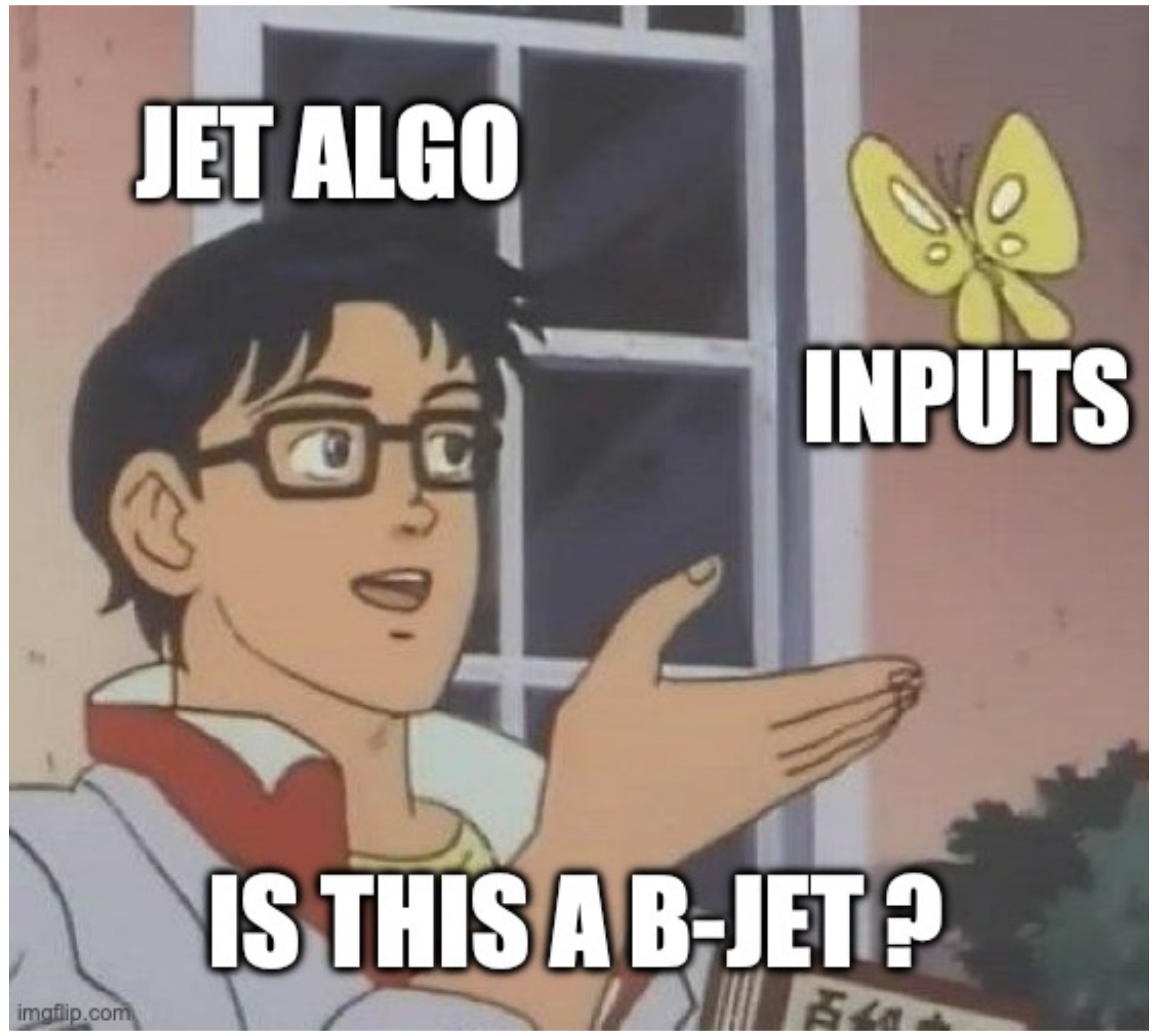
Deep Learning for jets: towards a unified jet algorithm

Alexandre De Moor

Today's menu

1. Jet tagging 101: what is a jet and Deep Learning?



Today's menu

1. Jet tagging 101: what is a jet and Deep Learning?

2. Jet algorithm evolution: from likelihood ratio to Transformer models

Today's menu

1. Jet tagging 101: what is a jet and Deep Learning?
2. Jet algorithm evolution: from likelihood ratio to Transformer models
3. Unified Jet approach: everyone joins the tagging battle!

