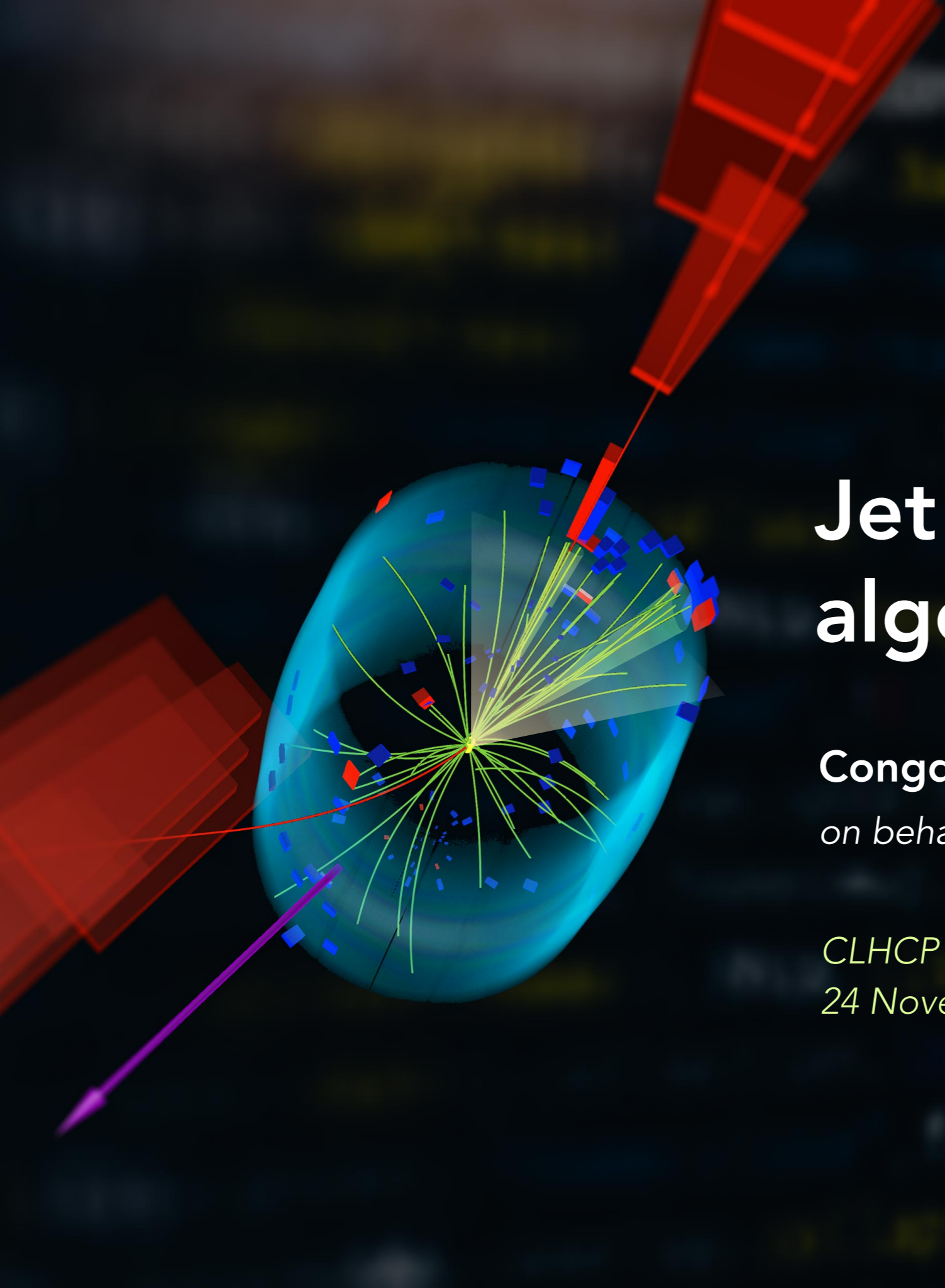




# Jet tagging algorithms in CMS

**Congqiao Li** (*Peking University*)  
*on behalf of the CMS Collaboration*

*CLHCP 2022 · Nanjing*  
*24 November, 2022*



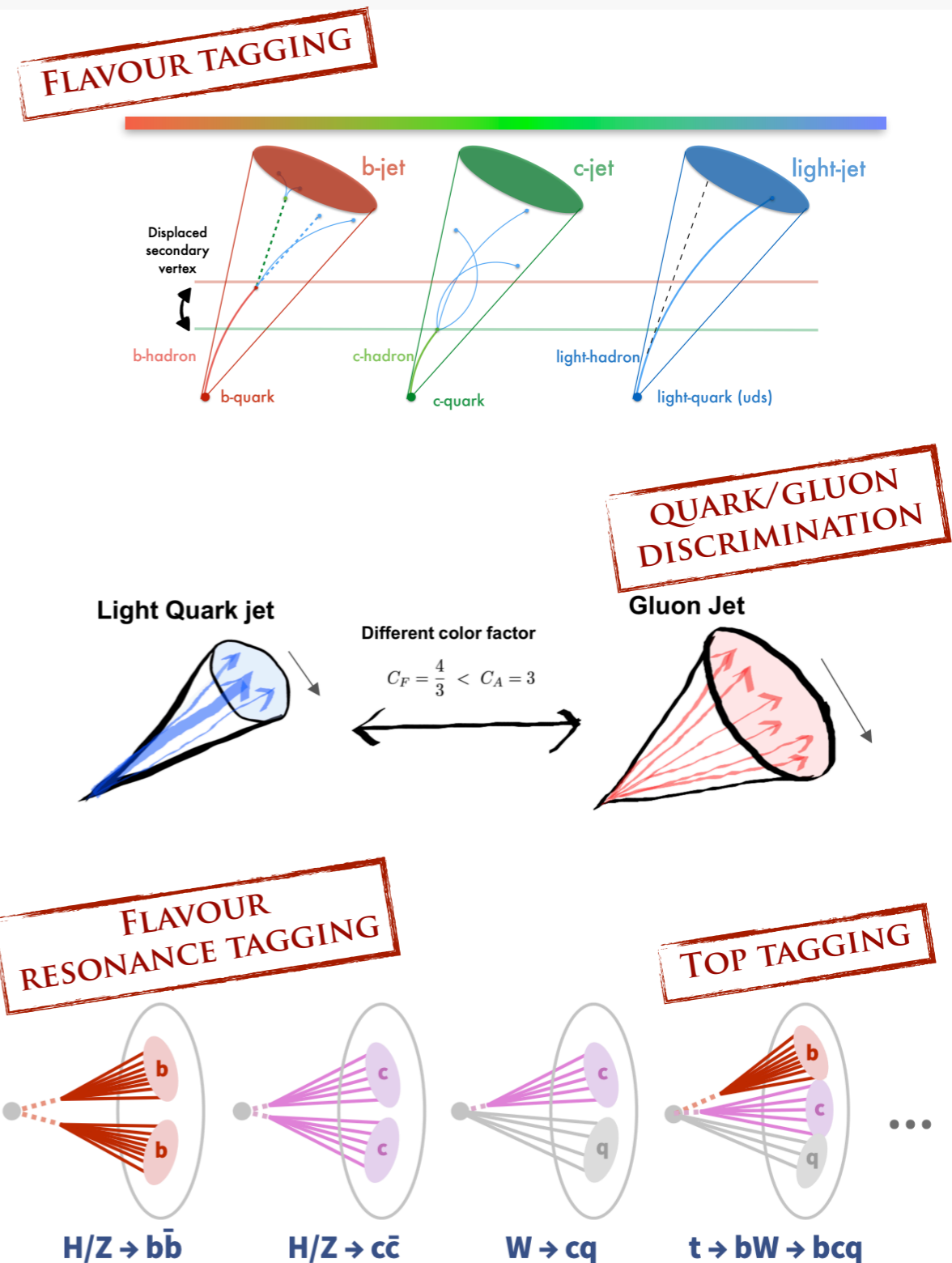
# Jet tagging × deep learning

→ Jet tagging in the deep learning era

- ❖ identifying the origin of a jet is a crucial topic for LHC experiments
- ❖ deep learning has taken jet tagging to a new performance level
  - ▶ already see a profound impact on a variety of analyses

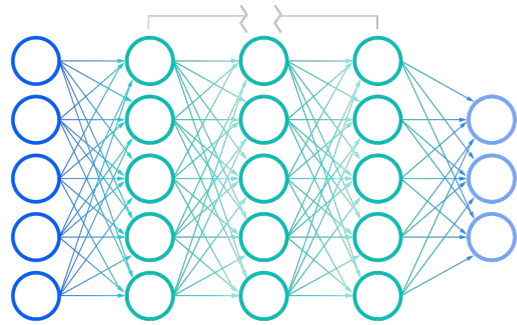
→ Aim of this talk

- ❖ introduce CMS jet tagging algorithms in the deep learning era
- ❖ share insights on developments of future taggers



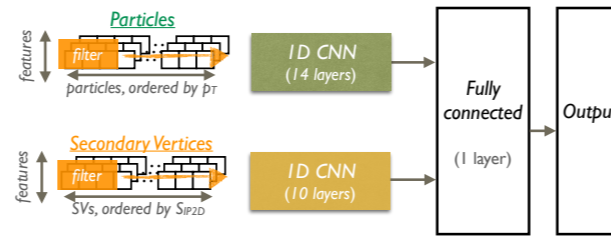
# Evolution of jet tagging algorithms

feed-forward NN (high-level inputs) ... 1D/2D CNN, RNN (low-level inputs) ... graph NN (low-level inputs)



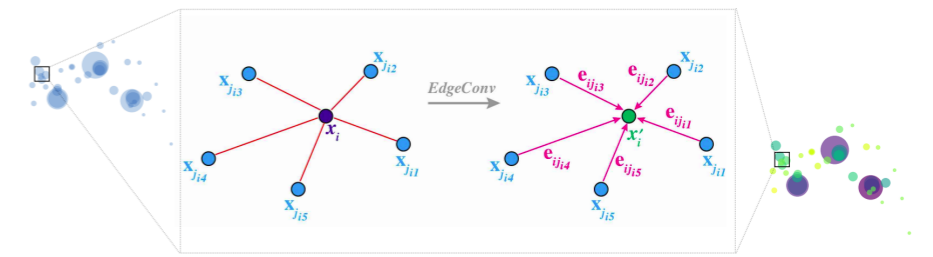
## Shallow networks

- Using high-level features as input to a BDT/neural network
  - e.g. using N-subjettiness variables



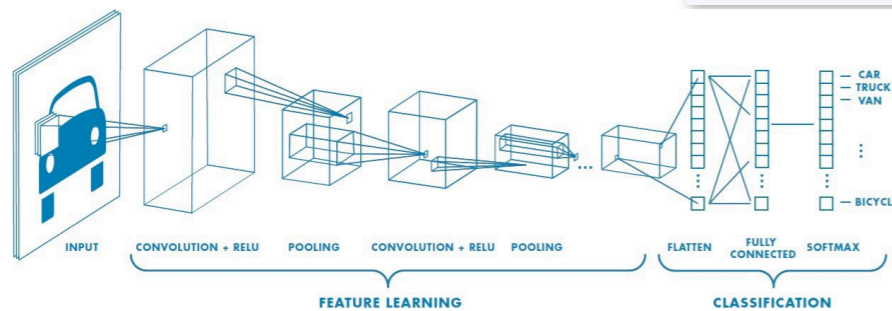
## Deep NN with low-level inputs

- Using particle-level features
- Input data structure determines the type of networks
  - jet as a *image* (fixed-grid data structure)  $\rightarrow$  2D CNN [*ImageTop*]
  - jet as a *sequence*  $\rightarrow$  1D CNN or RNN [*DeepJet, DeepDoubleX*]

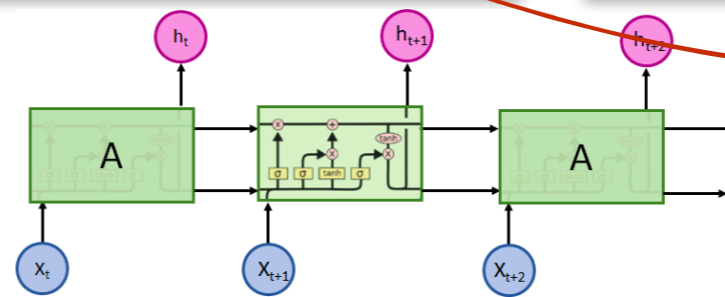


## More advanced structure

- Graph neural networks
  - proceed on a permutational-invariant set of particles (a.k.a. point cloud)
  - able to build "edges" between particles
  - a successful application: [*ParticleNet*]
- Beyond ParticleNet?



