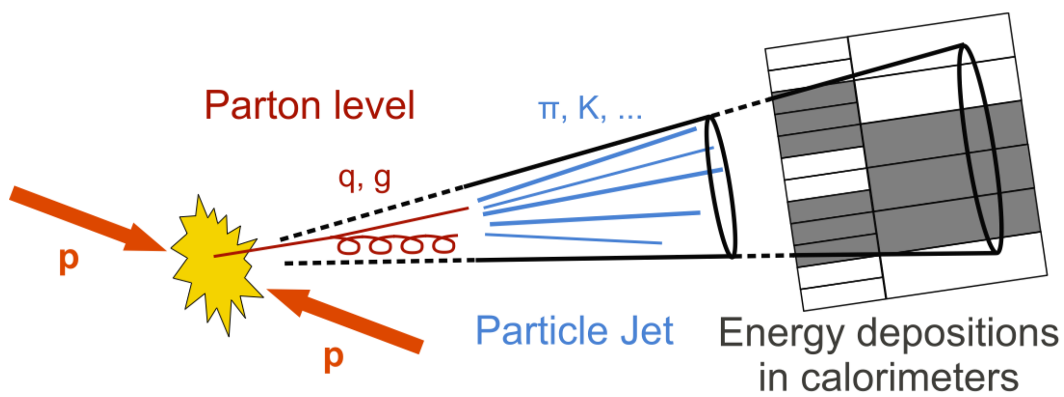
Graph Neural Networks for Jet Tagging

Huilin Qu (曲慧麟)

CERN

CEPC Workshop 2021 — November 11, 2021

INTRODUCTION



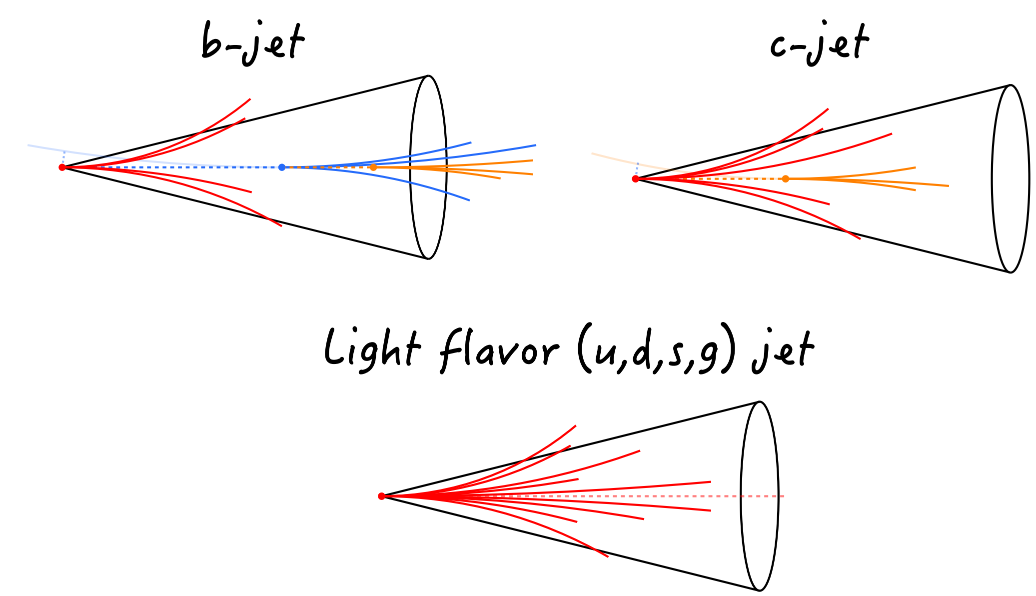
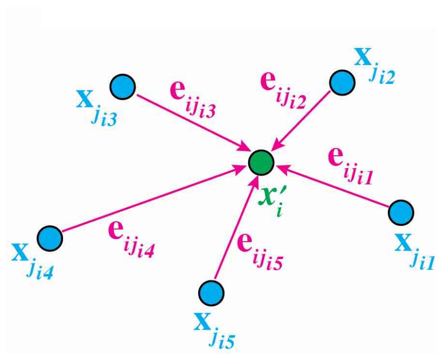


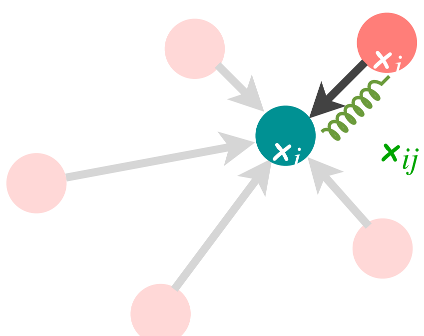
Image credit

- Jet: collimated spray of hadrons initiated by energetic quarks or gluons
  - a proxy to access the properties of the q/g produced in the hard scattering
    - fundamental building blocks for object / event reconstruction: H->bb, H->cc, H->ss, H->gg, …
  - study of jet production and jet internal structures improves our understanding of QCD
- Jet tagging: identifying the hard scattering particle that initiates the jet
  - heavy flavor tagging: bottom vs charm vs u/d/s/g
  - heavy resonance tagging: top/W/Z/Higgs
- Recent years: the rise of machine learning (ML) has brought lots of new progresses
  - novel approaches and techniques lead to significant improvement in performance, and also deeper insights into jet physics

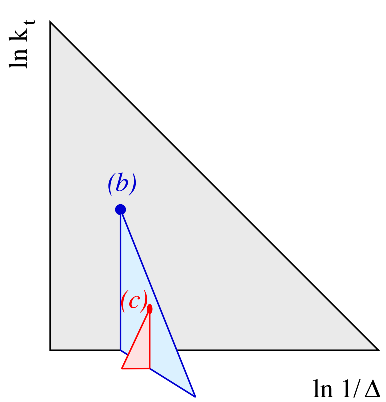
OUTLINE



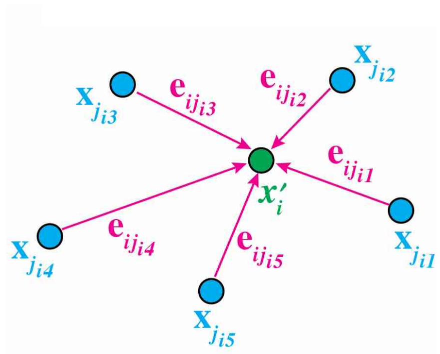
ParticleNet: jet tagging via particle clouds



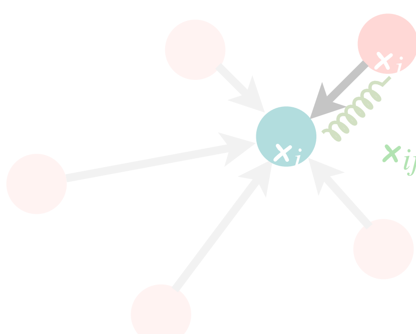
ParticleNeXt: pushing the limit of jet tagging



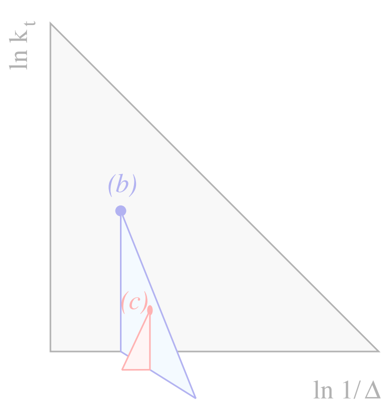
LundNet: graph neural network in the Lund plane