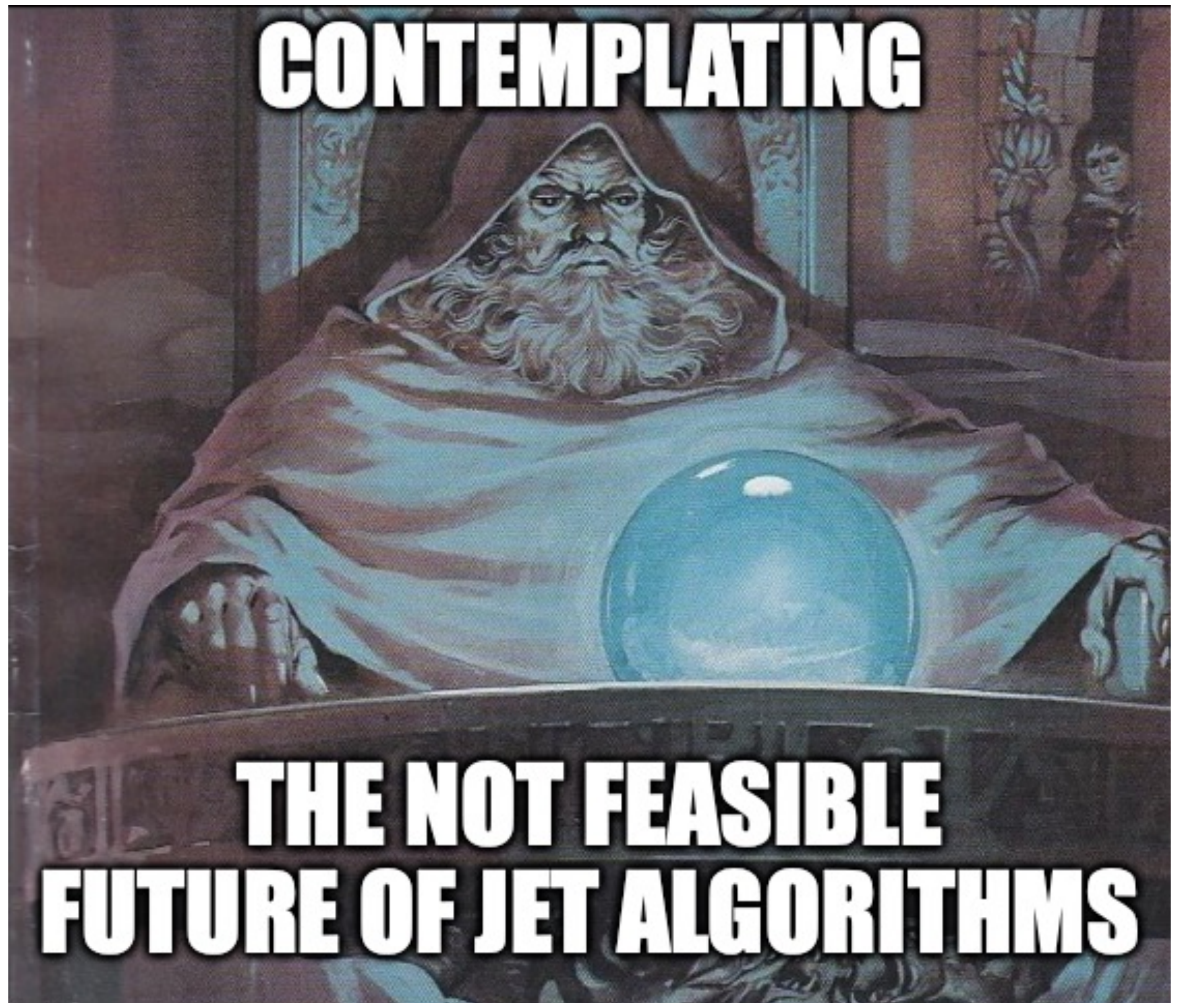
Today's menu

1. Jet tagging 101: what is a jet and Deep Learning?