Typical CNN



Typical RNN



Typical graph

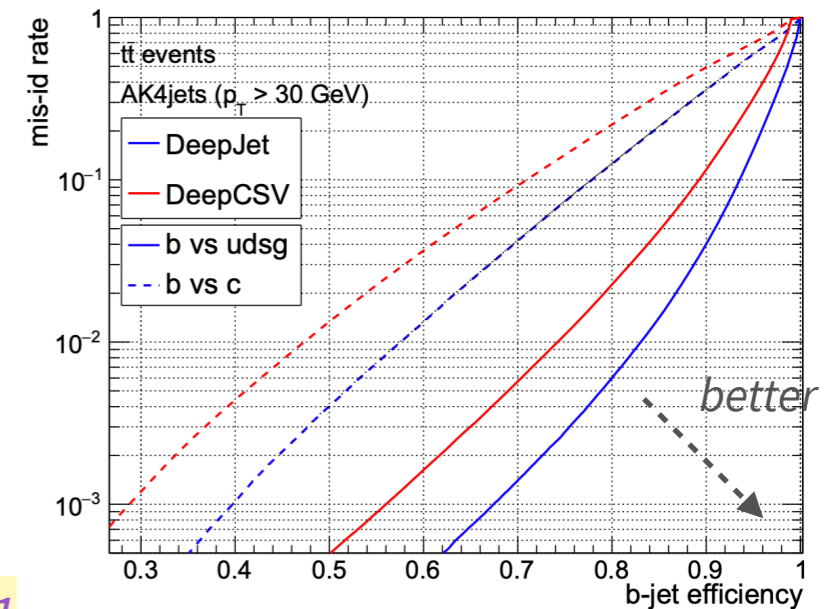
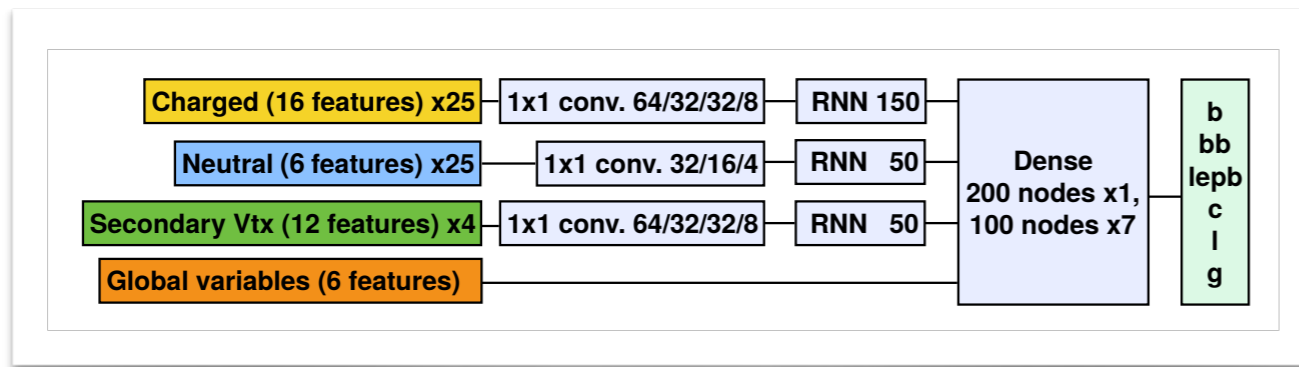
## Application in CMS

# Jet as sequence: DeepJet and DeepDoubleX

→ **DeepJet**: (b/c-tagging for small- $R$  jets)

[JINST 15 \(2020\) P12012](#)

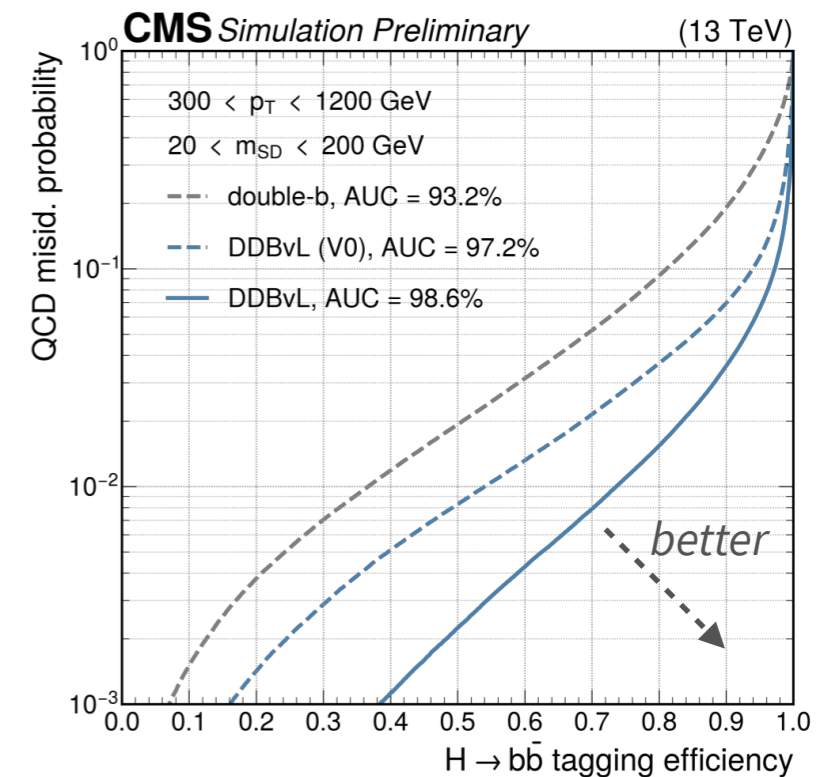
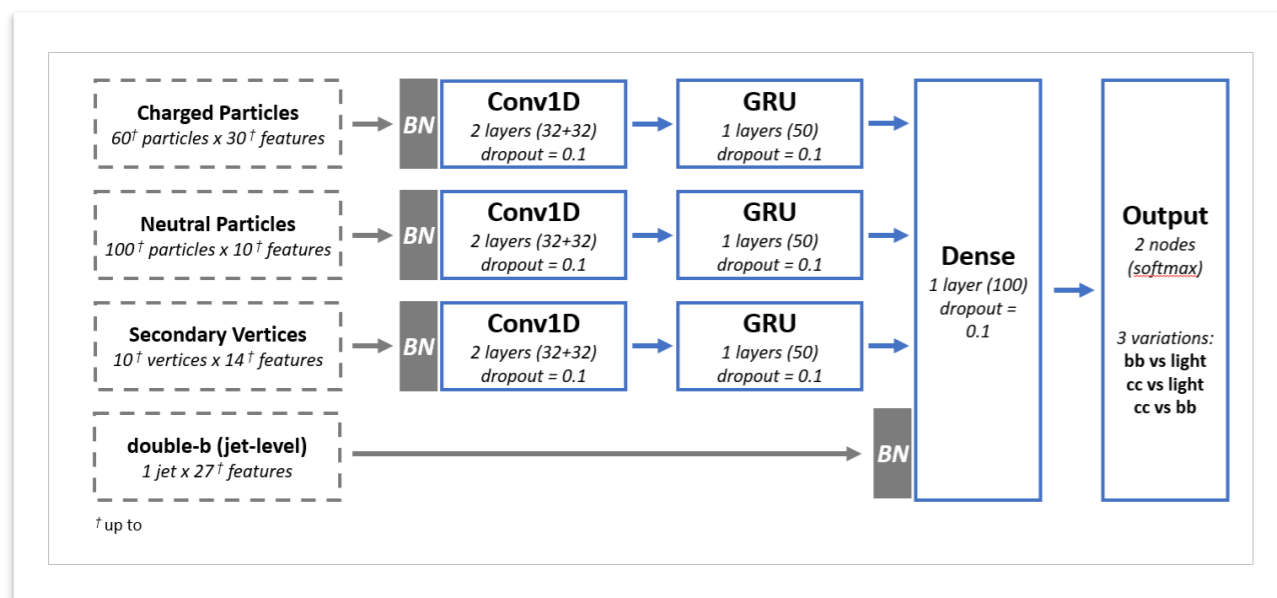
- ❖ particle-flow (PF) candidates, secondary vertices (organized as sequences) and high-level jet features 1D CNN+RNN output 6 scores



→ **DeepDoubleX**: ( $X \rightarrow bb/cc$  tagging for large- $R$  jets)

[CMS-DP-2022-041](#)

- ❖ PF candidates, SVs (organized as sequences) and jet-level inputs 1D CNN+GRU two scores in 3 schemes (BvsL, CvsL, CvsB)



# Path to ParticleNet

→ Representation of low-level input: from “sequence” to “graph”

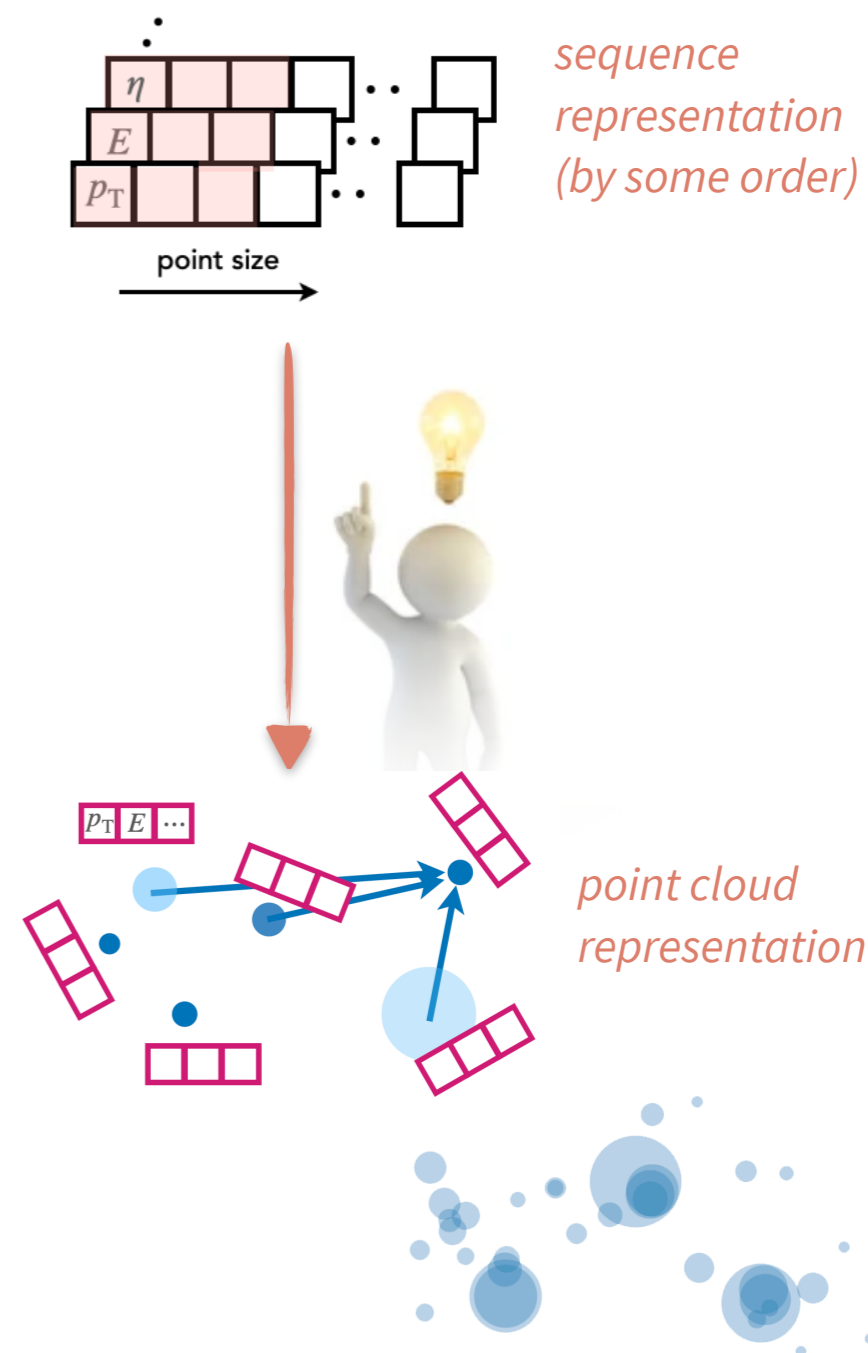
- ❖ **DeepJet** @ AK4, **DeepAK8/DeepDoubleX** for AK8  
**e.g. DeepAK8**: organize “PF candidates” and “secondary vertices (SV)” as two sequences ▶ input to two 1D CNNs ▶ concatenate, pass to dense layer, output multiple (17) scores (**multi-class classification**)

