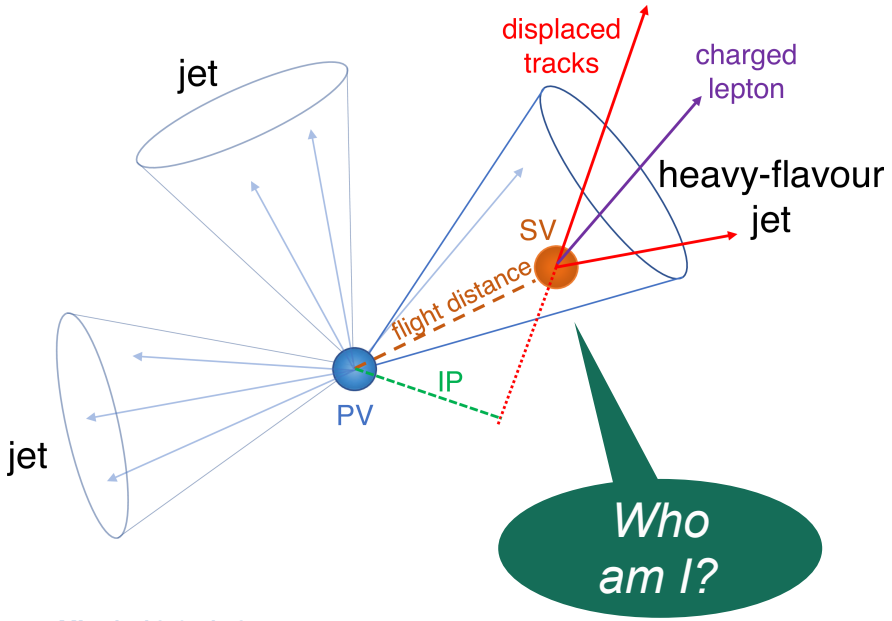
Flavour Tagging Short Exercise

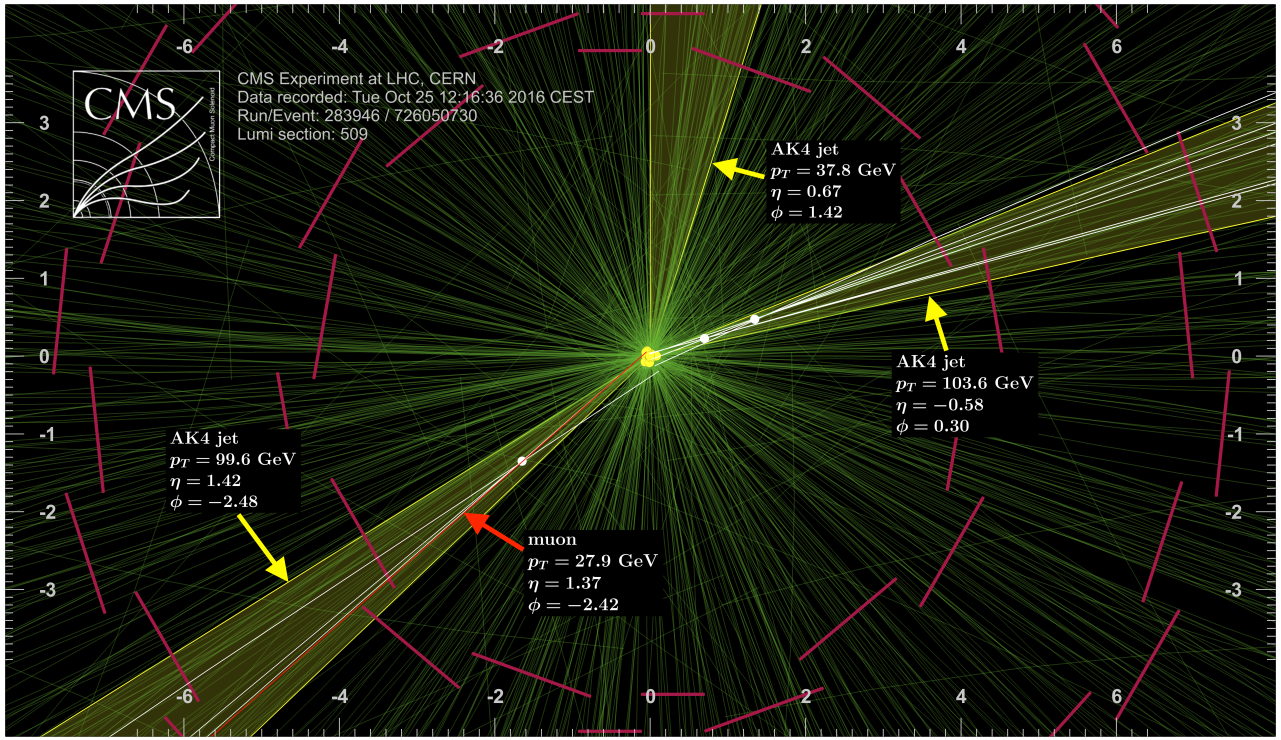
Annika Stein

CMSDAS@CERN 2023
06.06.2023

Jet tagging basics



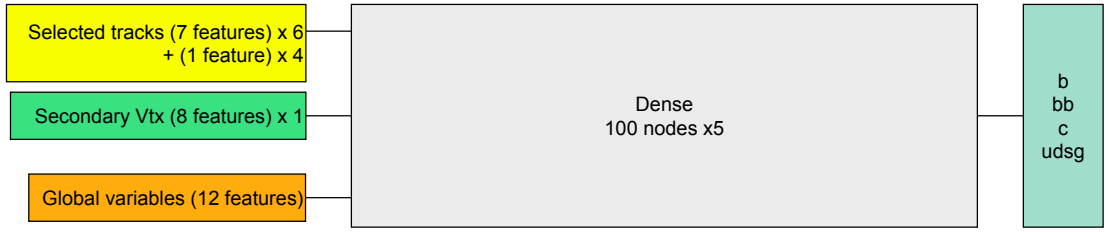
arXiv:1712.07158

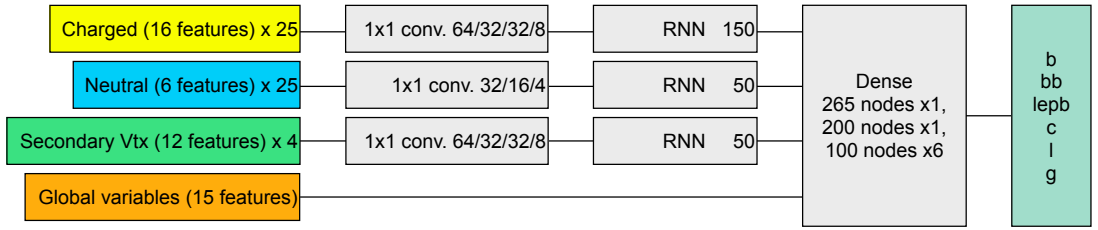


Heavy-flavour jets (b & c jets)

- Long lifetime of b/c hadrons → secondary vertex & displaced tracks
- Larger mass, harder fragmentation compared to light jets
- (Soft) charged lepton in 20% (10%) of the cases for b (c) jets

Jet tagging algorithms at CMS (Run 2)