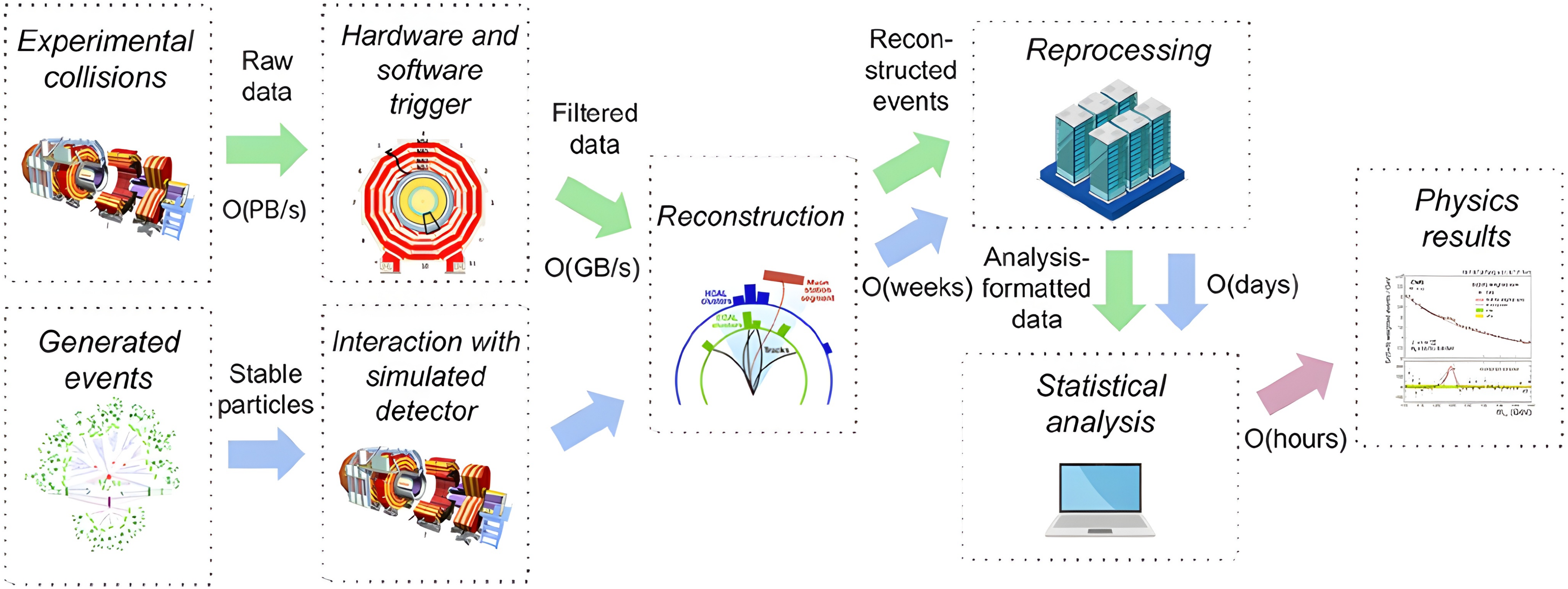
2. Jet algorithm evolution: from likelihood ratio to Transformer models