[JINST 15 \(2020\) P06005](#)



- ❖ **ParticleNet** (**current state-of-the-art in CMS**)  
 represent PF candidates and SVs in a **point cloud**  
 ▶ use GNN architecture, apply edge convolutions to **exploit geometric features** ▶ output multiple scores

[CMS-DP-2020-002](#)

- ✓ *permutational invariant: more effective representation of input data*
- ✓ *enable message passing to neighbouring nodes*



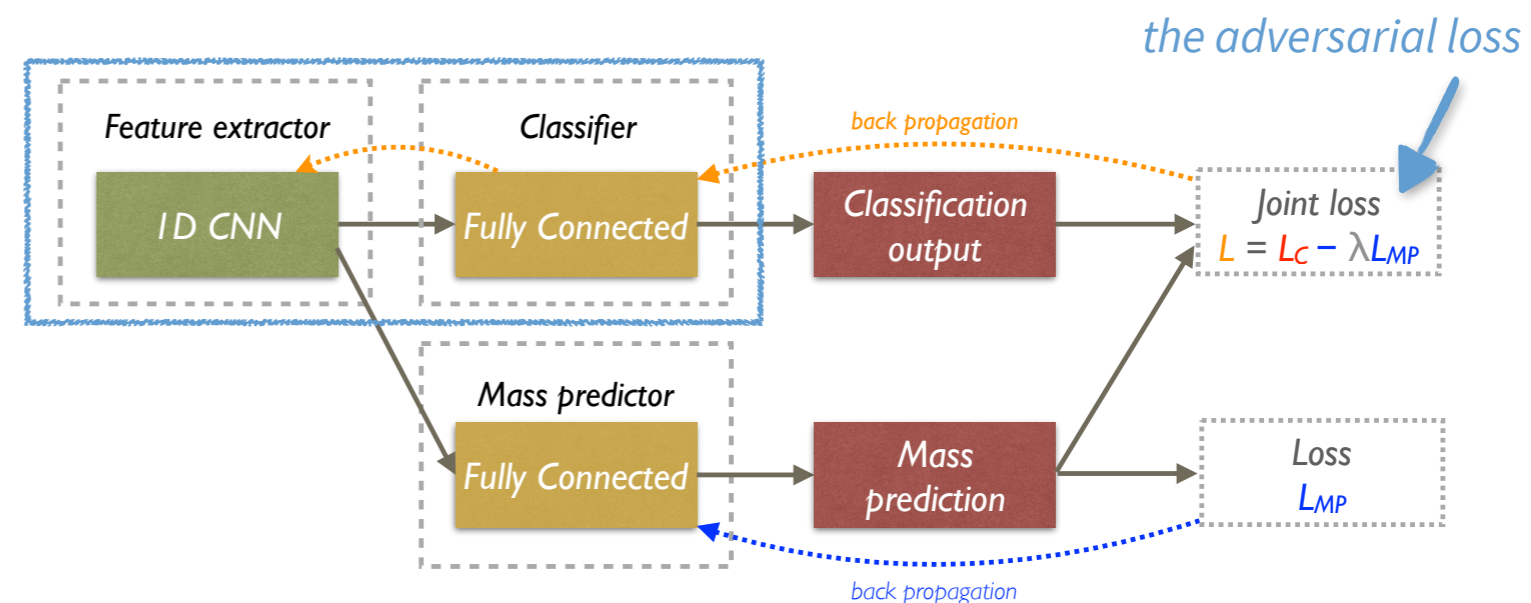
# Special technique: mass decorrelation (I)

- Mass decorrelation is a crucial technique in the design of tagger algorithms
  - ❖ a DNN would naturally learn from the jet kinematics, especially the jet mass, hence “sculpt a peak structure” in the background mass spectrum
- **By manual decorrelation**: spirit is to adopt different tagger working points for different bins
  - ❖ **designed decorrelated tagger (DDT)**: define jet bins on  $(\rho = \ln(m_{SD}^2/p_T^2), p_T)$   manual bin-dependent working point at BKG eff = 5%  define new tagger
 
$$N_2^{DDT}(\rho, p_T) = N_2(\rho, p_T) - N_2^{5\%}(\rho, p_T)$$

[JINST 15 \(2020\) P06005](#)



## → **By adversarial training**

- ❖ decorrelate the DNN score with mass by **adding an additional adversarial network** which contributes an adversarial loss (DeepAK8-MD)

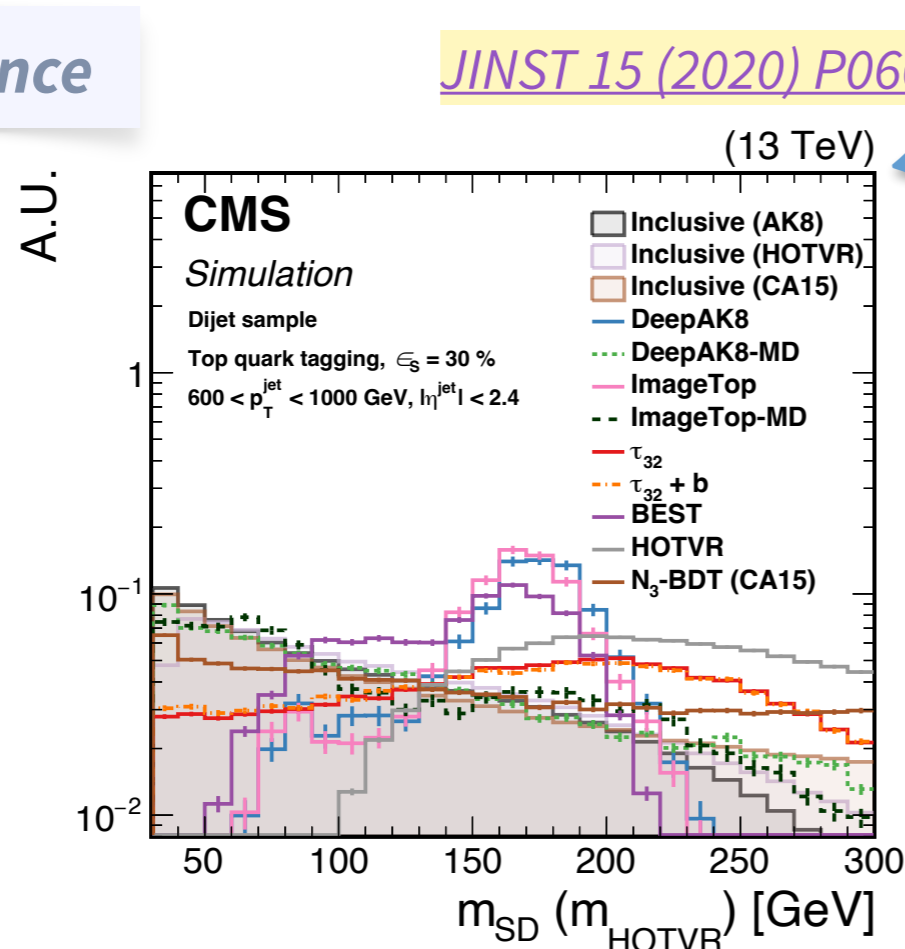


# Special technique: mass decorrelation (II)

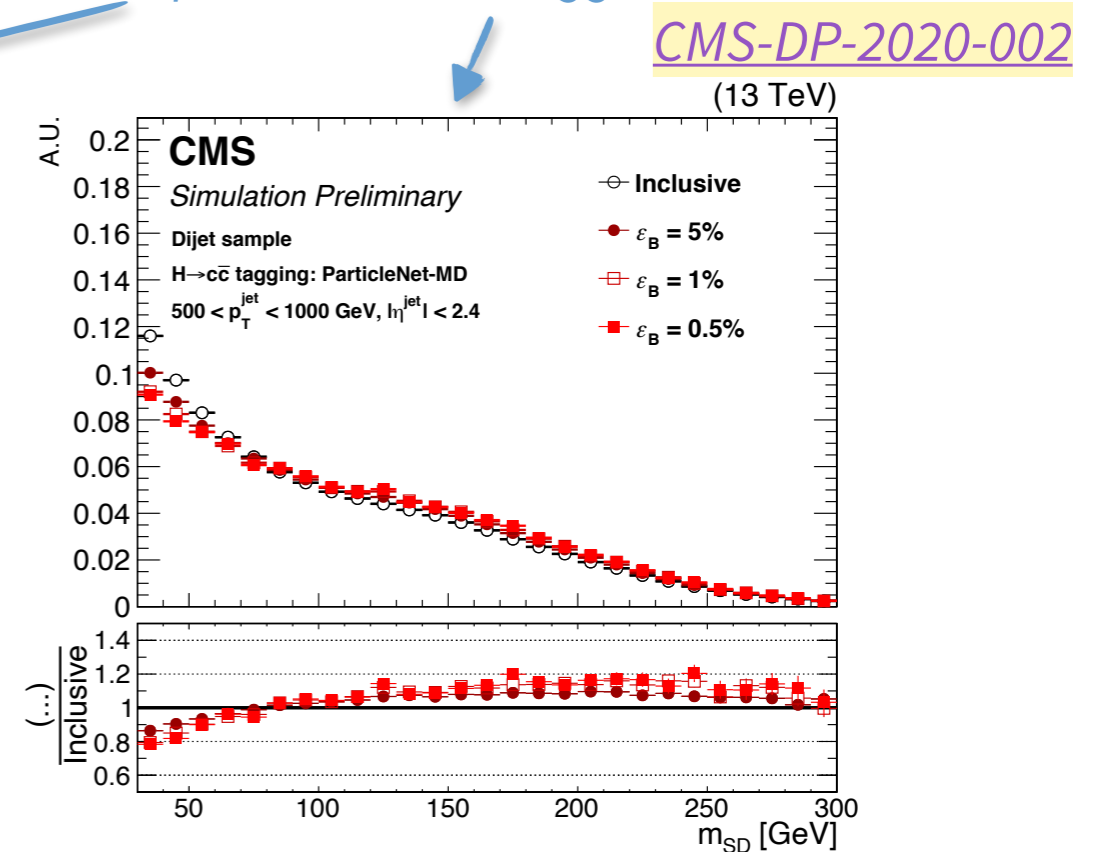
## → *By training with flat-mass sample*