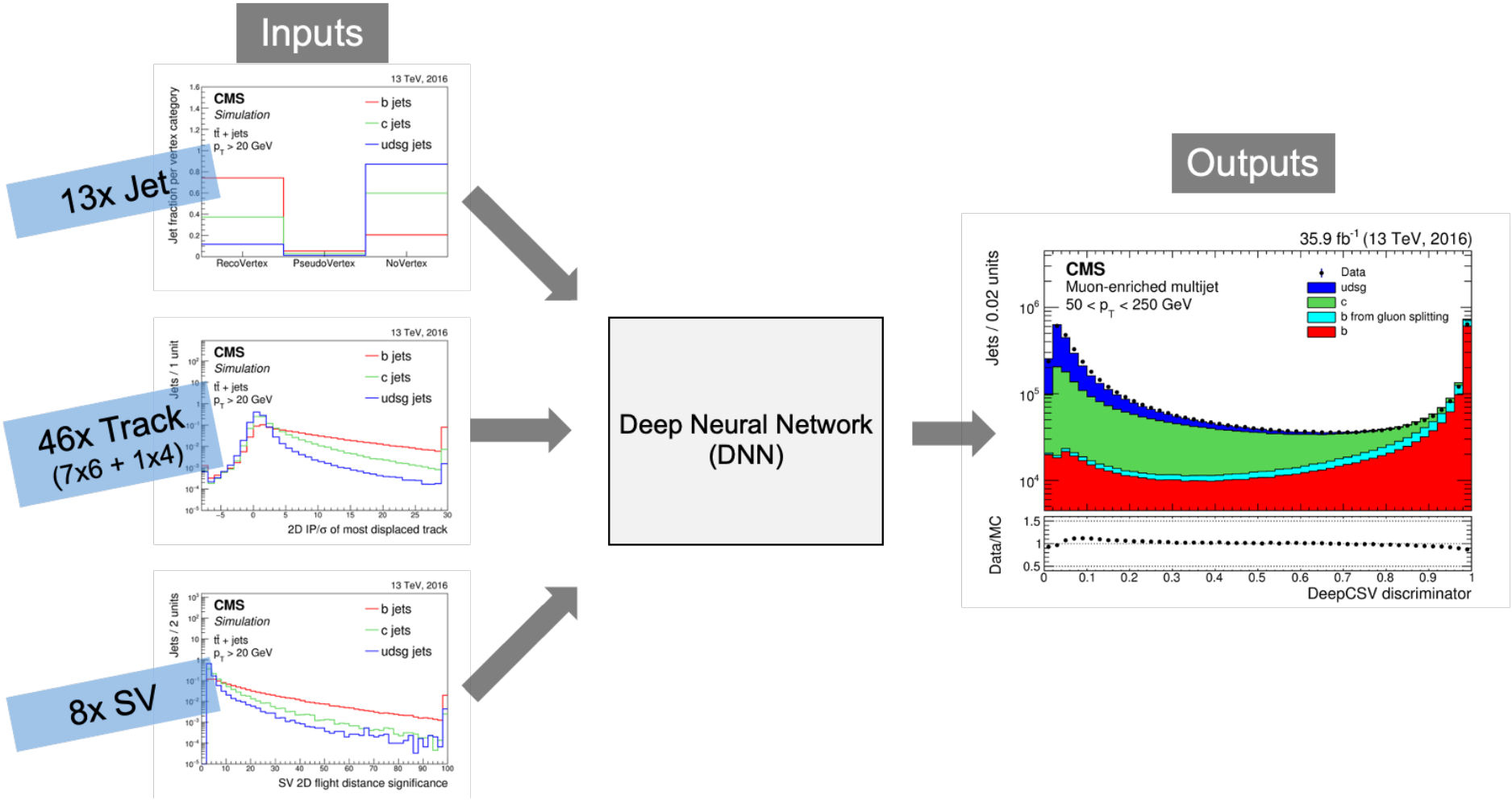




DeepCSV	DeepJet
Only fully-connected (**dense**) layers	**Convolutional** layers, recurrent layers (**LSTMs**), dense layers
66 features, from up to six tracks, one secondary vertex, and high-level jet features	**613** features, of which **many** are **low-level** features directly from up to 25 ParticleFlow candidates (charged & neutral) and four secondary vertices; high-level features from DeepCSV
Four outputs: b, bb, c, udsg	**Six outputs**: b, bb, lepb, c, uds, g

Typical workflow:
- train on **simulation**
- evaluate on simulation & **data**
- observe **differences** and correct by calibrating via scale factors

Inputs and how they can be used: example DeepCSV



arXiv:1712.07158

Samples you will work with

Format: PFNano

Like NanoAOD, but with additional information for jet constituents and more
Interesting for us, as is contains the *inputs* used by jet tagging algorithms
Branches described at this website
Important collections: **Jet**, JetPFCands, JetSVs, PFCands, SV
Naming convention for inputs: Jet_[Tagger]_[Constituent-Type (Cpfcan/Npfcan/sv)]_[Feature]_[Index]
(for DeepCSV no constituent type necessary, flat inputs only)

Run 3 samples

Change in jet collection (*NOW* **PUPPI**, *PREVIOUSLY* **CHS**, for more details see Jet Short Exercise)

Processes: semi-leptonic $t\bar{t}$ (and QCD in 170-300 GeV p_T bin)

/TTTo2J1L1Nu_CP5_13p6TeV_powheg-pythia8/Run3Winter22MiniAOD-122X_mcRun3_2021_realistic_v9-v2
/QCD_Pt_170to300_TuneCP5_13p6TeV_pythia8/Run3Winter22MiniAOD-122X_mcRun3_2021_realistic_v9-v2

First Exercise: inputs, targets, outputs

Definitions: Flavour (Truth)