3. Unified Jet approach: everyone joins the tagging

4. What's next: towards fully unified world models



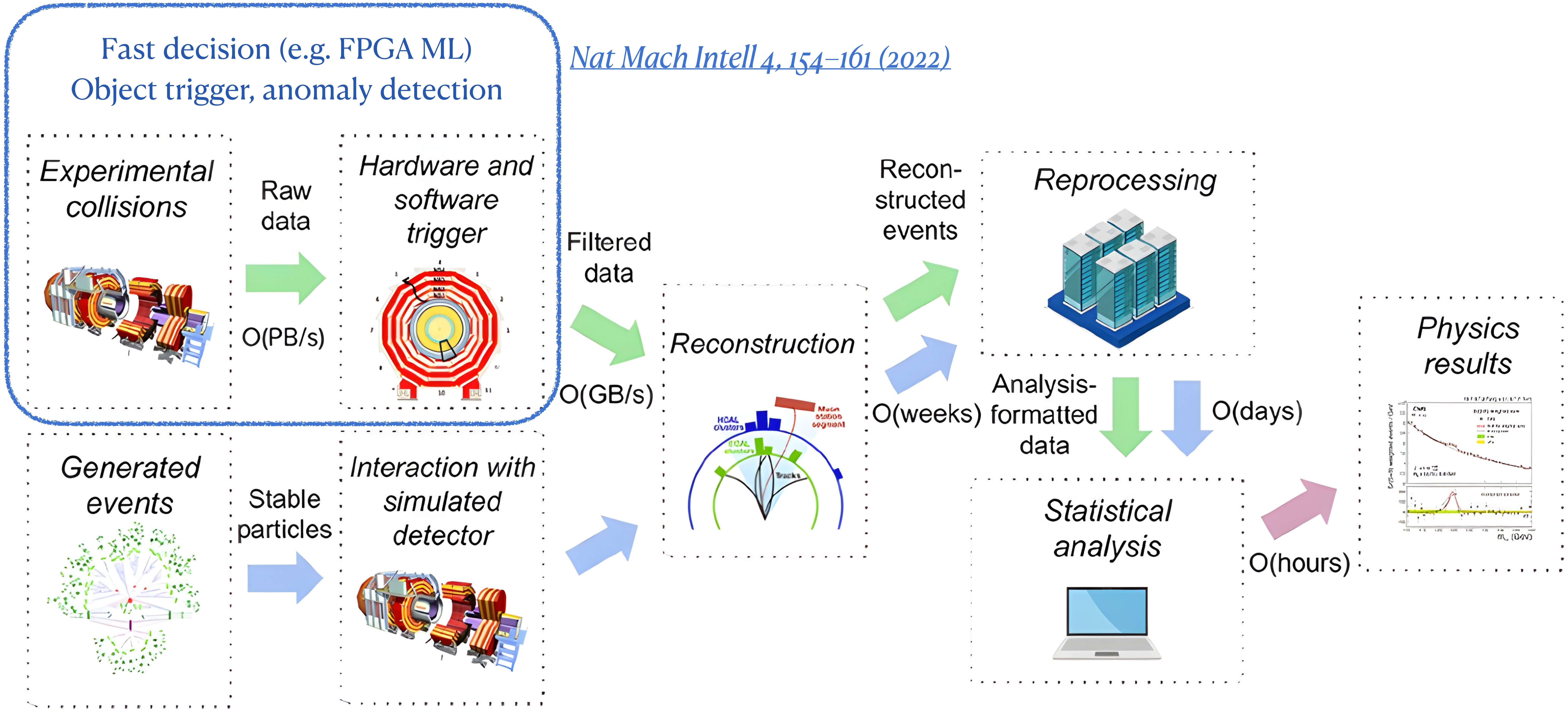