- ❖ mass decorrelation approach for **ParticleNet-MD**:  
construct  $X \rightarrow bb/cc/qq$  sample for training:  $X = \text{spin-0 scalar with variable-mass}$  
- dedicated reweighting on  $(p_T, m_{SD})$  from signal to QCD jets  training performed on same ParticleNet model
- ▶ **fewer performance loss w.r.t. adversarial training approach**

### Performance



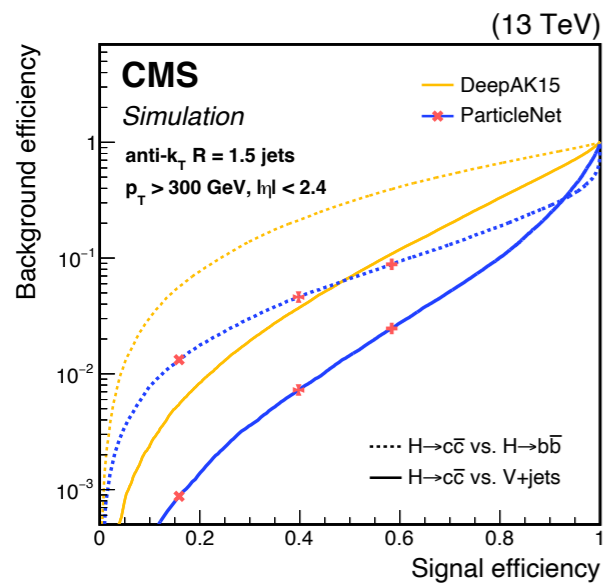
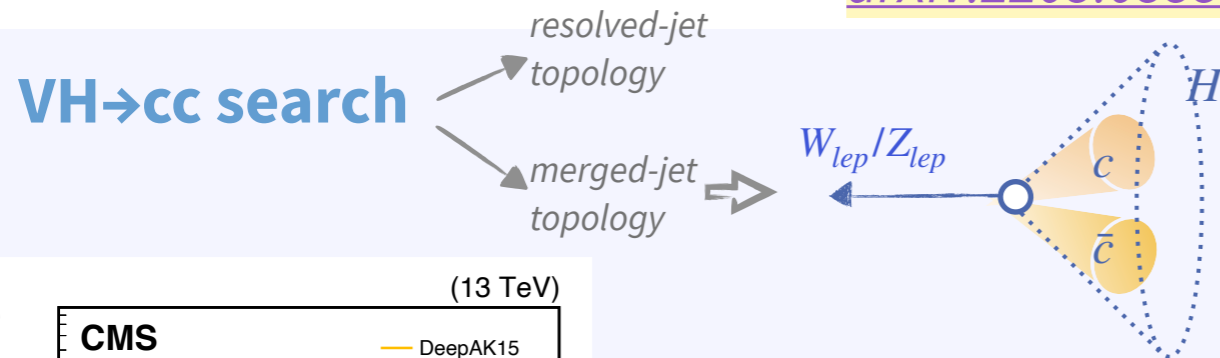
*flat mass shape after the MD tagger selection*



# Applications

→ Highlight recent CMS analyses that benefit from the ParticleNet tagger

[arXiv:2205.05550](https://arxiv.org/abs/2205.05550)

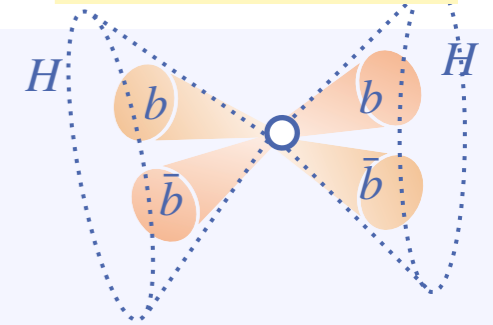


- Boosted  $H \rightarrow c\bar{c}$  jet jet tagged by **ParticleNet-MD** → **x5 improvement** in BKG (QCD & V+jets) rejection!

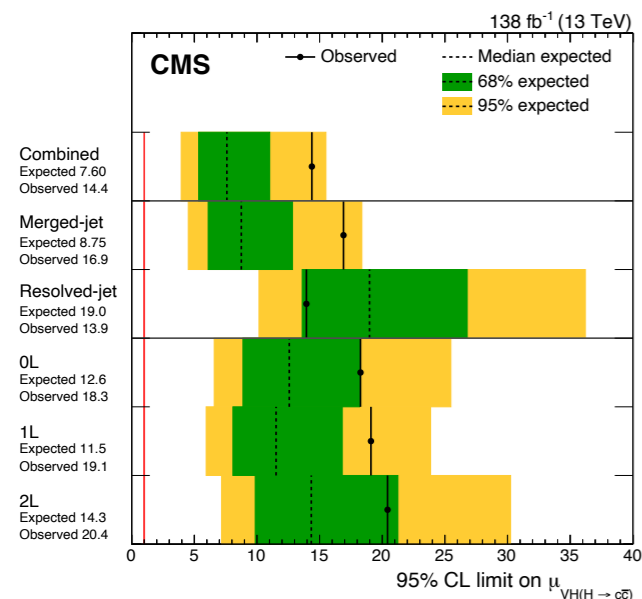
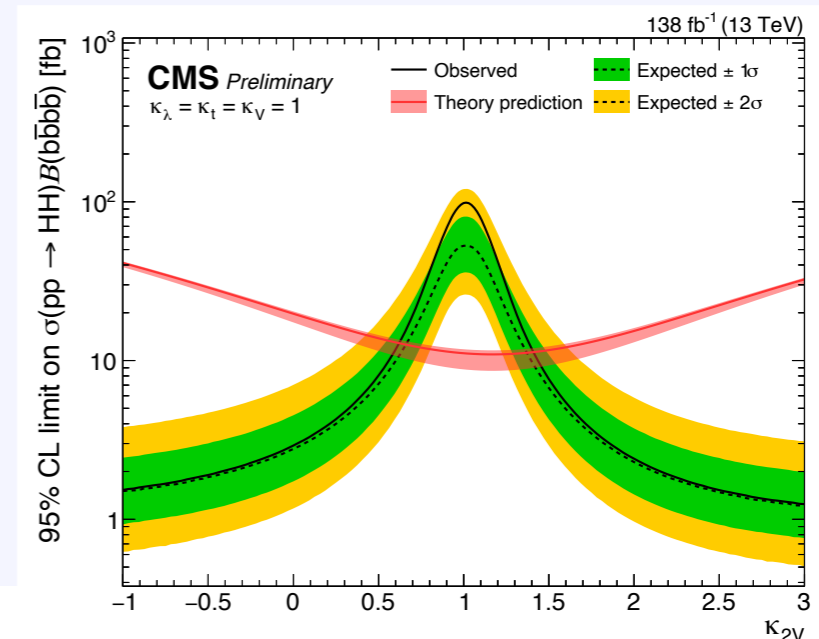
- Fit on “jet mass” (merged topology) and an event BDT variable (resolved topology)
- Most stringent limit on H-c coupling to date:  $1.1 < |\kappa_c| < 5.5$

[arXiv:2205.06667](https://arxiv.org/abs/2205.06667)

## Boosted HH→4b search



- $H \rightarrow b\bar{b}$  jet jet tagged by **ParticleNet-MD** → **x2 improvement** in BKG rejection
- Regression on  $H \rightarrow b\bar{b}$  jet mass based on ParticleNet → **40% improvement** in resolution
- Most stringent limit on  $\kappa_{2V}$  to date:  $0.6 < \kappa_{2V} < 1.4$





# Beyond ParticleNet? (I)

CMS-DP-2022-050

→ Where to seek further improvement in future tagger design? 🤔

→ More advanced neural networks?

- ❖ borrow latest ML advances ; deepen our understanding of the input data
- ❖ example: Particle Transformer, first time applied to CMS in small- $R$  jet tagging



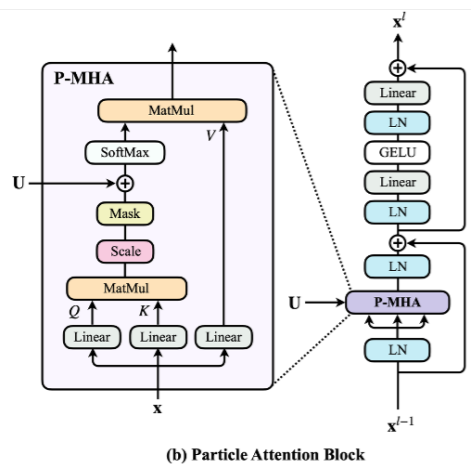
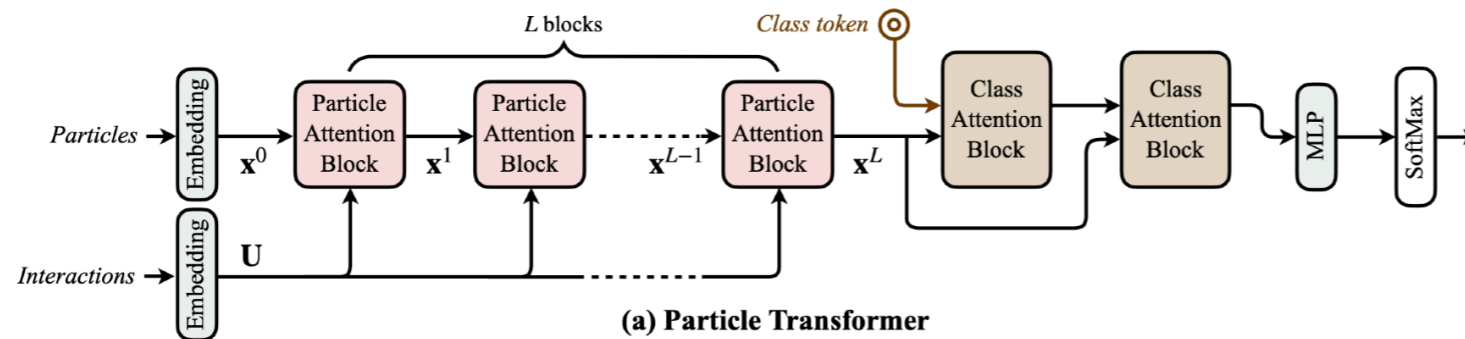
## Benefits come from:

- ✓ use advanced Transformer architecture  
→ equiv as more efficient fully-connected graph
- ✓ introduce pairwise mass features etc.  
→ introduce a sub-network invariant to Lorentz (sub)symmetries