ParticleNet: jet tagging via particle clouds

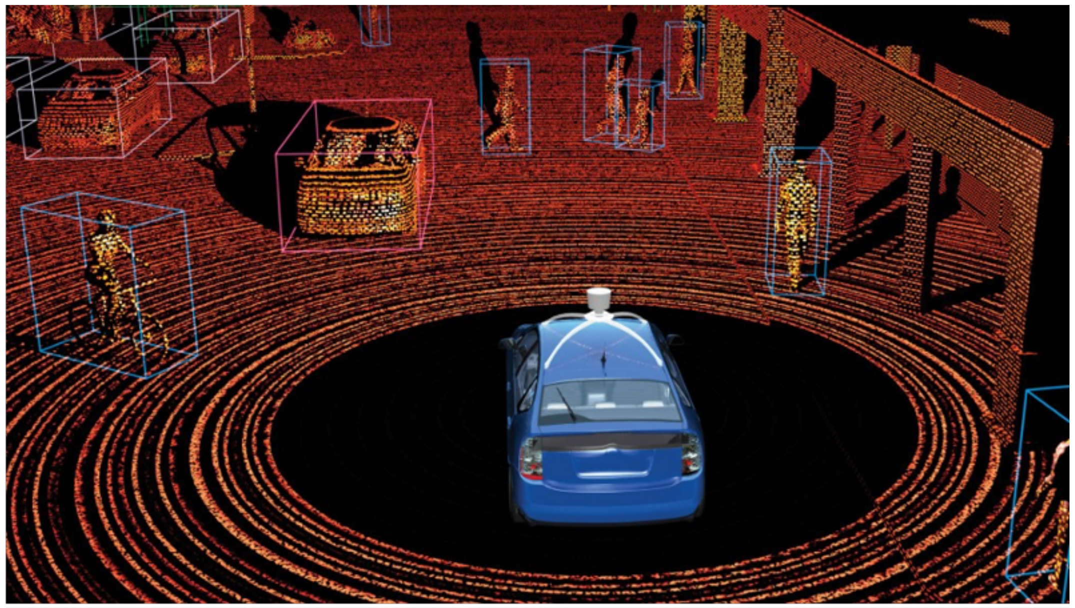


ParticleNeXt: pushing the limit of jet tagging



LundNet: graph neural network in the Lund plane

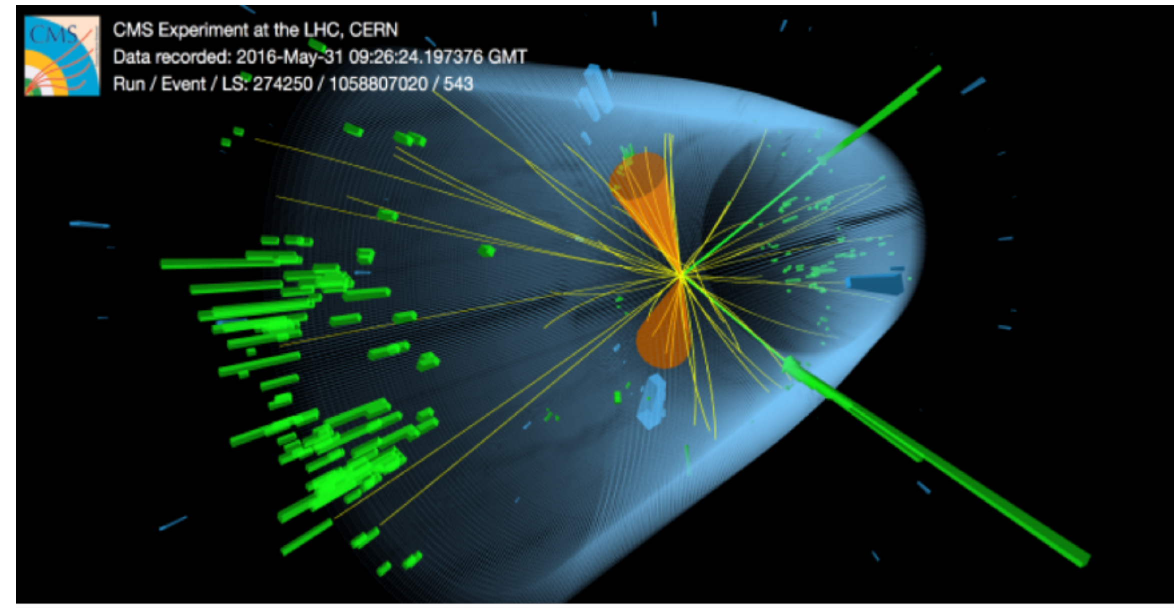
Jet As a Point Cloud



Point cloud

From Wikipedia, the free encyclopedia

A **point cloud** is a set of data points in space. Point clouds are generally produced by 3D scanners, which measure a large number of points on the external surfaces of objects around them.

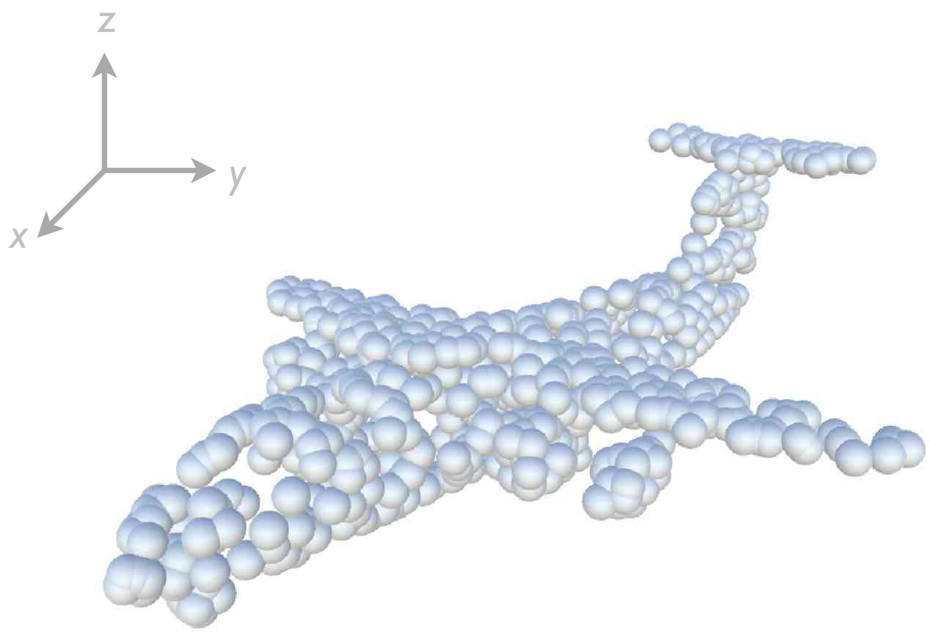


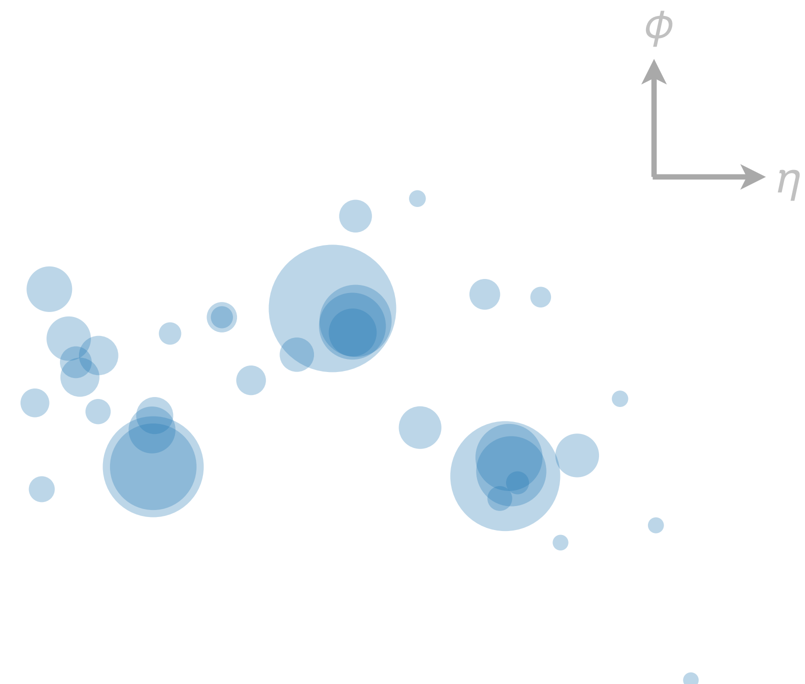
Jet (Particle cloud)

From Wikipedia, the free encyclopedia

A **jet (particle cloud)** is a set of particles in space. Particle clouds are generally created by clustering a large number of particles measured by particle detectors, e.g., ATLAS EXPERIMENT and CMS.

Point vs Particle Clouds





- Point cloud
  - points are intrinsically ***unordered***
  - points are distributed in space
    - spatial coordinates (3D xyz) encode geometric structure information

- Particle cloud
  - particles are intrinsically ***unordered***
  - particles are distributed in space
    - spatial distribution (2D coordinates in the η-ϕ space) reflects radiation patterns

But particles have many more features:

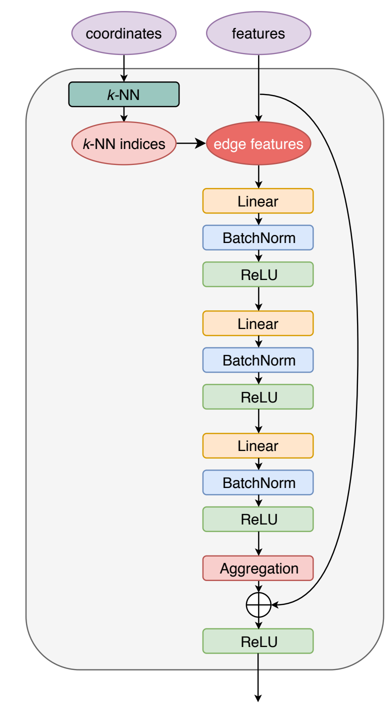
- *energy/momenta/displacement/particle ID/etc.*
- *more interesting than a plain point cloud!*

ARCHITECTURE: *PARTICLENET*

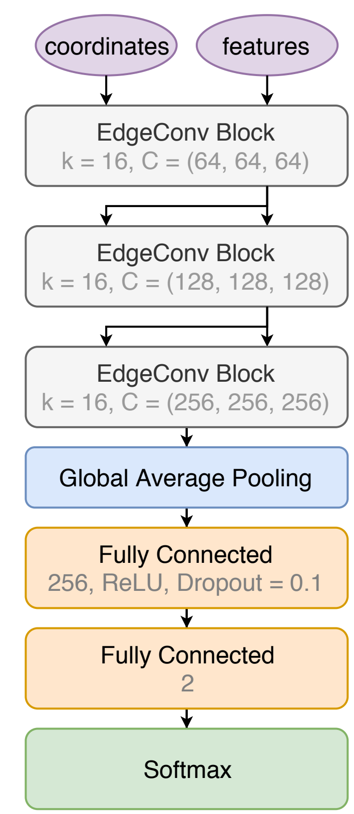
HQ and L. Gouskos
[*Phys.Rev.D 101 (2020) 5, 056019*]