HEP data workflow and ML



Front.Big Data 4 (2021) 661501

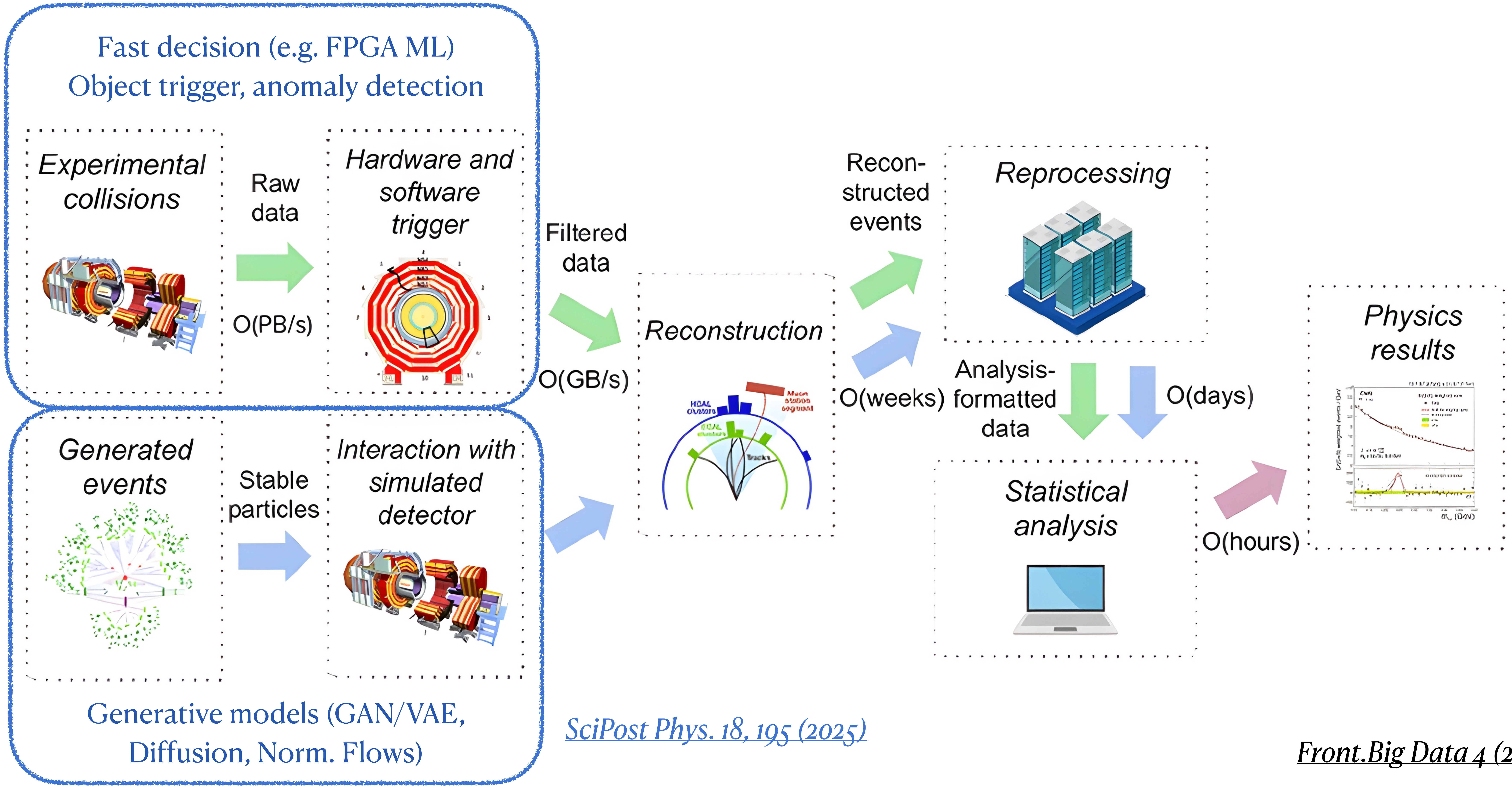
HEP data workflow and ML

Nat Mach Intell 4, 154–161 (2022)