### Particle Transformer:

ArXiv: 2202.03772

#### the transformer designed for particle physics



$$P\text{-MHA}(Q, K, V) = \text{SoftMax}(QK^T / \sqrt{d_k} + \mathbf{U})V$$

$d_k$ : dimension of  $K$

Choice of the pair-wise features: from LundNet

$$\Delta = \sqrt{(y_a - y_b)^2 + (\phi_a - \phi_b)^2}$$

$$k_T = \min(p_{T,a}, p_{T,b}) \cdot \Delta$$

$$z = \min(p_{T,a}, p_{T,b}) / (p_{T,a} + p_{T,b})$$

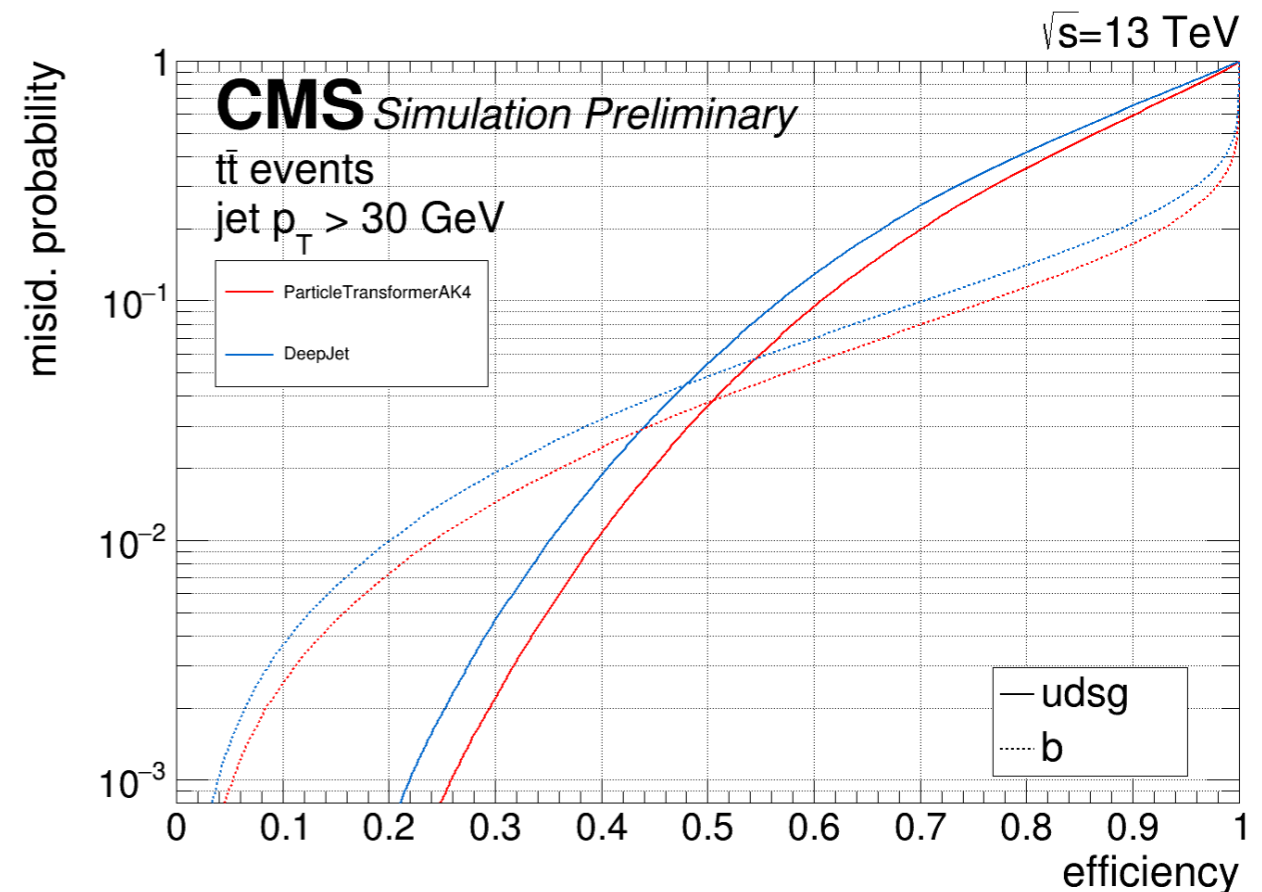
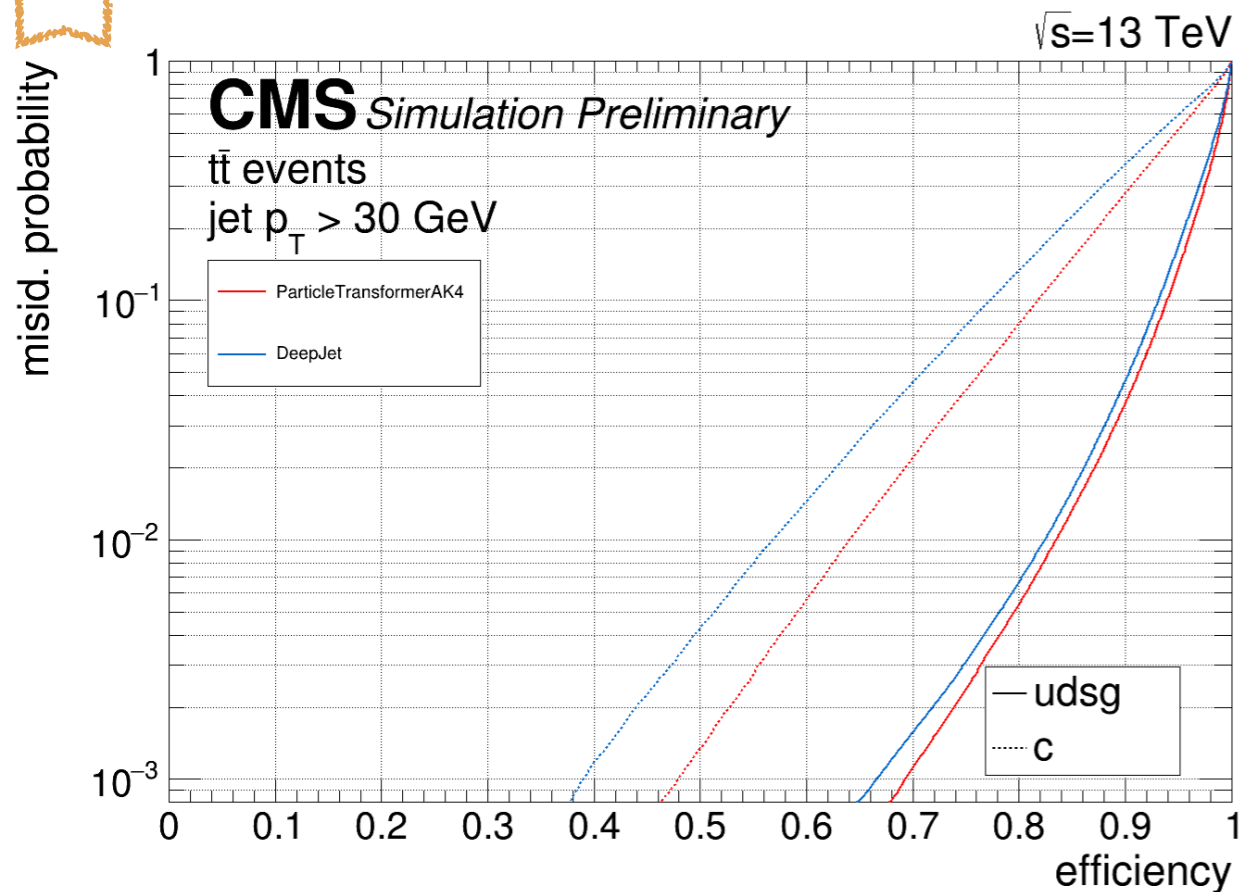
$$m^2 = (E_a + E_b)^2 - \|\mathbf{p}_a + \mathbf{p}_b\|^2$$

S. Qian @ML4Jets2022

# Beyond ParticleNet? (I)

[CMS-DP-2022-050](#)

- Where to seek further improvement in future tagger design? 🤔
- More advanced neural networks?
  - ❖ borrow latest ML advances ; deepen our understanding of the input data
  - ❖ example: Particle Transformer, first time applied to CMS in small- $R$  jet tagging



# Beyond ParticleNet? (II)

CMS-DP-2022-049

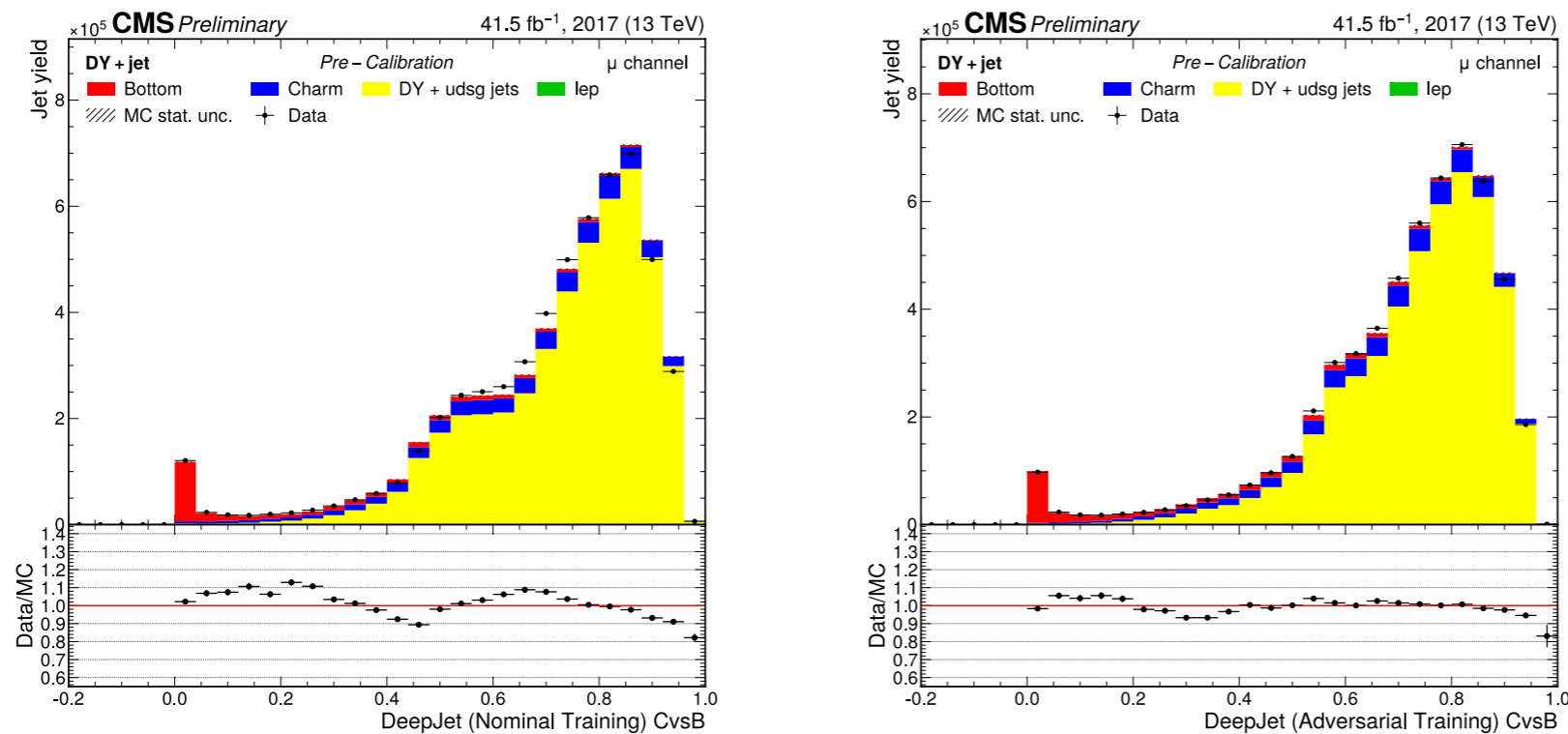
→ Better training strategy to mitigate data/MC discrepancy?

❖ e.g. consider adversarial training

- ▶ concept: to adopt an adversarial attack to the input data, and train the network alternatively in the adversarial route to “defend” such attack



## Comparing Data/MC agreement for nominal and adversarial training: light jet selection



→ Agreement improves! More examples in backup.

