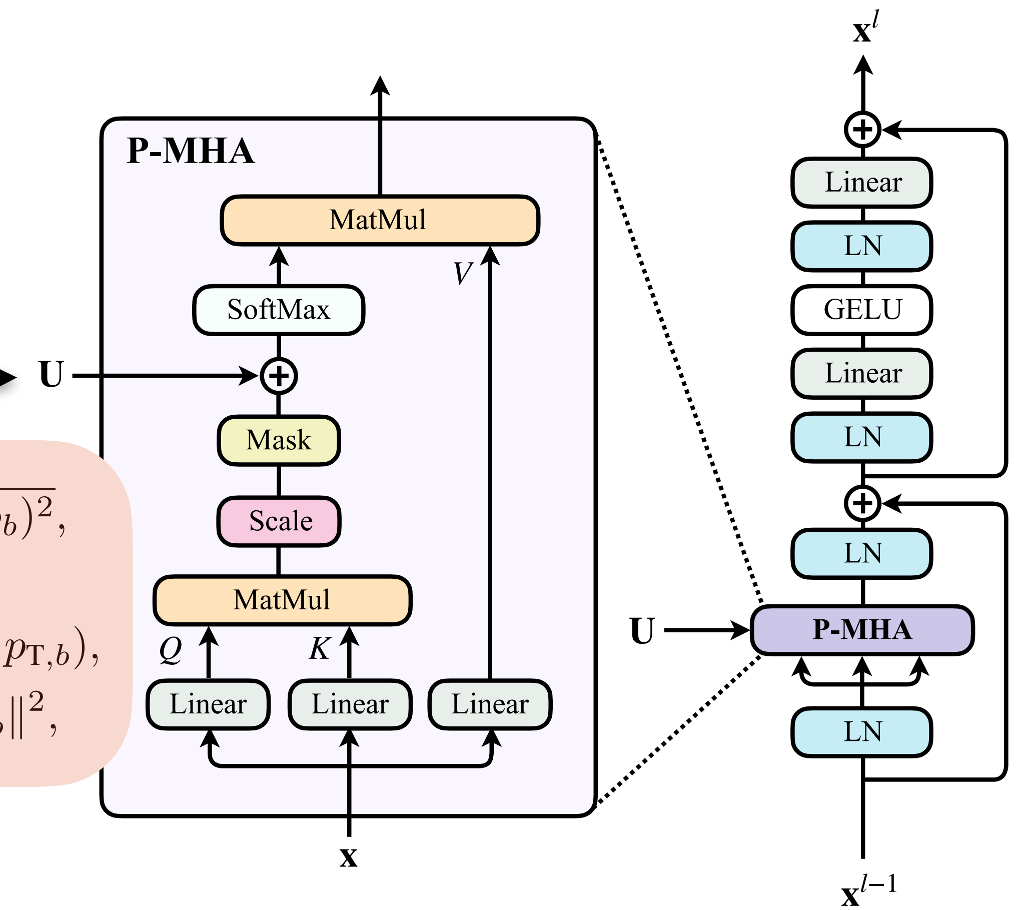
- → Transformer model is the new state-of-the-art architecture introduced in DL community
  - ❖ Language models: BERT, GPT-3…
  - ❖ Computer Vision: ViT, Swin-T
  - ❖ AI for Science: AlphaFold2 for protein structure prediction
- → Transformers architecture
  - ❖ consists only of self-attention blocks
  - ❖ 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		$H \to b\bar{b}$	$H \to c\bar{c}$	$H \to gg$	$H \to 4q$	$H \to \ell\nu qq'$	$t \to bqq'$	$t \to b\ell\nu$	$W \to qq'$	$Z \to q\bar{q}$
	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