Front.Big Data 4 (2021) 661501

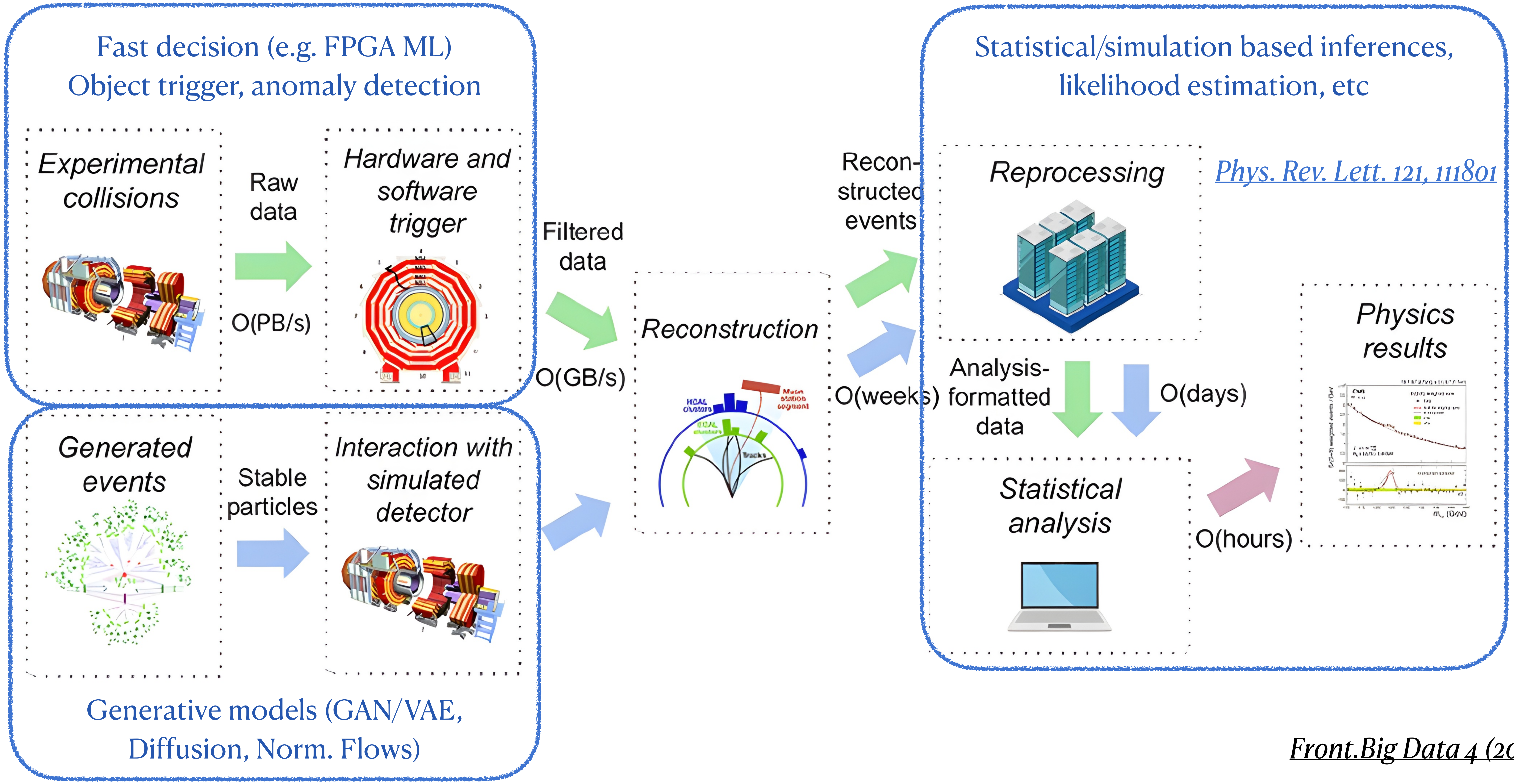
HEP data workflow and ML



SciPost Phys. 18, 195 (2025)