*Better agreement between data/MC*

# Summary & outlook

- **Recent advances in jet tagging algorithms start to impose a huge impact on CMS analyses**
  - ❖ deep learning advances → more powerful tagging algorithms
    - developing path: **single/few rule-based jet observables → “shallow ML” using jet inputs → directly using low-level input to train deep NN**
  - ❖ sensitivity in analyses greatly improved
    - improvement from  $N_x$  BKG rejection  $\approx$  collects  $N_x$  more data !
    - results in more precise SM measurements, more stringent limits; or **even accelerate finding of a new particle!**
- **Progress still ongoing...**
  - ❖ exploiting more advanced architectures, better input data engineering
  - ❖ strategy to achieve better data/MC similarity
  - ❖ a path towards a general tagger applicable to all phase-spaces...
- **Long but optimistic journey ahead!**

# Backup

---

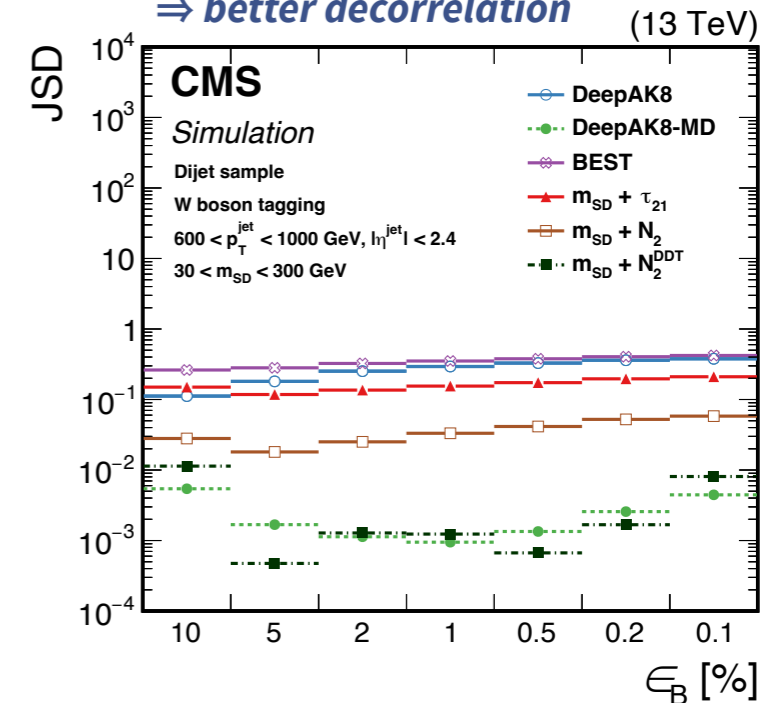
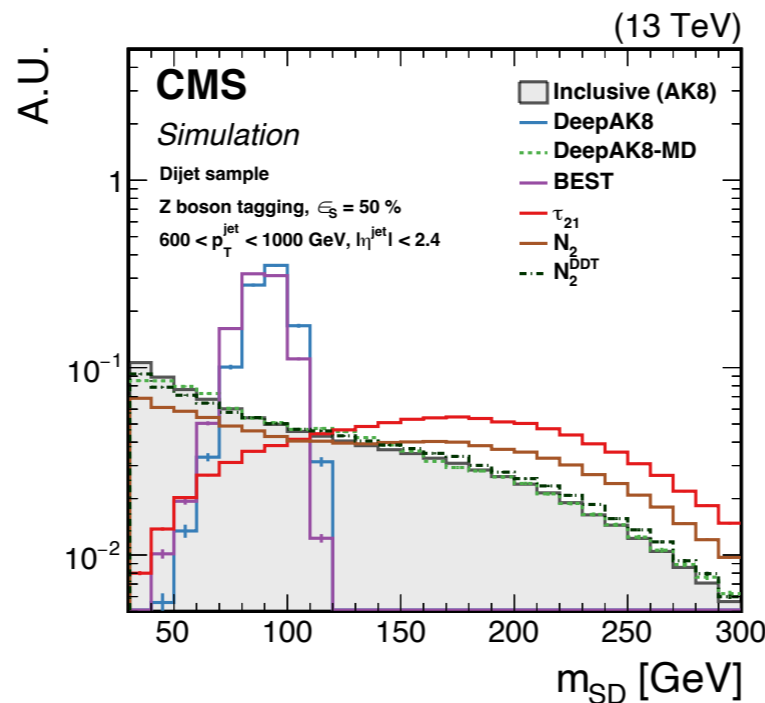
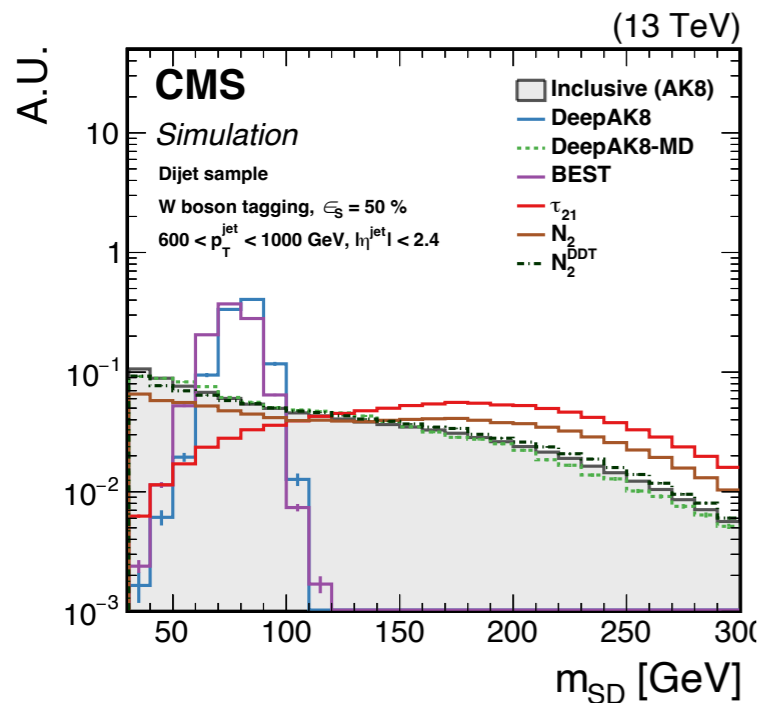
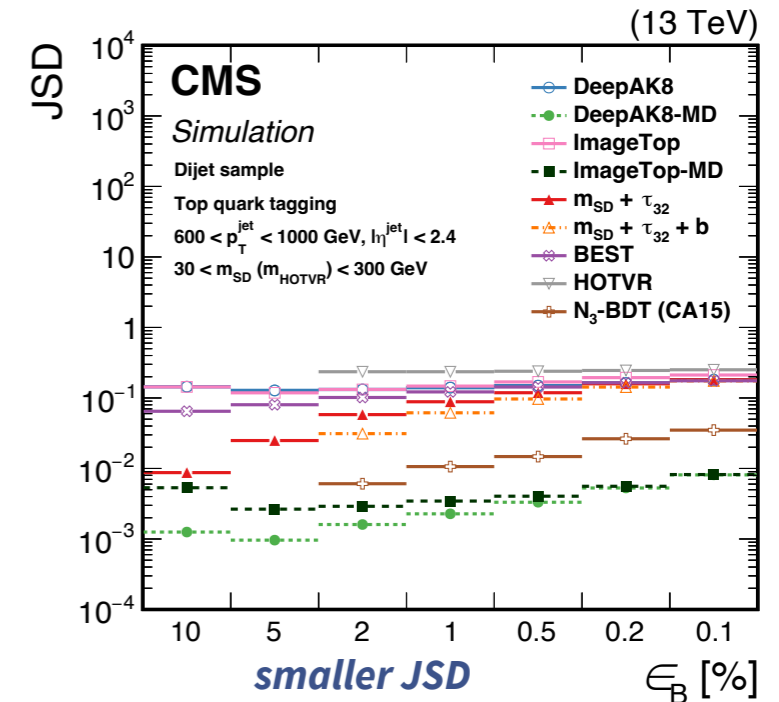
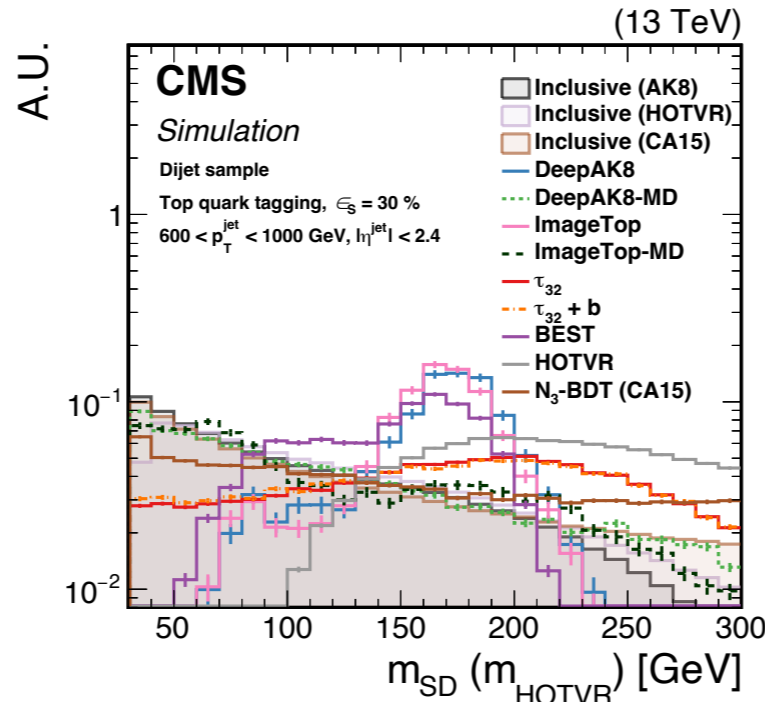
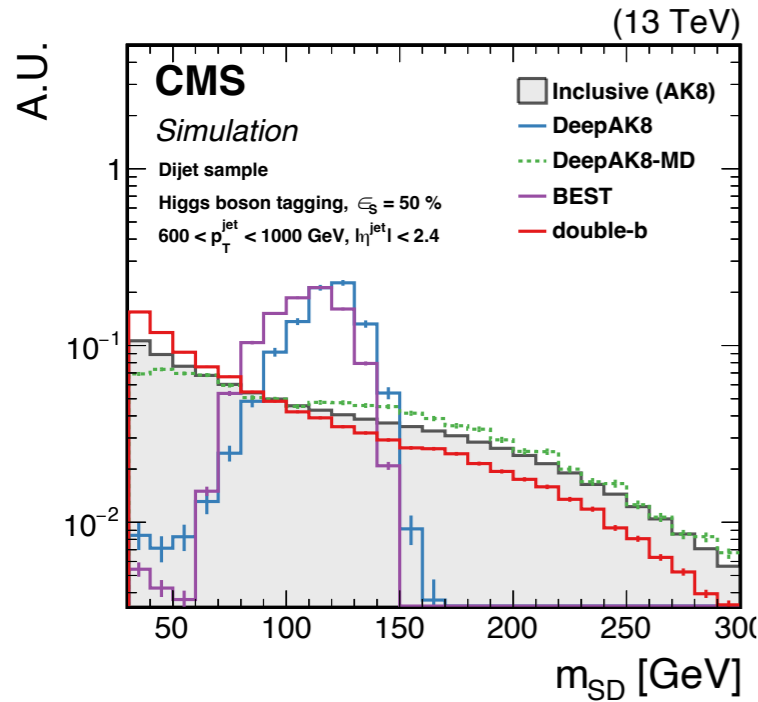
# Mass decorrelation plots

mass sculpting effect in various taggers

[JINST 15 (2020) P06005]

Jensen-Shannon divergence (JSD) as a function of BKG efficiency

[JINST 15 (2020) P06005]