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



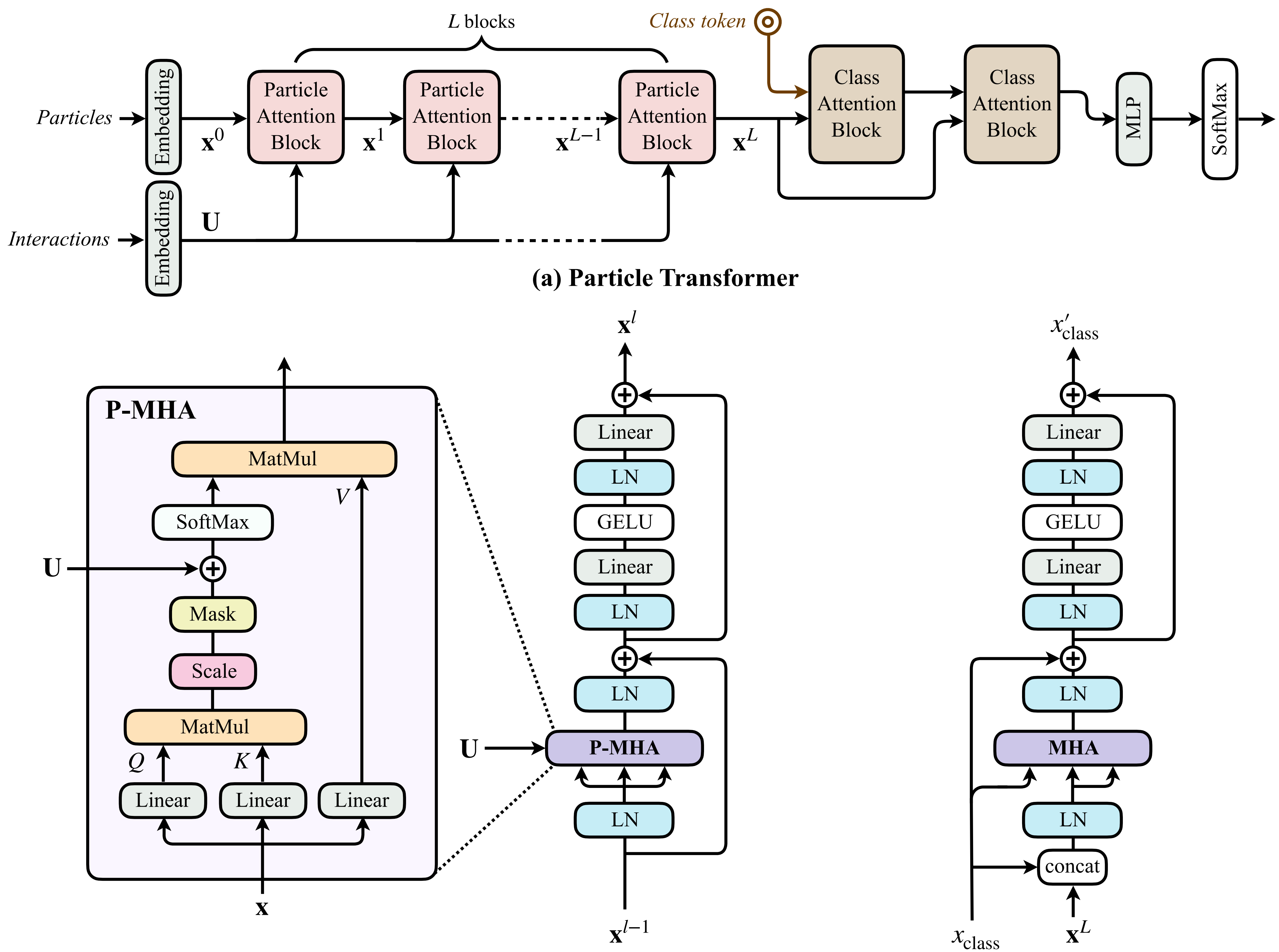
(b) Particle Attention Block

	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!