- Important branches in NanoAOD:
  - Jet_hadronFlavour
  - Jet_partonFlavour

```
"Flavour of the jet, numerical codes: "
"isG: 0, "
"isUD: 1, "
"isS: 2, "
"isC: 400, "
"isCC: 410, "
"isGCC: 411, "
"isB: 500, "
"isBB: 510, "
"isGBB: 511, "
"isLeptonicB: 520, "
"isLeptonicB_C: 521, "
"isTAU: 600, "
"isPU: 999,"
"isUndefined: 1000. "
```

decreasing priority

Category	Label	Definition	
b	bb	hadronFlavour==5	N(b-hadrons) > 1
	lepb		N(b-hadrons) == 1 && ΔR(v from b-/c-hadron, jet) < 0.4
	b		N(b-hadrons) == 1 && !lepb
c	c	hadronFlavour==4	-
uds	uds	hadronFlavour!=5 && hadronFlavour!=4	partonFlavour==1 || partonFlavour==2 || partonFlavour==3
g	g		partonFlavour==21

by H. Qu

- Additionally in PFNano:
  - Jet_FlavSplit

Definitions: Outputs, Discriminants

Tagger scores

After application of softmax function, output scores (short: **outputs**) can be interpreted as **probability** for a jet to be of a certain **class** (which classes are taken into account can vary for different taggers)

Discriminants

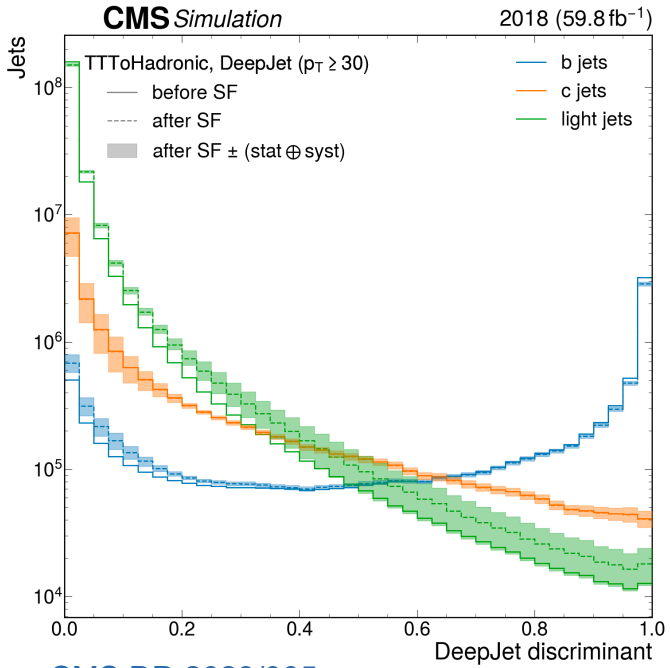
Given certain probability scores and for Monte Carlo simulation a certain subset of the available jets falling into a specific category X or Y, a **discriminant** may be defined via

$$\mathrm{XvsY} = \frac{\mathrm{Prob(X)}}{\mathrm{Prob(X)} + \mathrm{Prob(Y)}}$$

Data / MC for discriminants

Tagger can be evaluated for data as well (inputs exist as well there)
But you will see discrepancies between data & MC!
➔ Scale factors