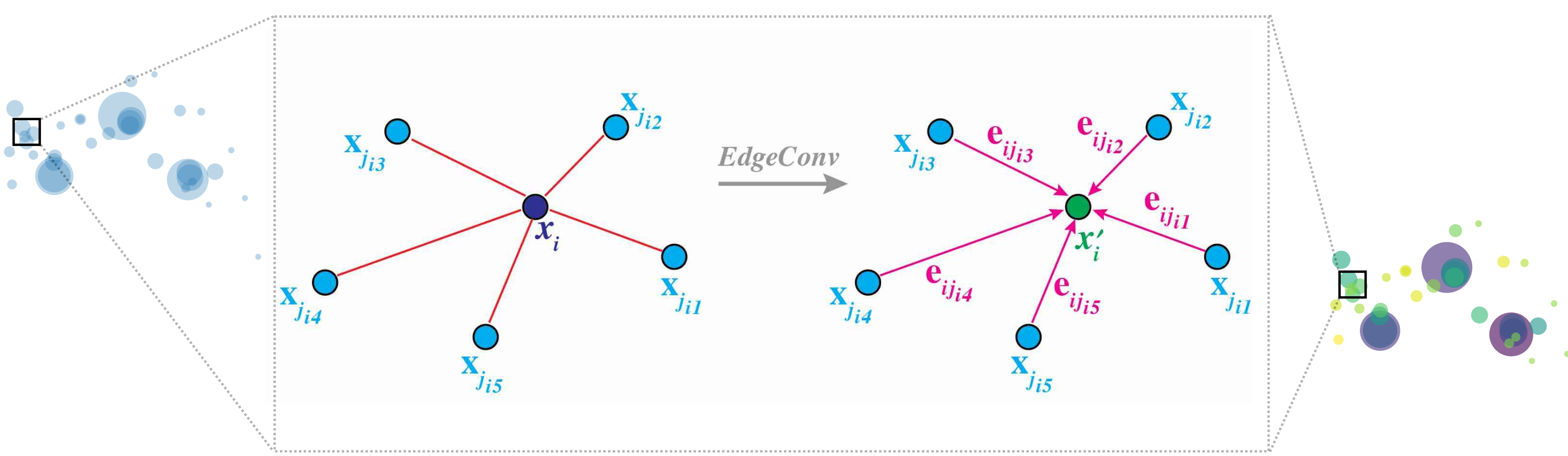
- ParticleNet
  - customized graph neural network architecture for jet tagging with the point cloud approach, based on Dynamic Graph CNN [Y. Wang et al., *arXiv:1801.07829*]
  - explicitly respects the permutation symmetry of the point cloud
- Key building block: *EdgeConv*
  - treating a point cloud as a graph: each point is a vertex
    - for each point, a local patch is defined by finding its k-nearest neighbors
  - designing a permutation-invariant "convolution" function
    - define "edge feature" for each center-neighbor pair: $e_{ij} = h_\Theta(x_i, x_j)$
      - same h_Θ for all neighbor points, and all center points, for symmetry
    - aggregate the edge features in a symmetric way: $x_i' = \mathrm{mean}_j\, e_{ij}$
- EdgeConv can be stacked to form a deep network
  - learning both local and global structures, in a hierarchical way

EdgeConv block



ParticleNet architecture





Performance of ParticleNet

- Performance on the public top tagging benchmark dataset
  - ParticleNet achieves the highest performance among all algorithms

G. Kasieczka et al. [SciPost Phys. 7 (2019) 014]

Architecture used by DeepAK8

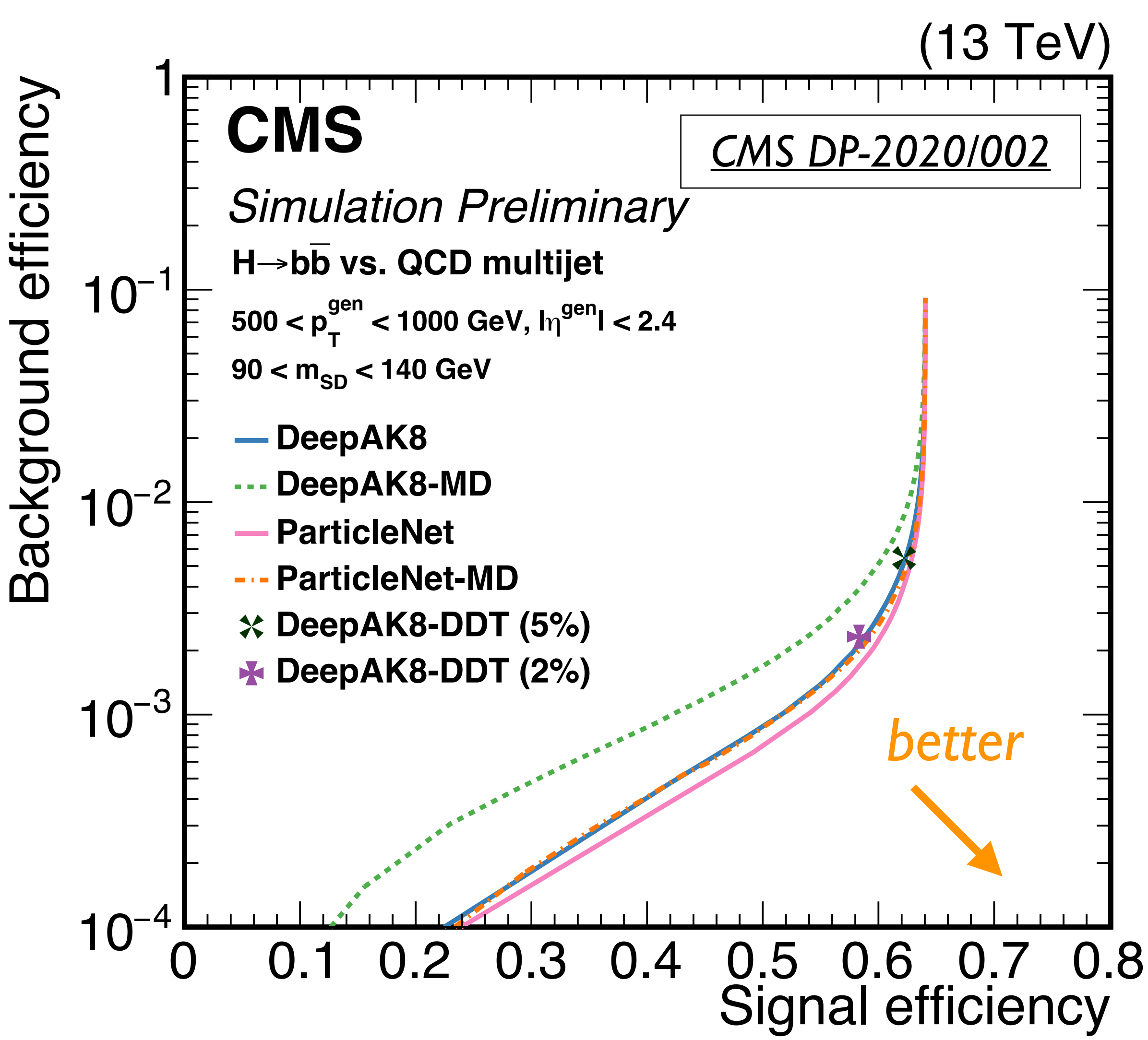
Ensemble of all taggers

	AUC	Acc	$1/\epsilon_B$ ($\epsilon_S = 0.3$) single	mean	median	#Param
CNN [16]	0.981	0.930	914±14	995±15	975±18	610k
ResNeXt [30]	0.984	0.936	1122±47	1270±28	1286±31	1.46M
TopoDNN [18]	0.972	0.916	295±5	382± 5	378 ± 8	59k
Multi-body N-subjettiness 6 [24]	0.979	0.922	792±18	798±12	808±13	57k
Multi-body N-subjettiness 8 [24]	0.981	0.929	867±15	918±20	926±18	58k
TreeNiN [43]	0.982	0.933	1025±11	1202±23	1188±24	34k
P-CNN	0.980	0.930	732±24	845±13	834±14	348k
ParticleNet [47] (Preliminary ver.)	0.985	0.938	1298±46	1412±45	1393±41	498k
LBN [19]	0.981	0.931	836±17	859±67	966±20	705k
LoLa [22]	0.980	0.929	722±17	768±11	765±11	127k
Energy Flow Polynomials [21]	0.980	0.932	384			1k
Energy Flow Network [23]	0.979	0.927	633±31	729±13	726±11	82k
Particle Flow Network [23]	0.982	0.932	891±18	1063±21	1052±29	82k
GoaT	0.985	0.939	1368±140		1549±208	35k
ParticleNet-Lite	0.984	0.937	1262±49			26k
ParticleNet	0.986	0.940	1615±93			366k