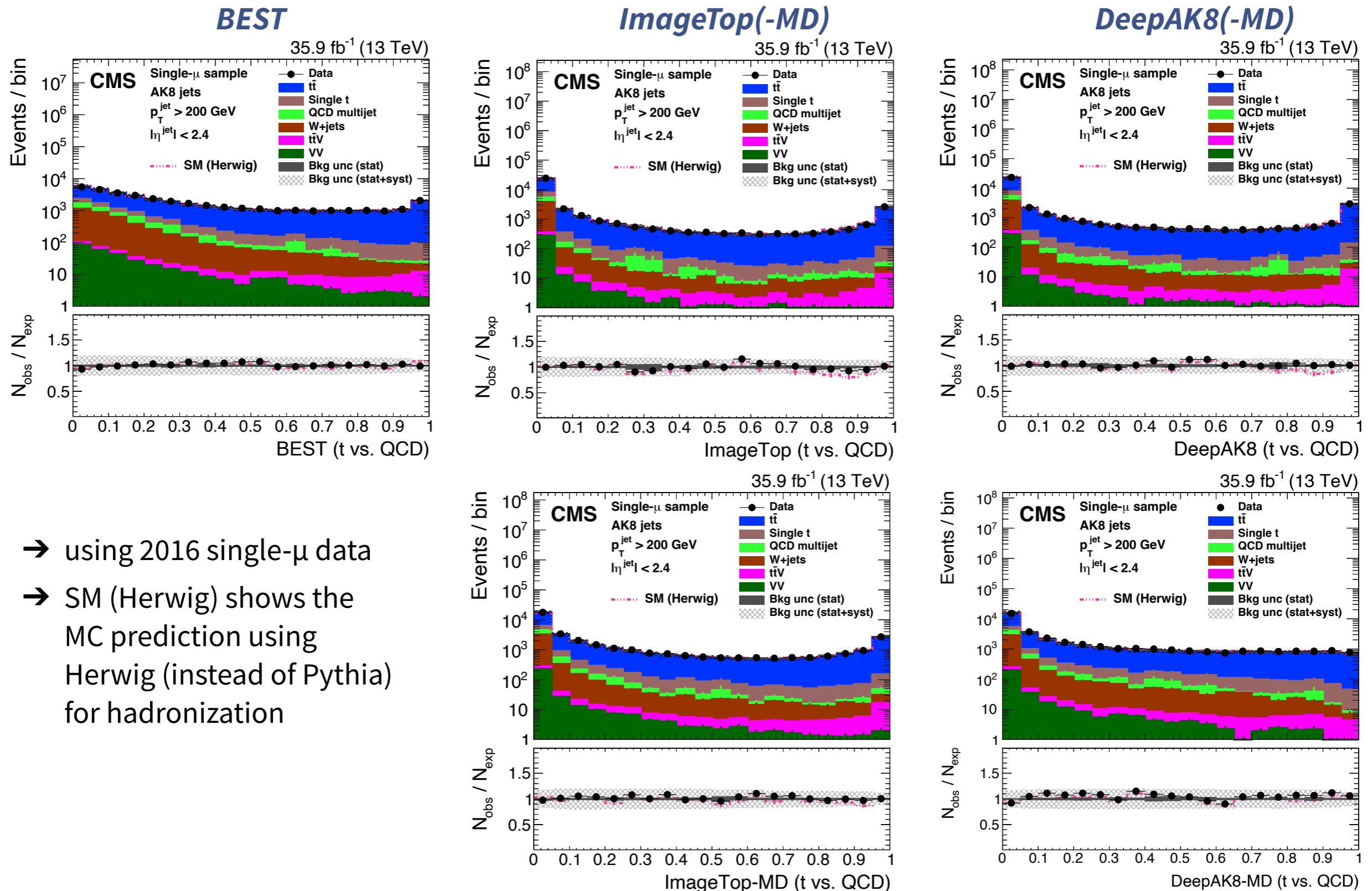
smaller JSD  
 => better decorrelation

$$\text{JSD}(P\|Q) = \frac{1}{2} \left( \text{KLD}(P\|M) + \text{KLD}(Q\|M) \right), \text{ where } M = \frac{P+Q}{2}$$

$$\text{KLD}(P\|Q) = \sum_i P(i) \log_{10} \frac{P(i)}{Q(i)}$$

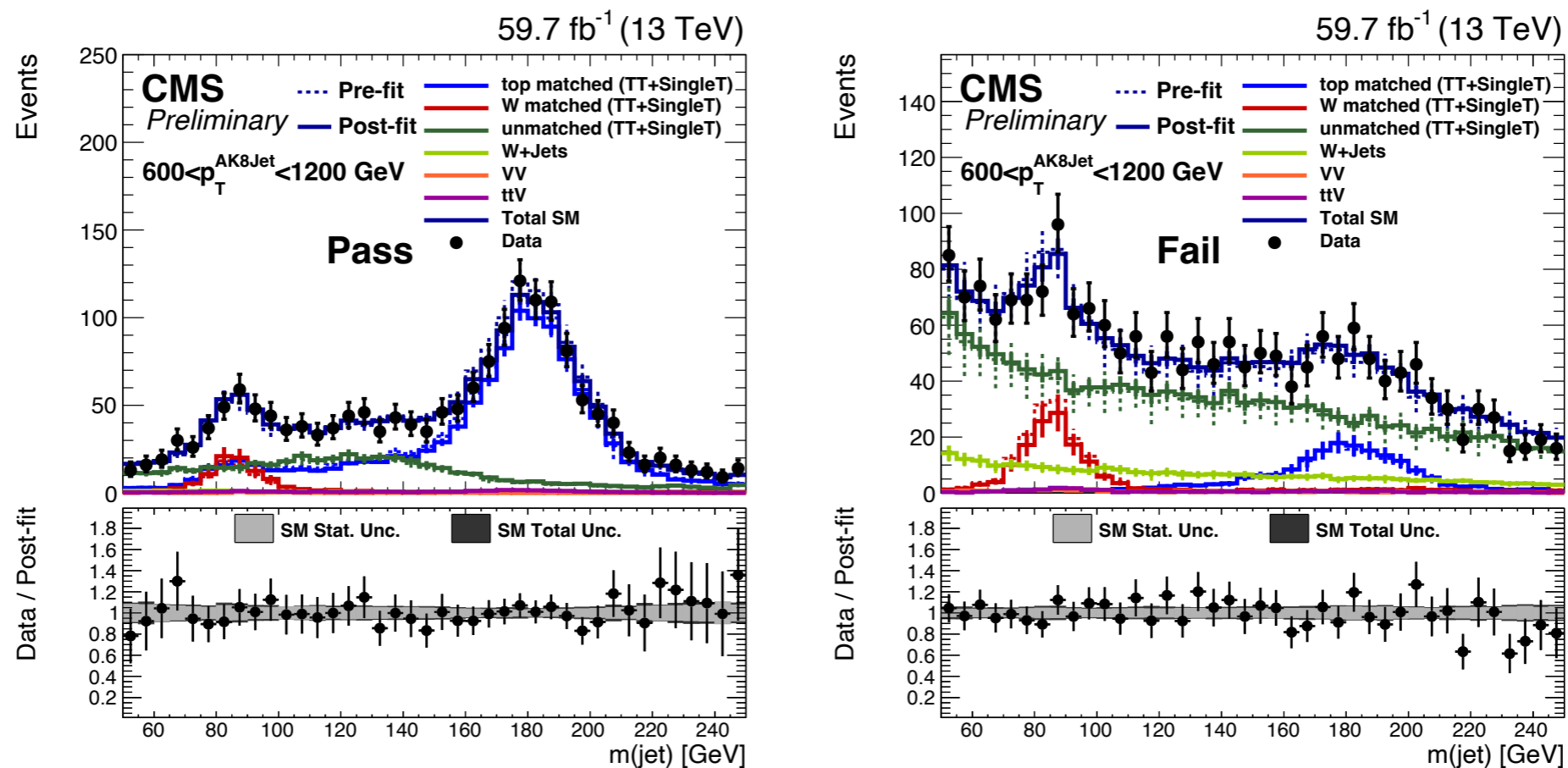
# Data/MC comparison

data/MC comparison on single- $\mu$  samples [[JINST 15 \(2020\) P06005](#)]



# Calibration of W/top taggers

CMS-DP-2020-025

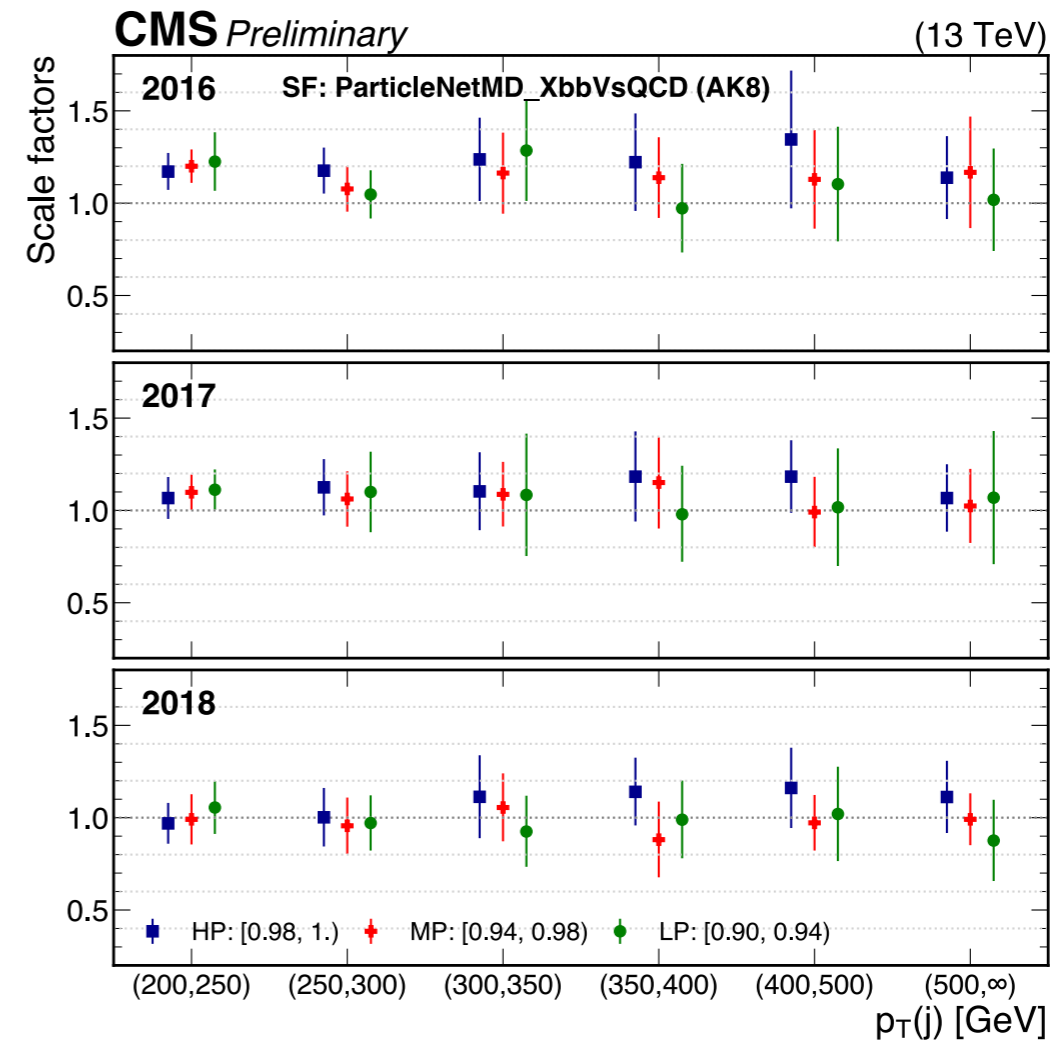
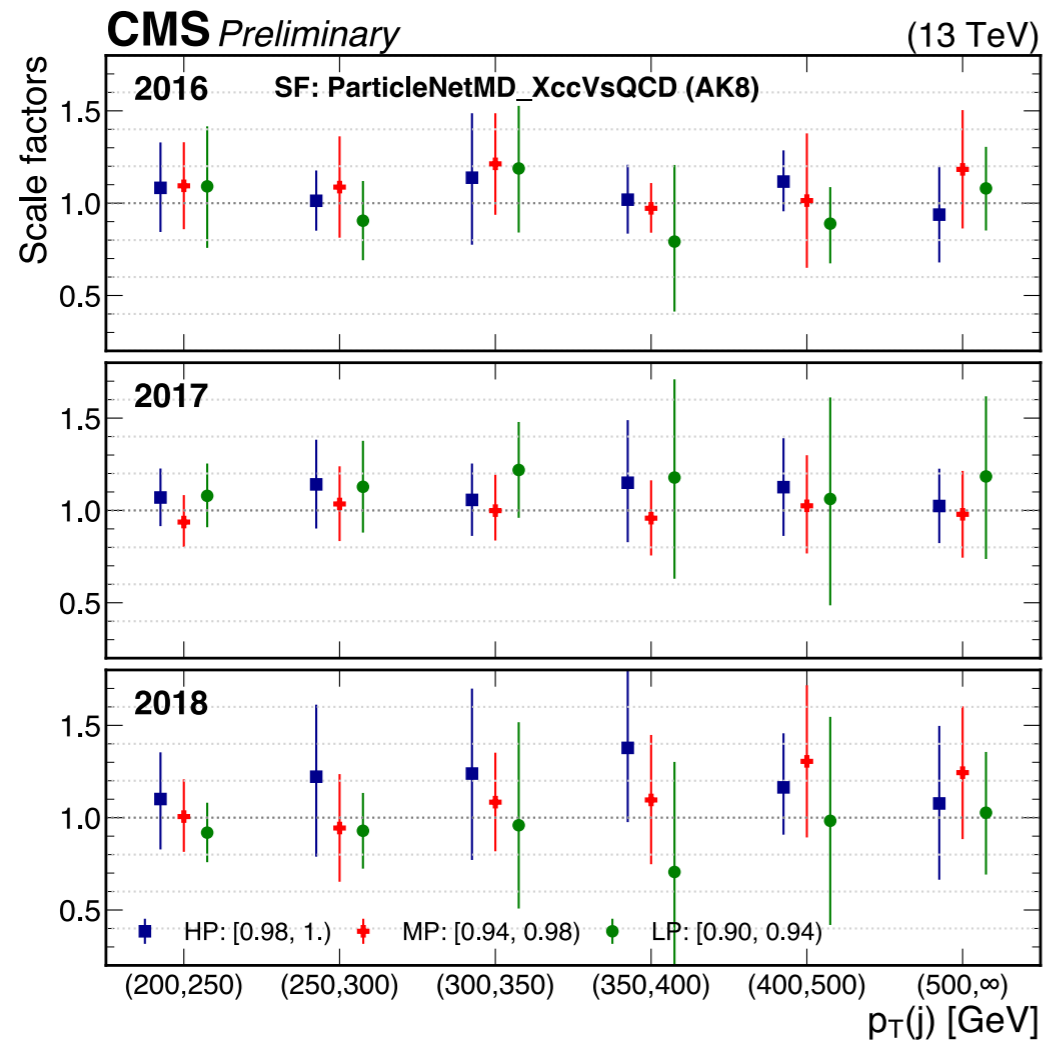