ParT architecture

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

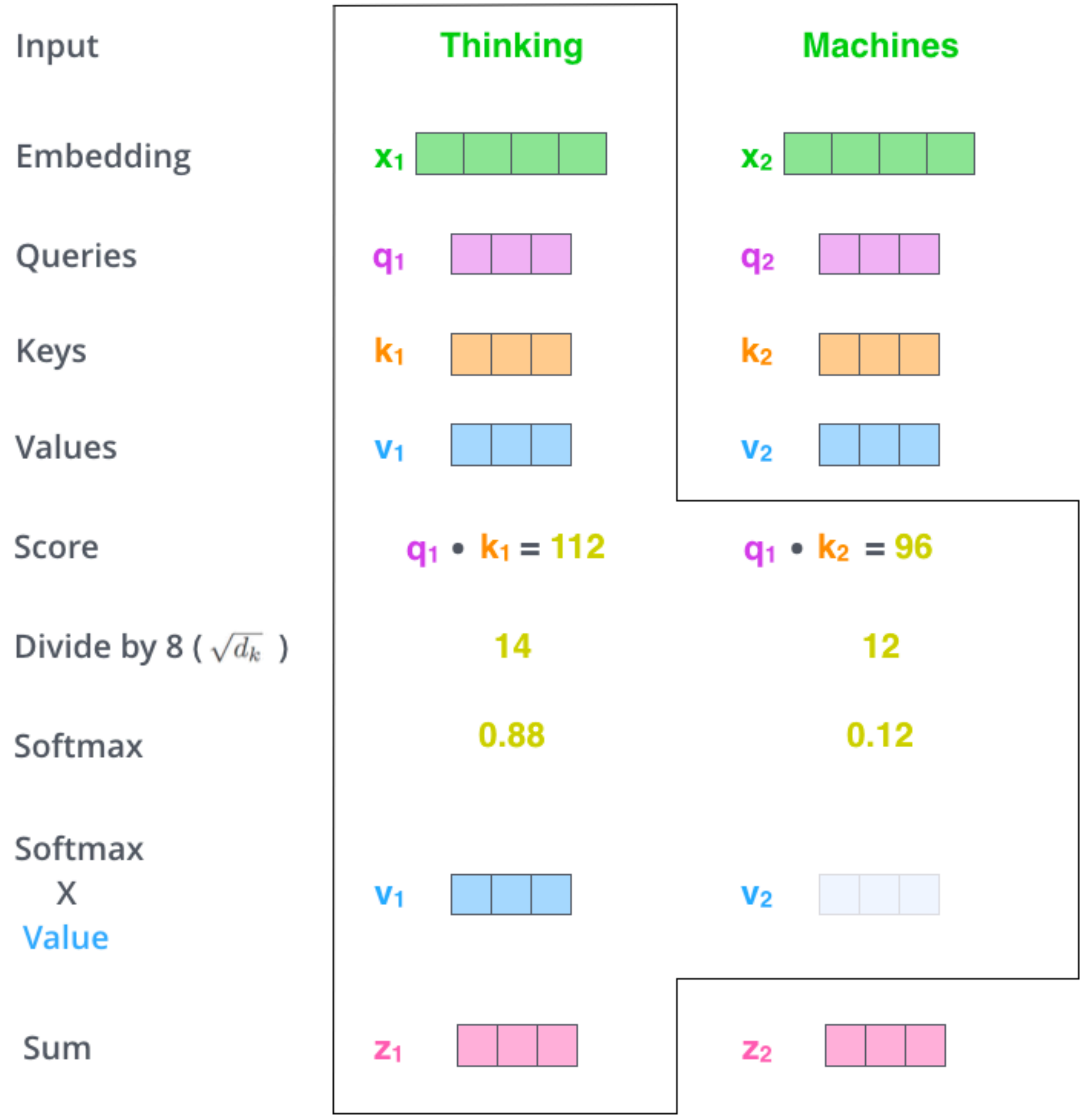


(a) Particle Transformer

(b) Particle Attention Block

(c) Class Attention Block

Transformer illustration



[image from link]