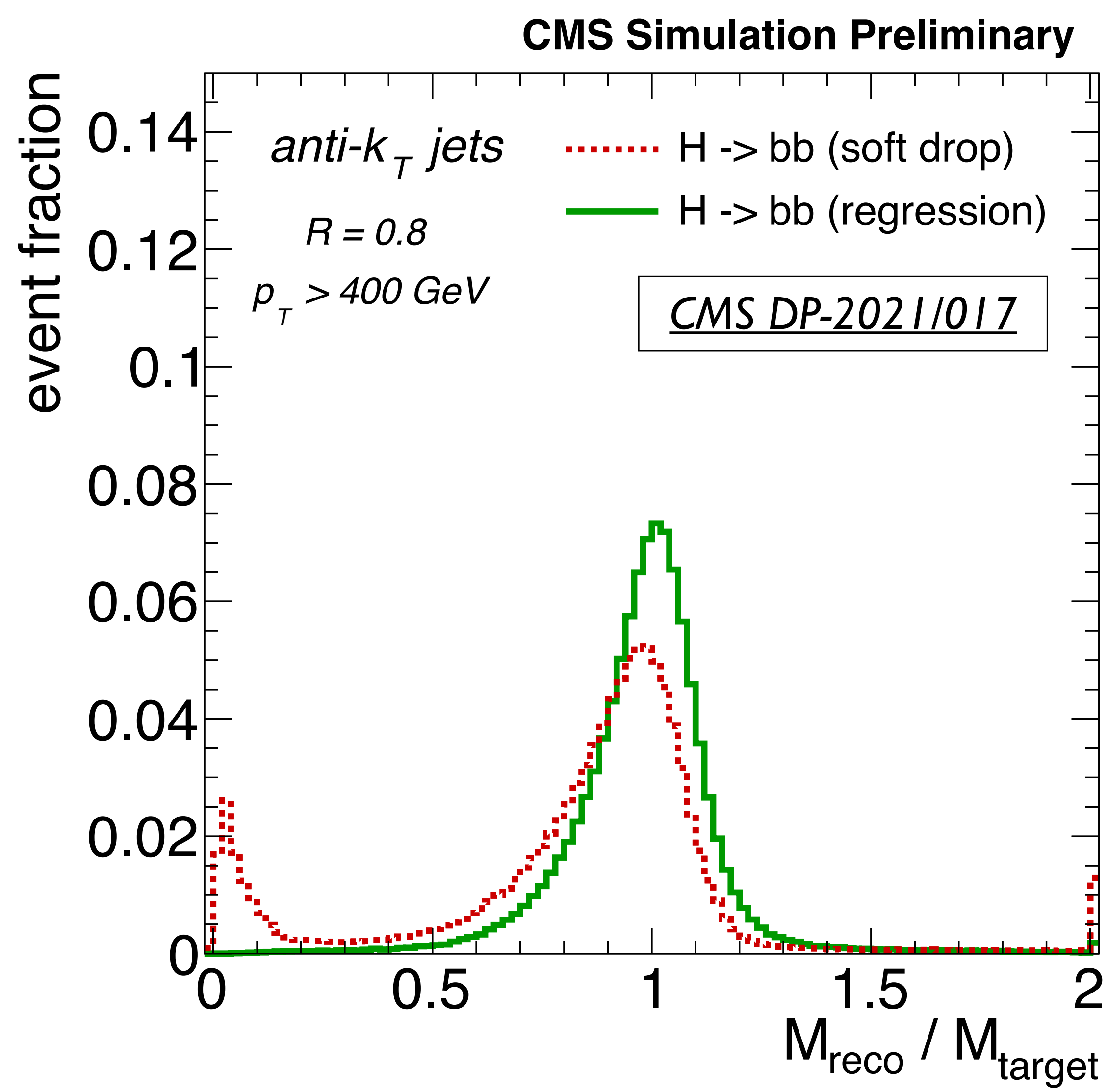
ParticleNet in Action

- ParticleNet has become a standard jet tagging algorithm in CMS

Tagging boosted H→bb jets



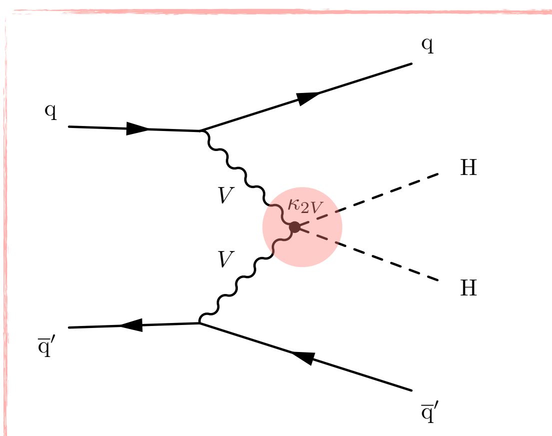
Mass regression for boosted H→bb jets

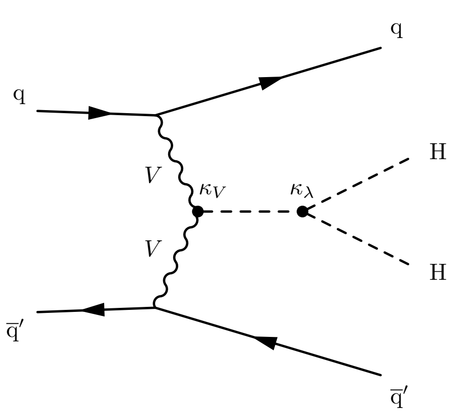


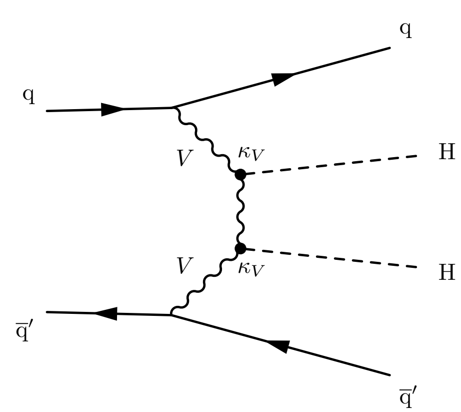
ParticleNet in Action (II)

- ParticleNet has become a standard jet tagging algorithm in CMS

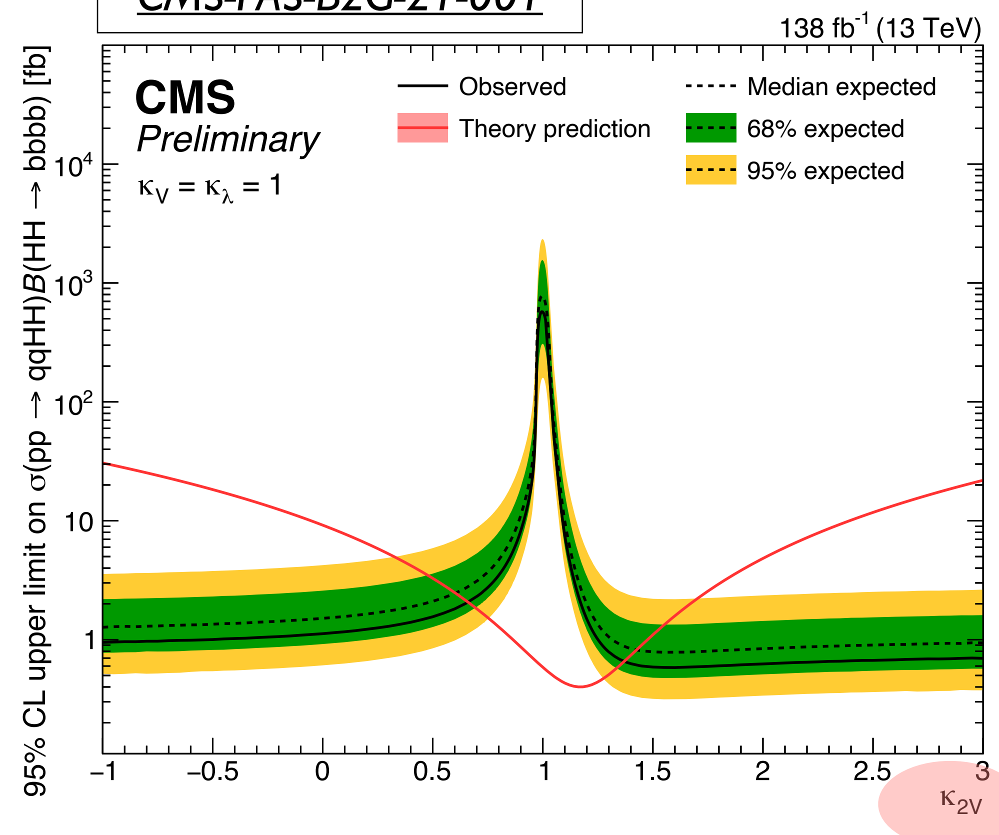
Search for VBF HH production





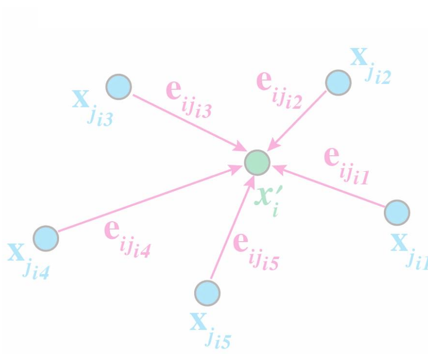


CMS-PAS-B2G-21-001

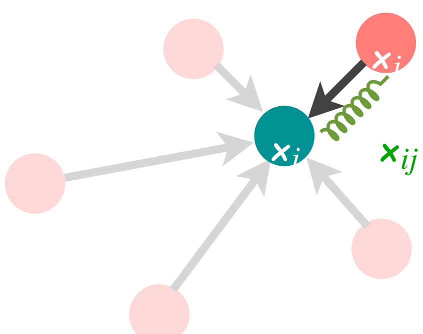


- Most stringent constraint on κ_{2V} to date
  - **0.6 < κ_{2V} < 1.4**
  - κ_{2V} = 0 **excluded for the first time!**

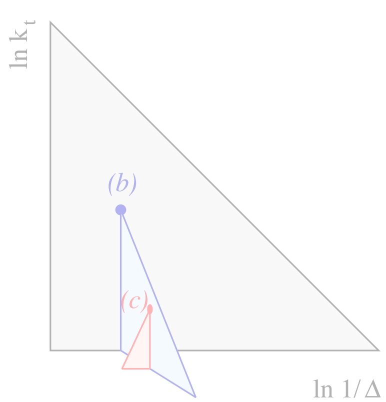
- ParticleNet also being explored for detector design studies for future lepton colliders
  - see e.g., talk by Michele Selvaggi at this workshop