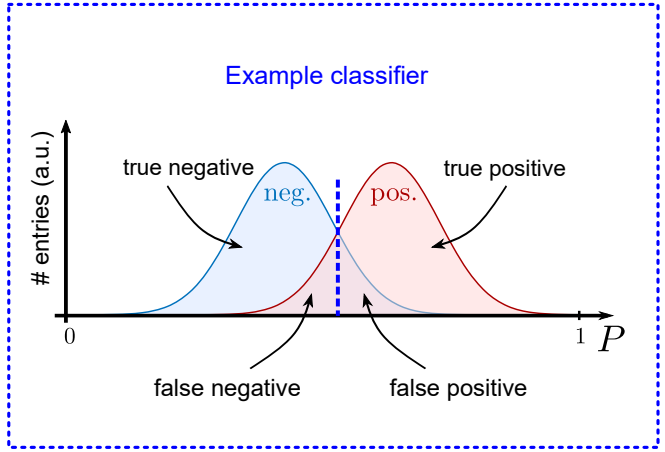
Example: BvsAll

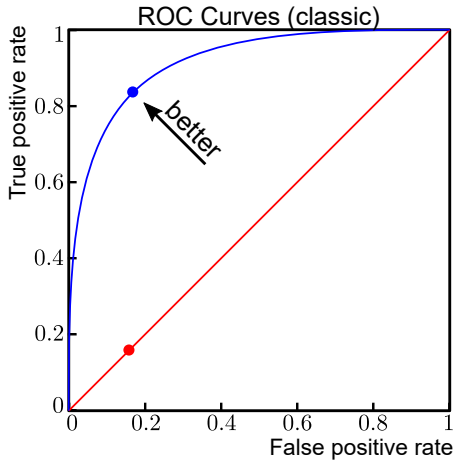


CMS DP-2023/005

Second Exercise: performance

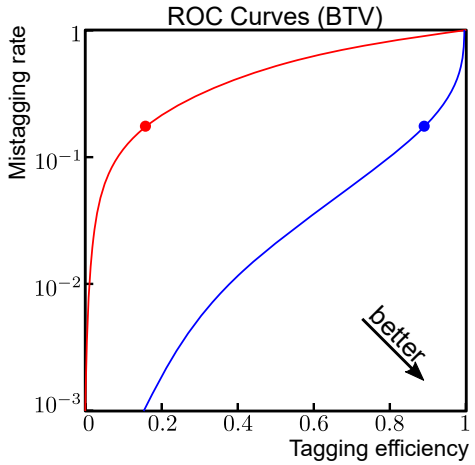
Performance metrics: ROC curves and AUC



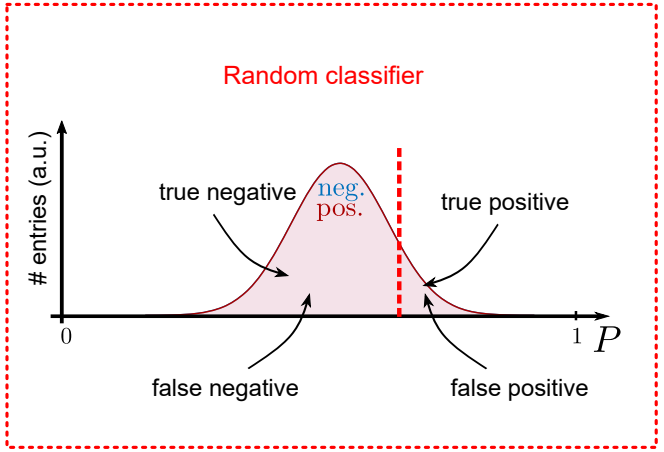


$0.5 < \mathrm{AUC} < 1$

$\mathrm{AUC} = 0.5$



AUC: area under curve
(e.g. under the ones on the left ←)



A. Stein

Backup

Useful links

Recommended:

BTV Public Wiki (Under construction)
BTV Internal Wiki (Docs)

Slowly phased out / outdated:

BTV TWiki

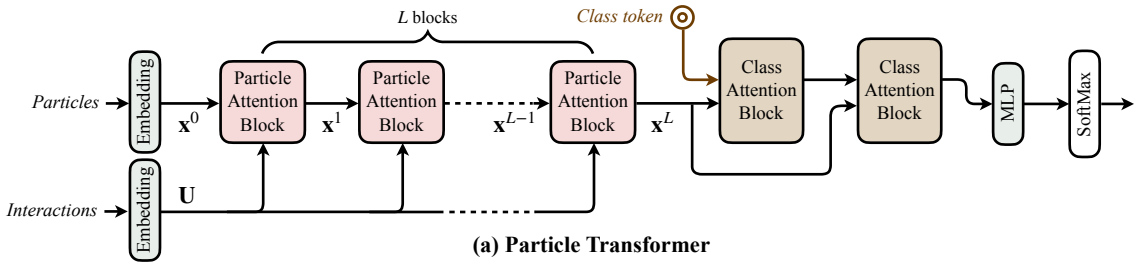
Stay up to date with new flavour tagging integrations for your preferred data tier:

BTV at XPOG Gitlab

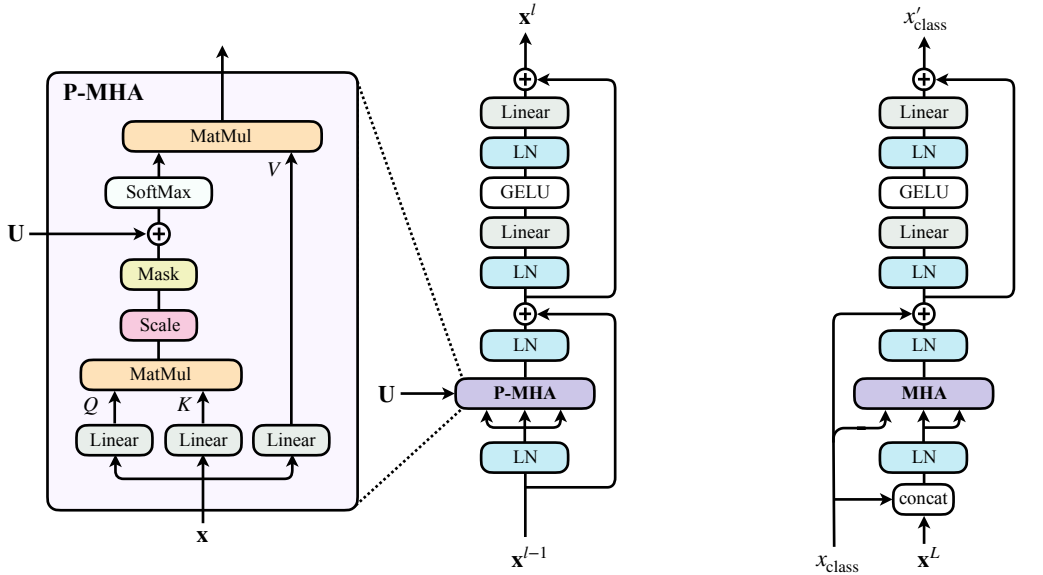
New developments for Run 3: Transformer models

However, with higher performance comes higher susceptibility

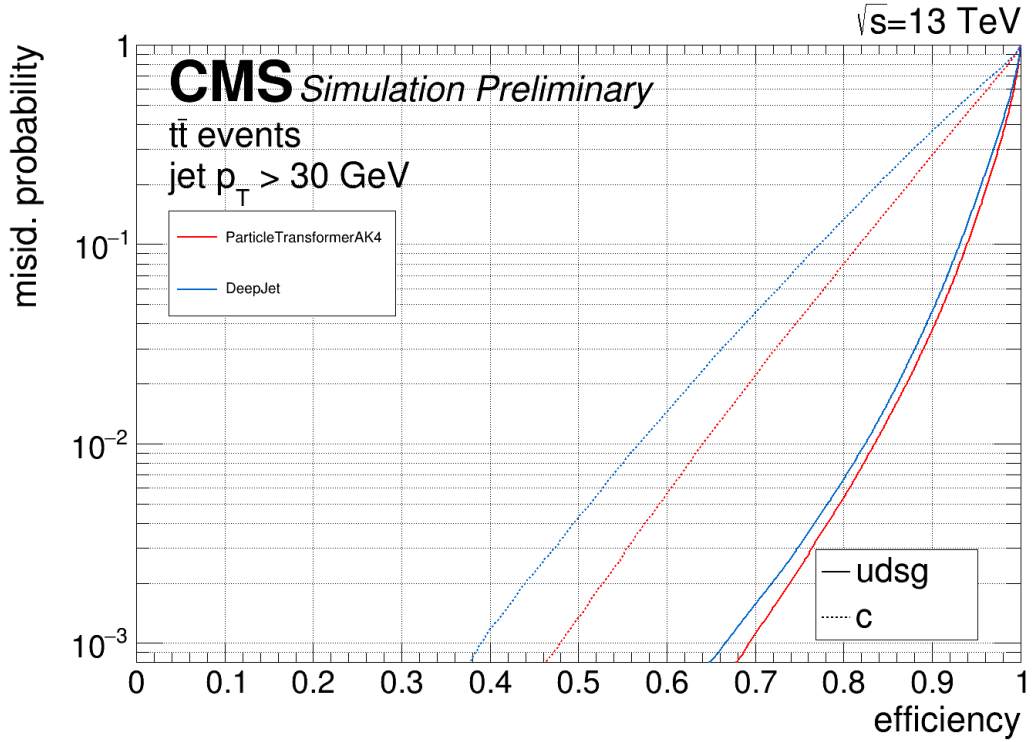
- Performance improvements via new architecture, attention mechanism



(a) Particle Transformer



(b) Particle Attention Block arXiv:2202.03772 (c) Class Attention Block



CMS DP-2022/050

- More features, including **pair-wise** features between charged and neutral candidates as well as secondary vertices