**Figure 12:** The  $m_{\text{jet}}$  distribution for data and simulation in the passing (left) and failing (right) categories for the mass decorrelation version of the top tagging (1% mis-identification rate) on the  $p_T$  window  $600 < p_T^{\text{AK8Jet}} < 1200$  GeV. The solid lines correspond to the contribution of each category after performing maximum likelihood fit. The contribution from QCD multijet events is included in the total SM. The dashed lines are the expectation from simulation before the fit. The lower panel shows the data-to-simulation ratio. The "top/W matched" convention used here indicate that a simulated top quark/W boson is overlapping with the large-radius jet, but not necessarily all of its decay products.



# Calibration of $X \rightarrow b\bar{b}/c\bar{c}$ taggers

[CMS-DP-2022-005](#)



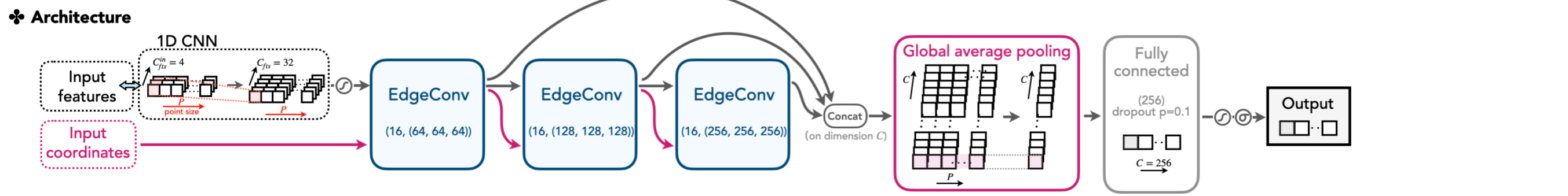
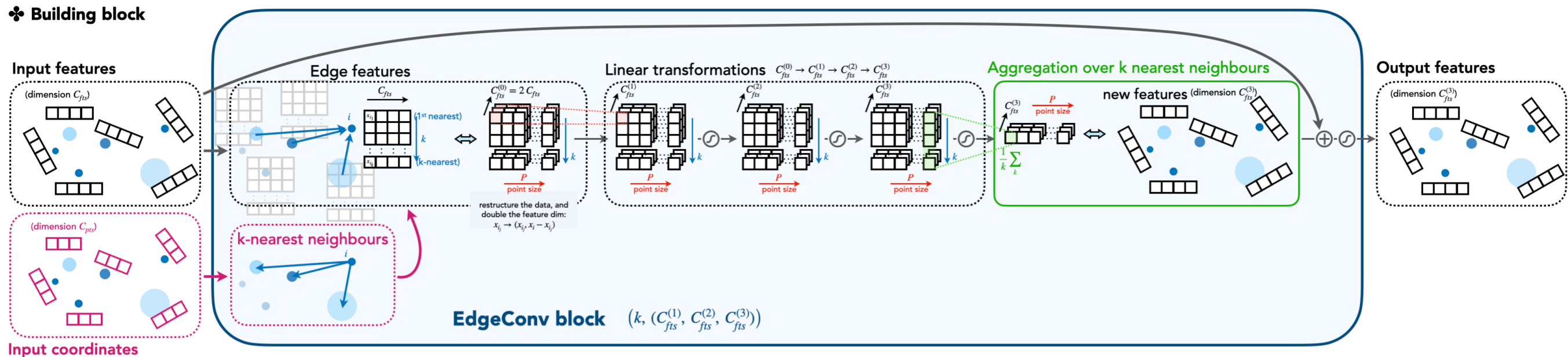
# Explaining ParticleNet

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]



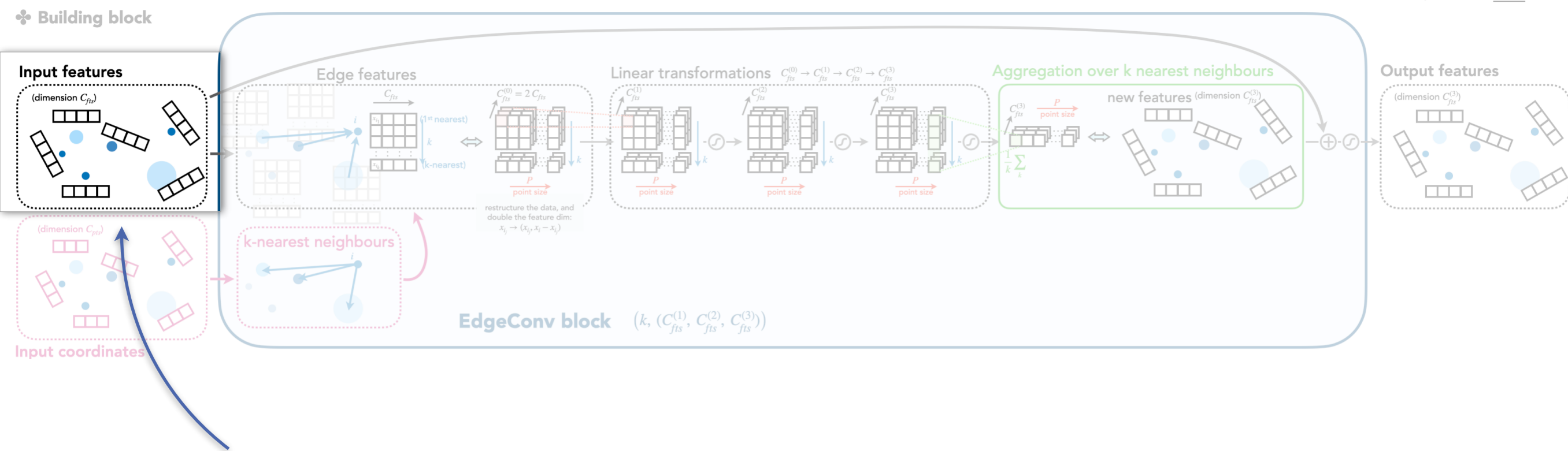
# Explaining ParticleNet

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

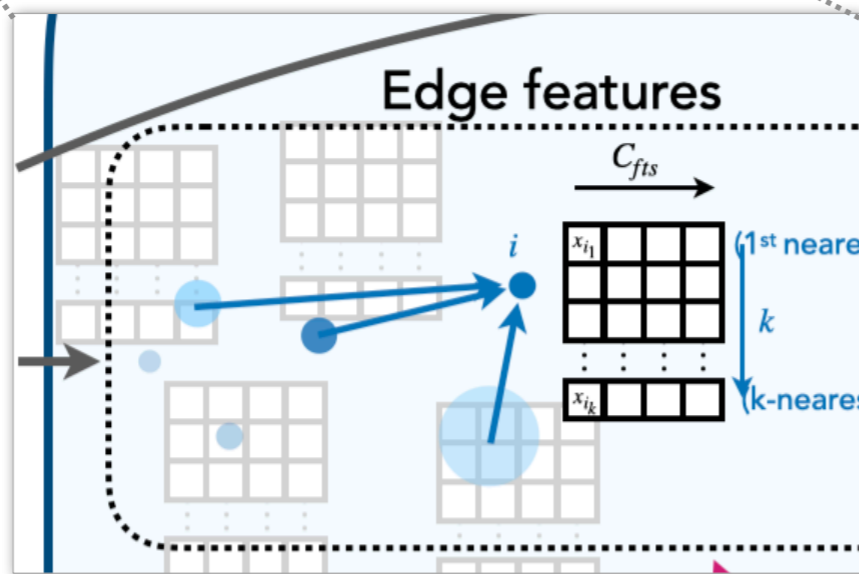
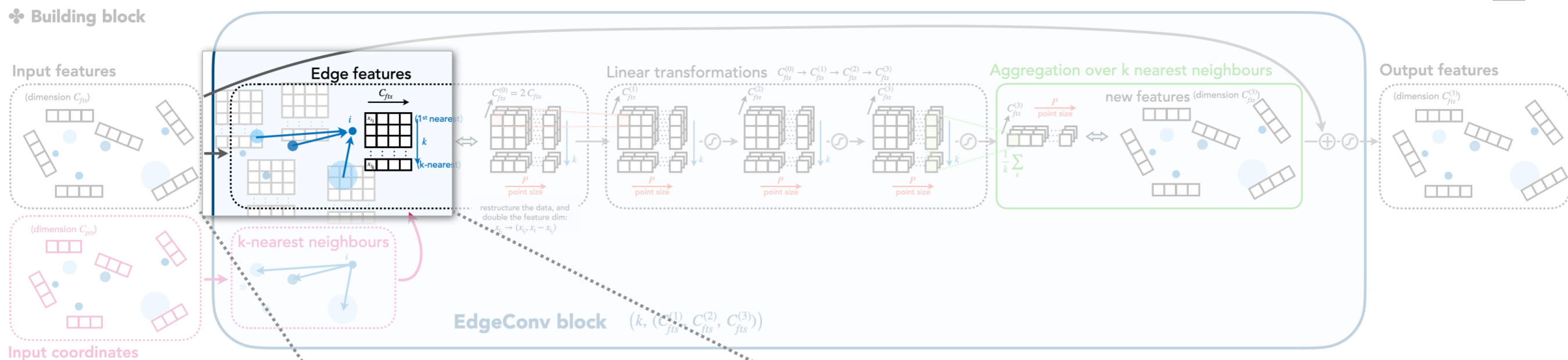
# Explaining ParticleNet

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

# Explaining ParticleNet

**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](#)]