ParticleNet: jet tagging via particle clouds



ParticleNeXt: pushing the limit of jet tagging



LundNet: graph neural network in the Lund plane

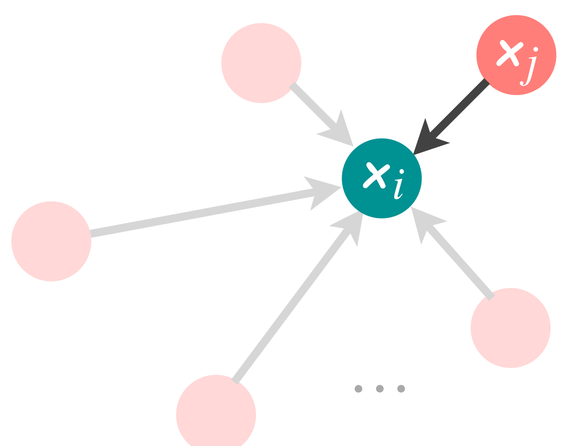
ParticleNeXt: Pairwise Features

- ParticleNeXt: next-generation of ParticleNet, for better performance

HQ [ML4Jets2021]

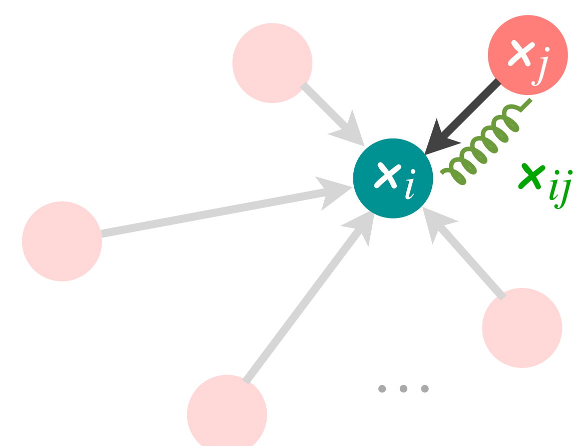
- The first enhancement is the addition of (explicit) pairwise features on the edges

ParticleNet



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

ParticleNeXt



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

- Examples of pairwise features:

$$\Delta_{ij}^2 \equiv (y_i - y_j)^2 + (\phi_i - \phi_j)^2, \quad m^2 \equiv (p_i + p_j)^2,$$

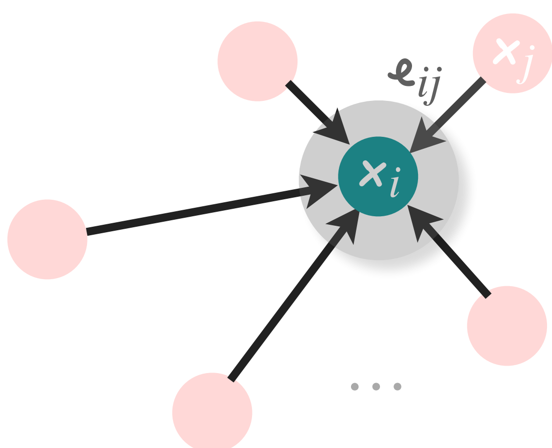
$$k_T \equiv \min(p_{T,i}, p_{T,j})\,\Delta_{ij}, \quad z \equiv \frac{\min(p_{T,i}, p_{T,j})}{p_{T,i} + p_{T,j}}$$

(use the logarithm to improve stability of the training)

ParticleNeXt: Attentive Pooling

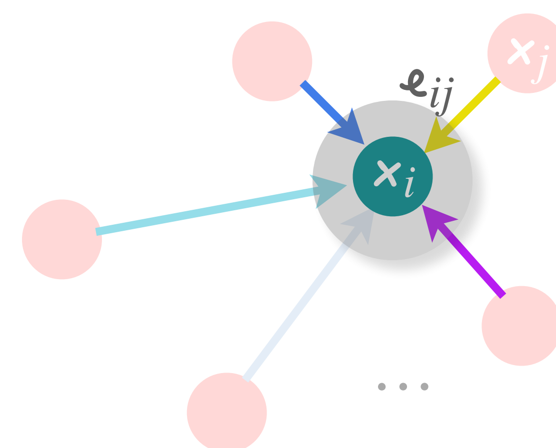
- Use attention-based pooling to increase the expressive power
  - for both the **local neighborhood pooling**, and the final **global pooling**

ParticleNet



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

ParticleNeXt



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

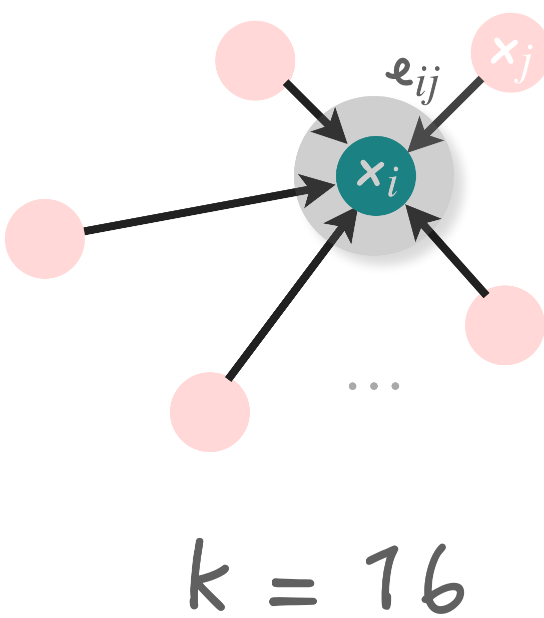
$$w_{ij} = \mathrm{softmax}_j(attn_{ij})$$

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

ParticleNeXt: Multi-Scale Aggregation

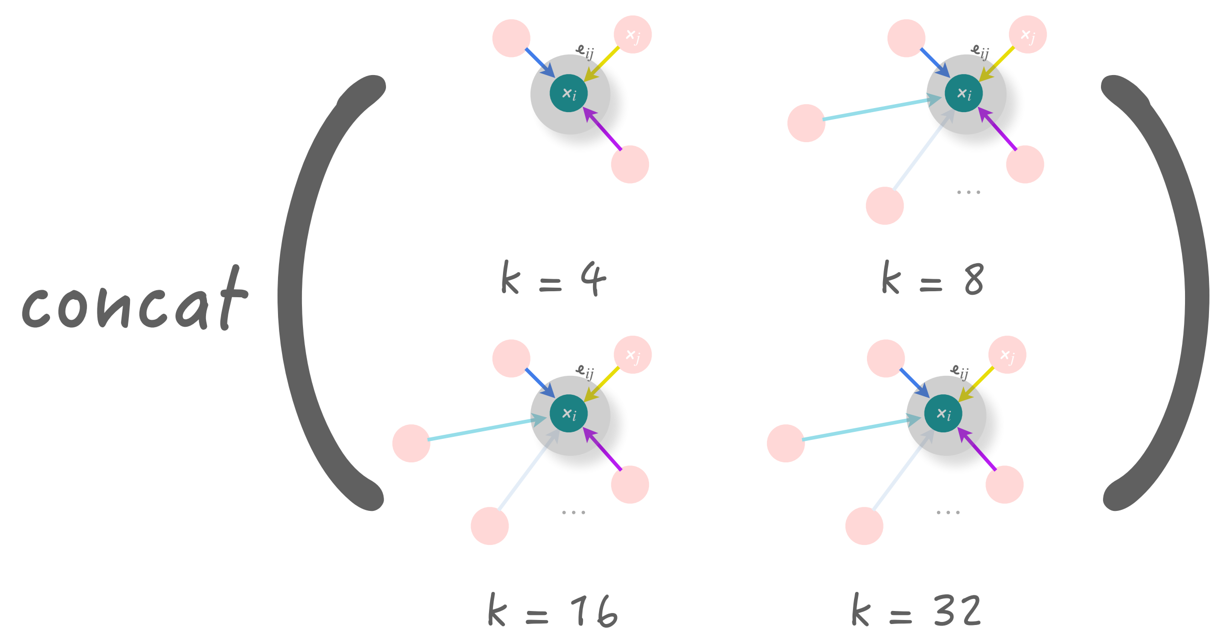
- Introduce multi-scale aggregation to better capture both short- and long-range correlations
  - perform local aggregation for the 4, 8, 16 and 32 nearest neighbors (with different attentive pooling) and combine the 4 aggregated representations with a MLP
  - on the other hand: remove dynamic kNN (based on learned features), i.e., use only kNN in $\eta-\phi$ space, to reduce computational cost
    - in this case the kNN needs to be performed only once, and then the graph connectivity is fixed

ParticleNet



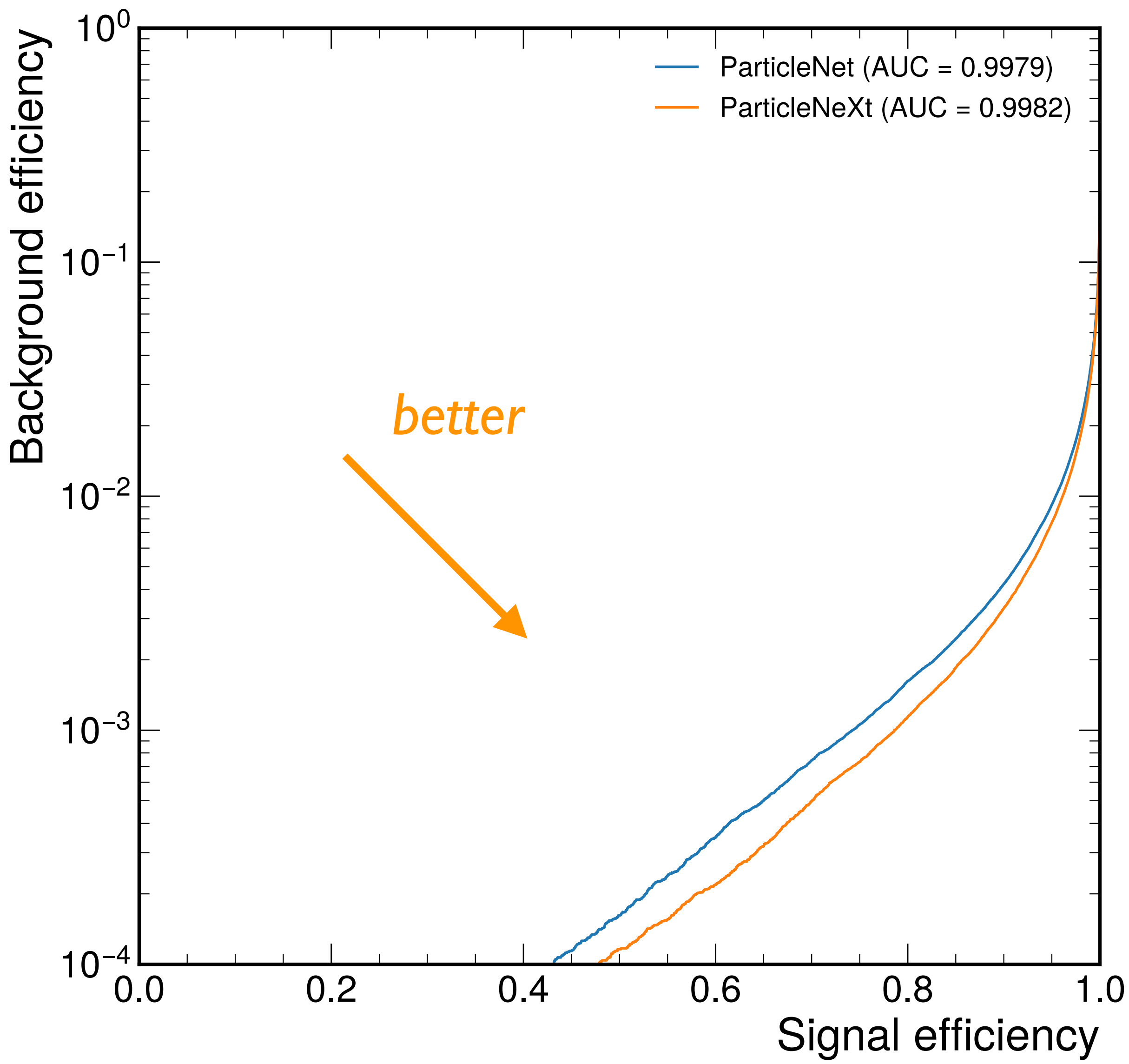
$$x_i' = z_i$$

ParticleNeXt



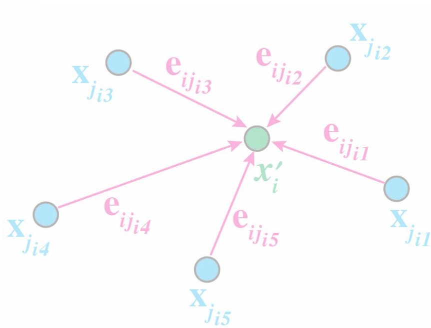
$$x_i' = \mathrm{MLP}(z_i^{concat})$$

PERFORMANCE: TOP TAGGING

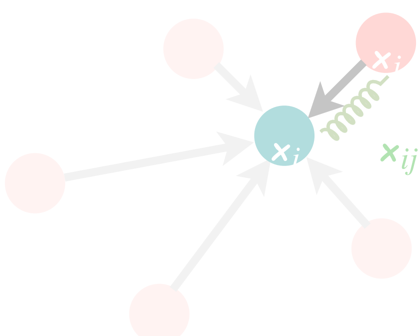


- Delphes simulation w/ CMS-like detector
- Training/validation/test splitting:
  - 1.6M / 0.4M / 2M
- Training repeated for 3 times starting from randomly initialized weights
  - the median-accuracy training is reported, and the standard deviation of the 3 trainings is quoted as the uncertainty
- Significant improvement in background rejection w/ ParticleNeXt
  - ~50% higher BKG rejection (@ϵ_S = 70%)
  - computational cost still under control

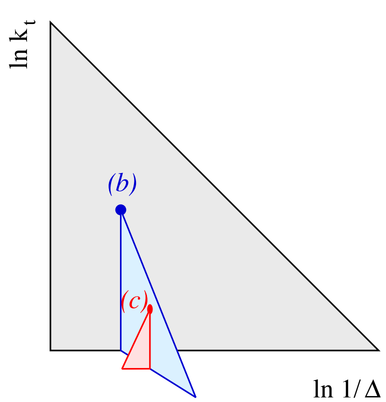
	Accuracy	AUC	$1/\varepsilon_b$ at		Parameters	Inference time		Training time
			$\varepsilon_s = 70\%$	$\varepsilon_s = 50\%$		(CPU)	(GPU)	(GPU)
ParticleNet	0.980	0.9979	1342 ± 4	6173 ± 425	366k	23 ms	0.30 ms	1.0 ms
ParticleNeXt	**0.981**	**0.9982**	$\mathbf{2008 \pm 75}$	$\mathbf{8621 \pm 309}$	560k	30 ms	0.54 ms	1.7 ms



ParticleNet: jet tagging via particle clouds



ParticleNeXt: pushing the limit of jet tagging

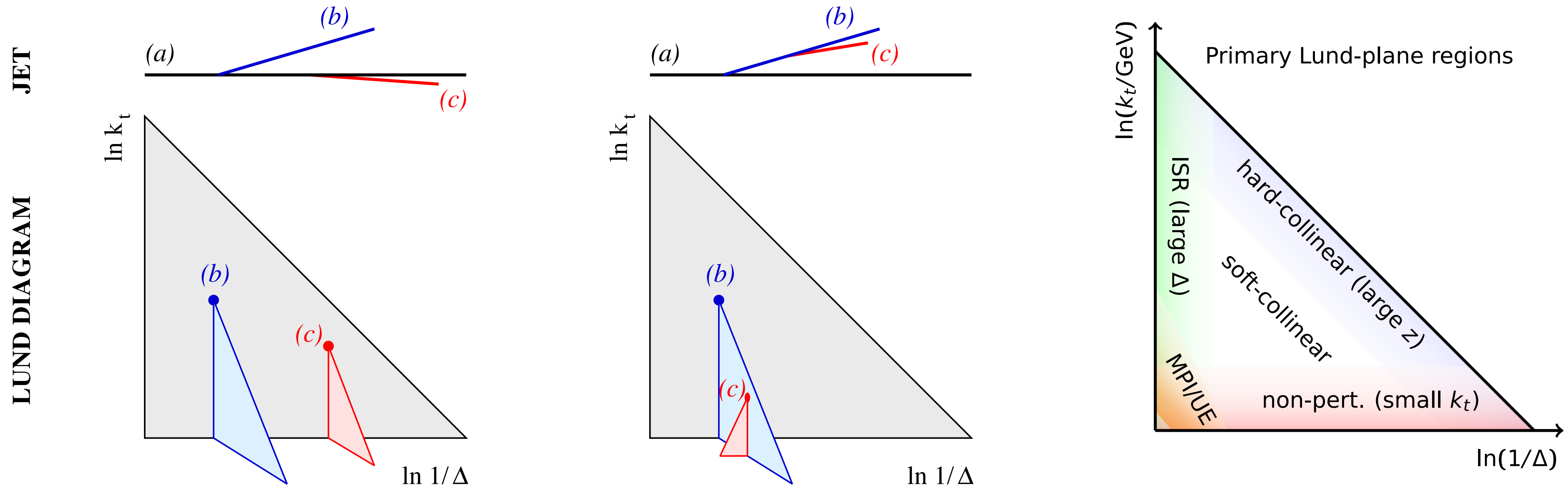


LundNet: graph neural network in the Lund plane

LundNet

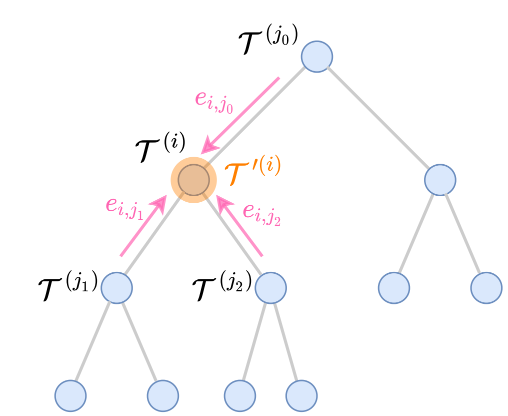
F. Dreyer and HQ
[JHEP 03 (2021) 052]

- The Lund jet plane provides an efficient description of the radiation patterns within a jet



  - each emission (splitting) is mapped to a point in the 2D (angle, transverse momentum) plane
    - further emissions (of the secondary particles) are represented in additional leaf planes
  - different kinematic regimes are clearly separated in the Lund plane
  - a natural input for ML algorithms on jets since it essentially encodes the full radiation patterns of a jet
- LundNet: a graph neural network based on the Lund plane representation of a jet
  - technically, the input is a binary tree (from Cambridge/Aachen clustering) <=> equivalent to the full Lund plane
  - for each node, a set of variables are be defined for the current splitting

$$\Delta^2 = (y_a - y_b)^2 + (\phi_a - \phi_b)^2, \quad k_t \equiv p_{tb}\Delta_{ab}, \quad m^2 \equiv (p_a + p_b)^2,$$
$$z \equiv \frac{p_{tb}}{p_{ta}+p_{tb}}, \quad \kappa \equiv z\Delta\,, \quad \psi \equiv \tan^{-1}\frac{y_b - y_a}{\phi_b - \phi_a},$$

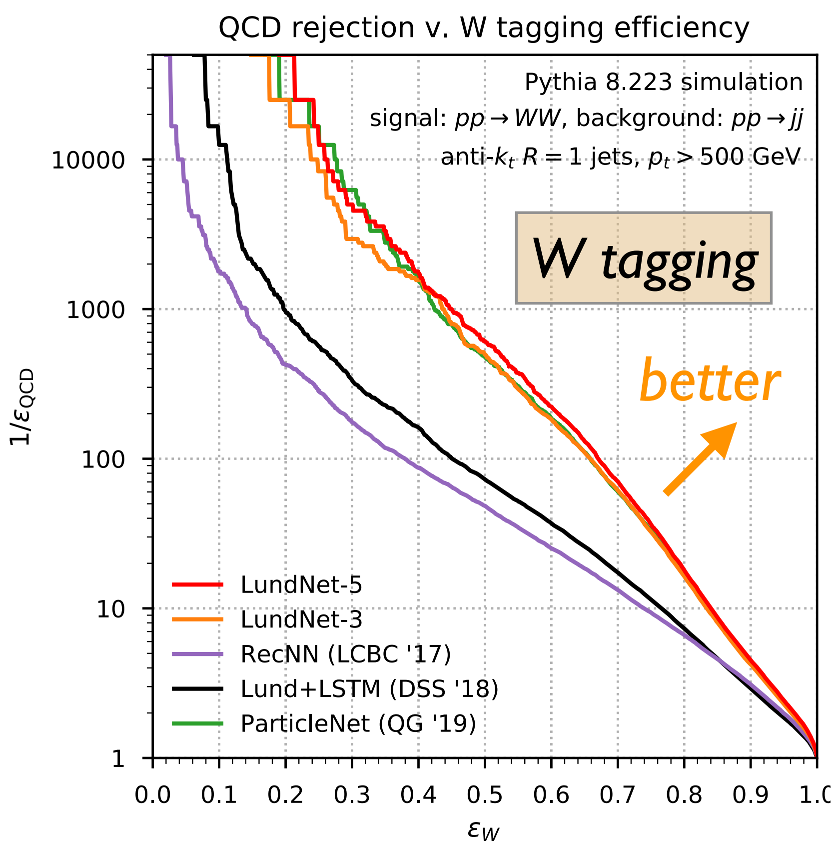


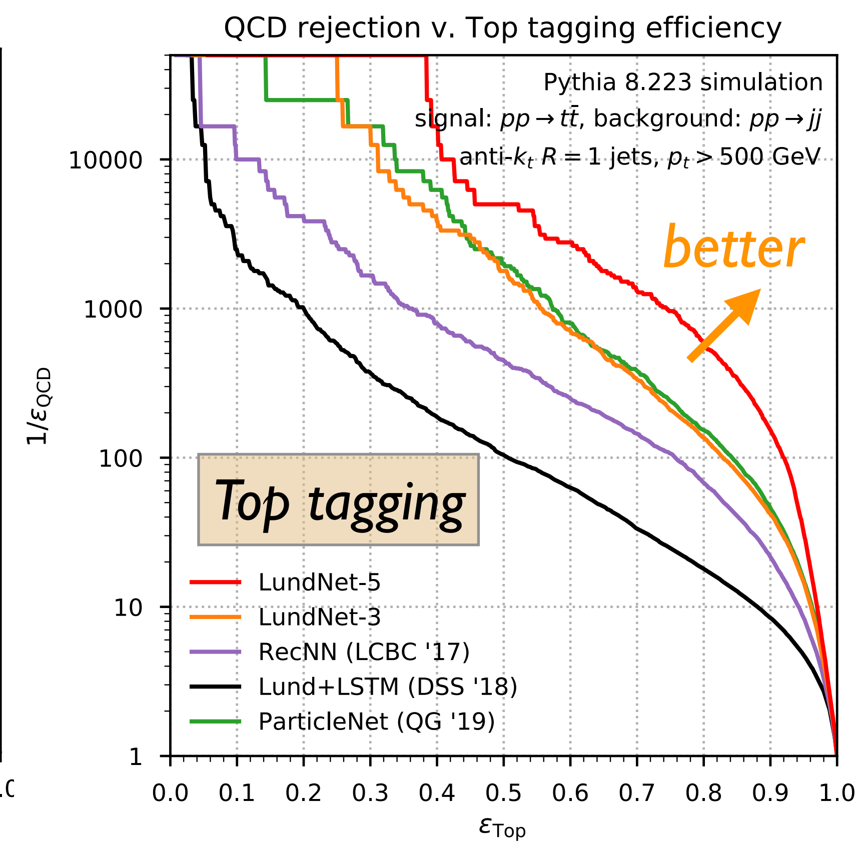
  - network architecture similar to ParticleNet, but the graph structure is fixed by the Lund tree (instead of kNN)

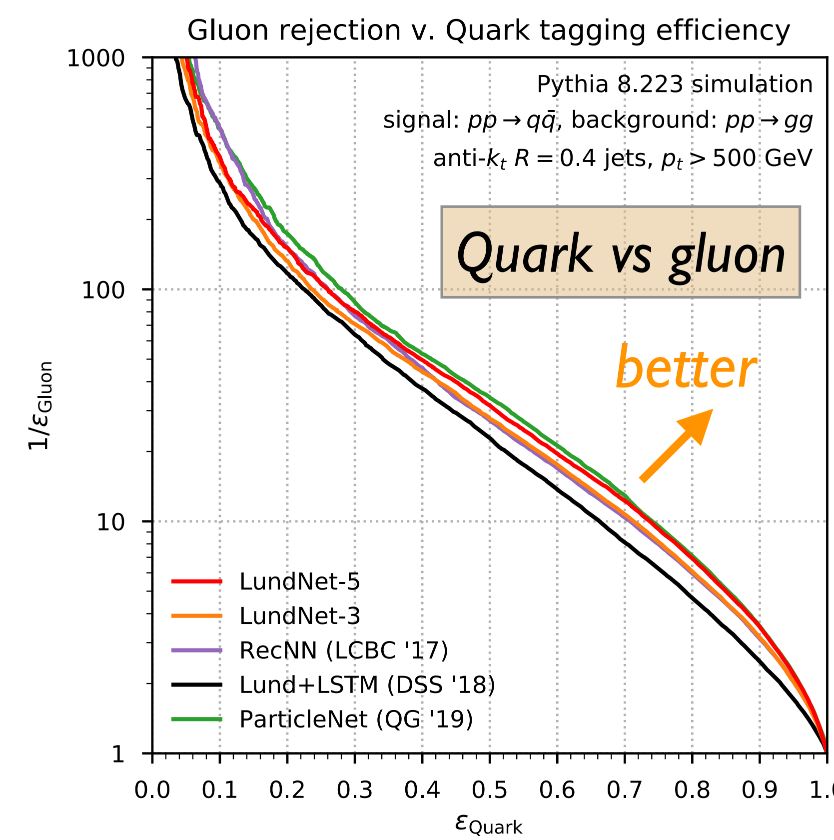
Performance of LundNet

F. Dreyer and HQ
[JHEP 03 (2021) 052]

- Significantly improved performance for top tagging compared to ParticleNet
  - similar performance for W tagging and q/g discrimination
- Almost an order of magnitude speed-up in training/inference time compared to ParticleNet







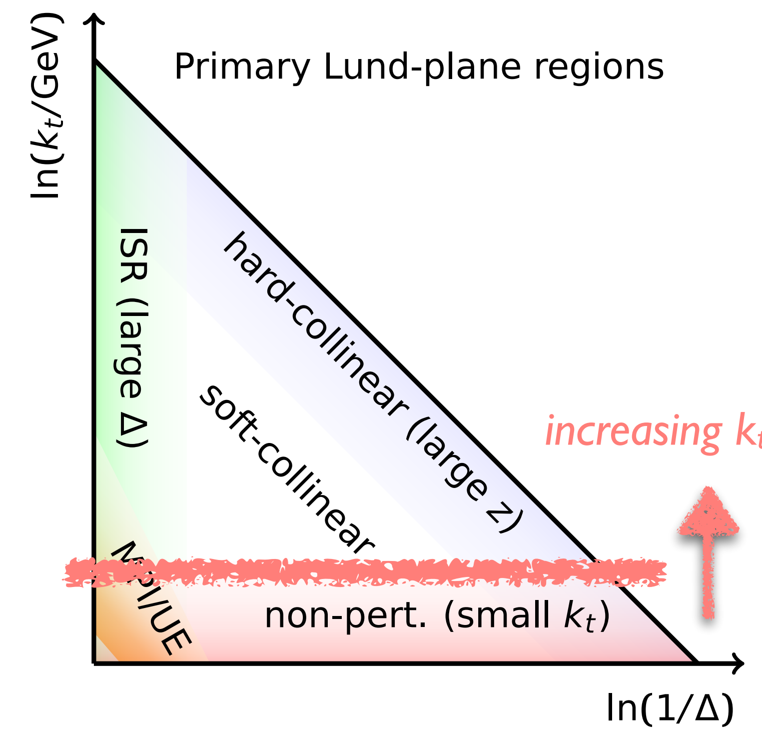
	Number of parameters	Training time [ms/sample/epoch]	Inference time [ms/sample]
LundNet	395k	0.472	0.117
ParticleNet	369k	3.488	1.036
Lund+LSTM	67k	0.424	0.131

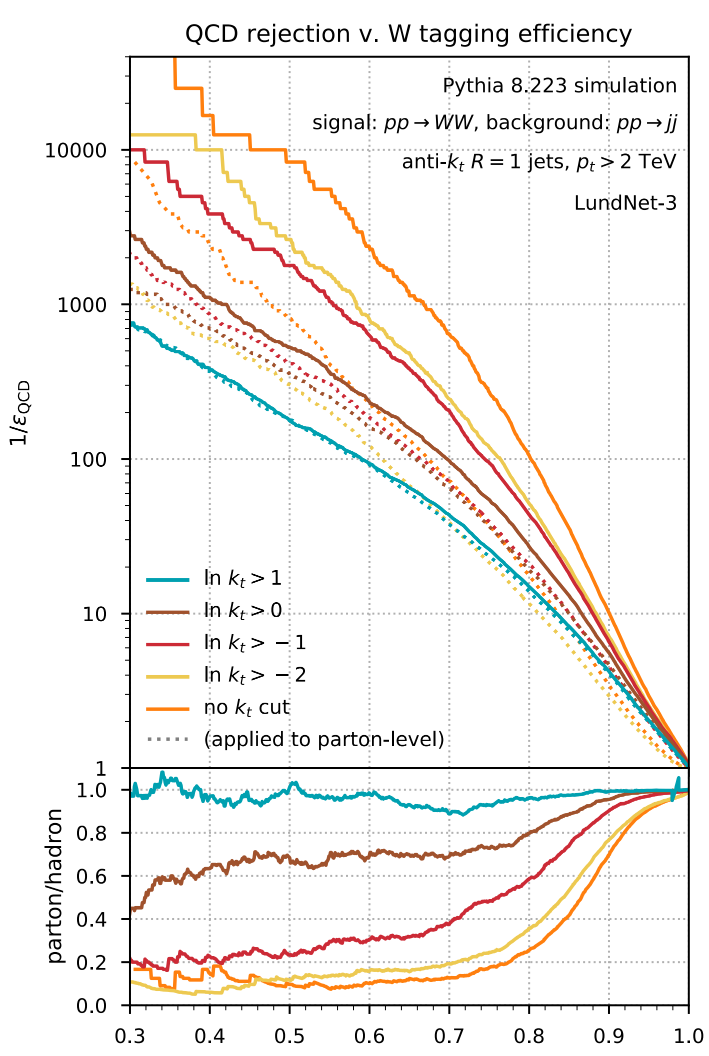
DGL + PyTorch
Nvidia GTX 1080Ti
batch size = 256

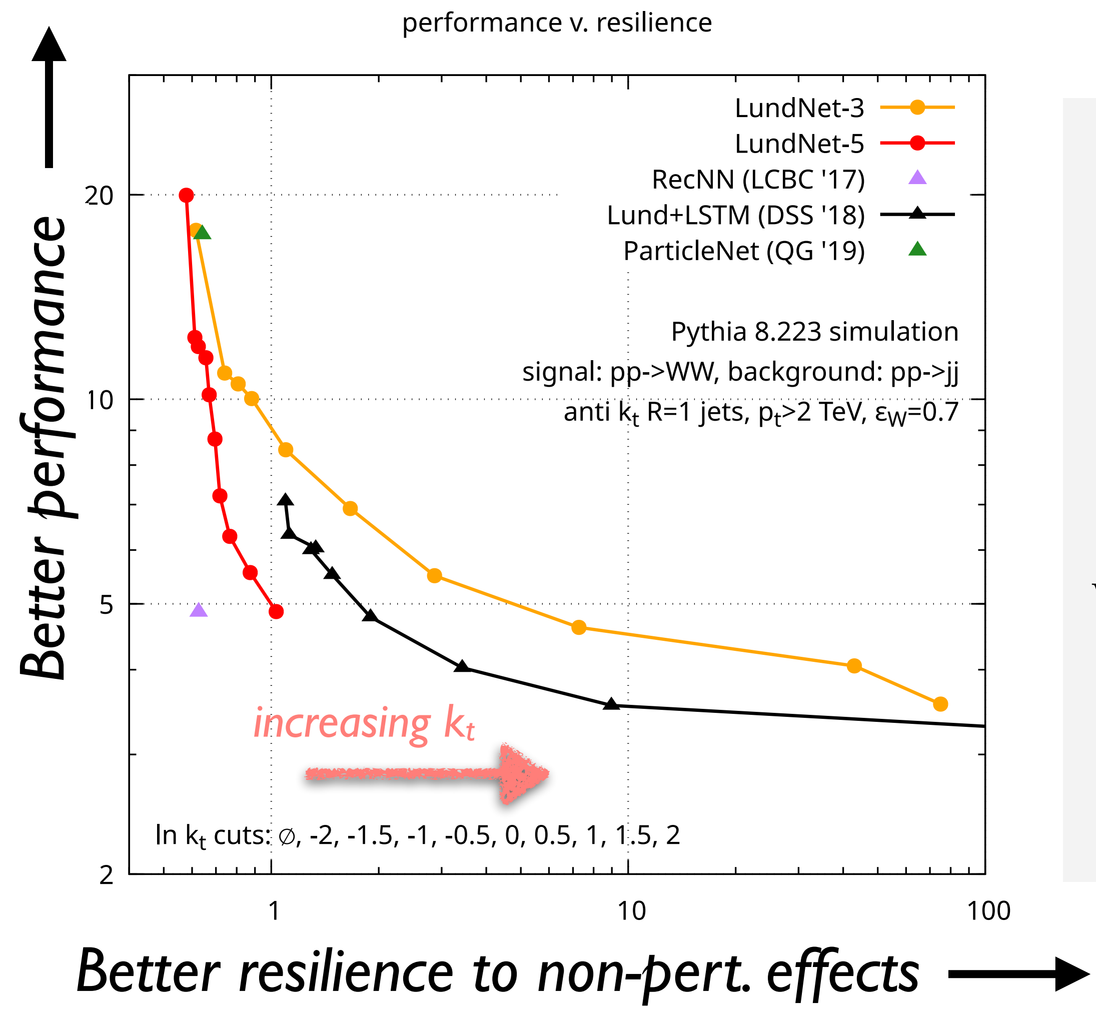
ROBUSTNESS OF LUNDNET

F. Dreyer and HQ [JHEP 03 (2021) 052]

- Moreover, LundNet provides a systematic way to control the robustness of the tagger
  - robustness assessed by applying the model trained on hadron-level samples to parton-level samples and compare the difference
  - the non-perturbative region can be effectively rejected by applying a k_t cut on the Lund plane, therefore improving the robustness of the tagger against non-perturbative effects
  - LundNet-3 shows much higher resilience than LundNet-5







Performance: $\dfrac{\epsilon_W}{\sqrt{\epsilon_{\mathrm{QCD}}}}$

Resilience:

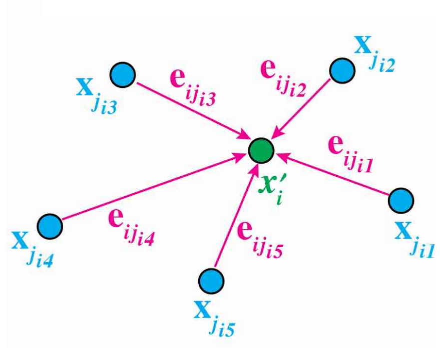
$$\zeta_{\mathrm{NP}} = \left(\frac{\Delta\epsilon_W^2}{\langle\epsilon\rangle_W^2} + \frac{\Delta\epsilon_{\mathrm{QCD}}^2}{\langle\epsilon\rangle_{\mathrm{QCD}}^2} \right)^{-1/2}$$

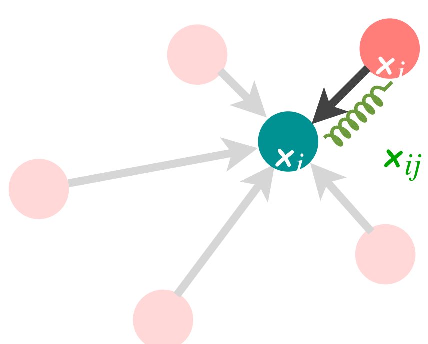
where $\Delta\epsilon = \epsilon - \epsilon'$ and $\langle\epsilon\rangle = 1/2\,(\epsilon + \epsilon')$

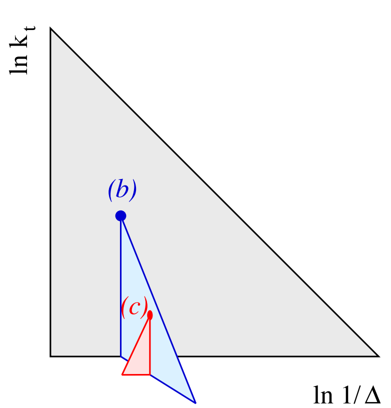
ϵ : *hadron-level*

ϵ' : *parton-level*

SUMMARY







- New ML-based approaches, especially Graph Neural Networks, significantly improve the performance of jet tagging
  - allow direct use of high-dimensional low-level inputs
  - simultaneously exploit substructure and flavor information
- Performance gains confirmed in real data
  - and translate to real gains physics analyses
- Promising prospects for future HEP experiments
  - method applicable to a broad range of applications:
    - jet tagging, full event discrimination, end-to-end reconstruction, …
  - exploiting underlying symmetry and physics principles proves key to successful ML applications in HEP
    - i.e., geometric deep learning
  - deeper understanding and better control of systematics remains an important topic for the future