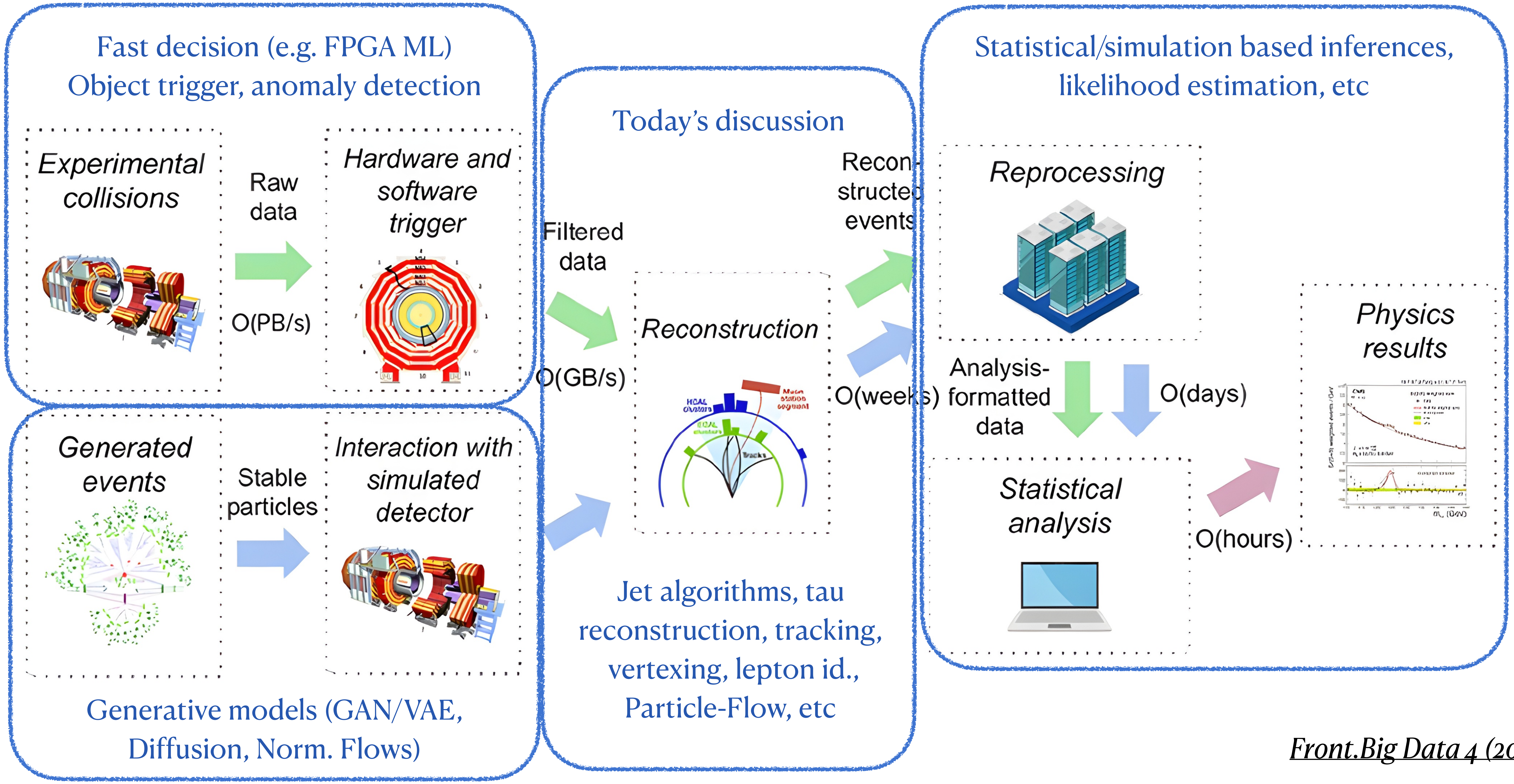
Front.Big Data 4 (2021) 661501

HEP data workflow and ML



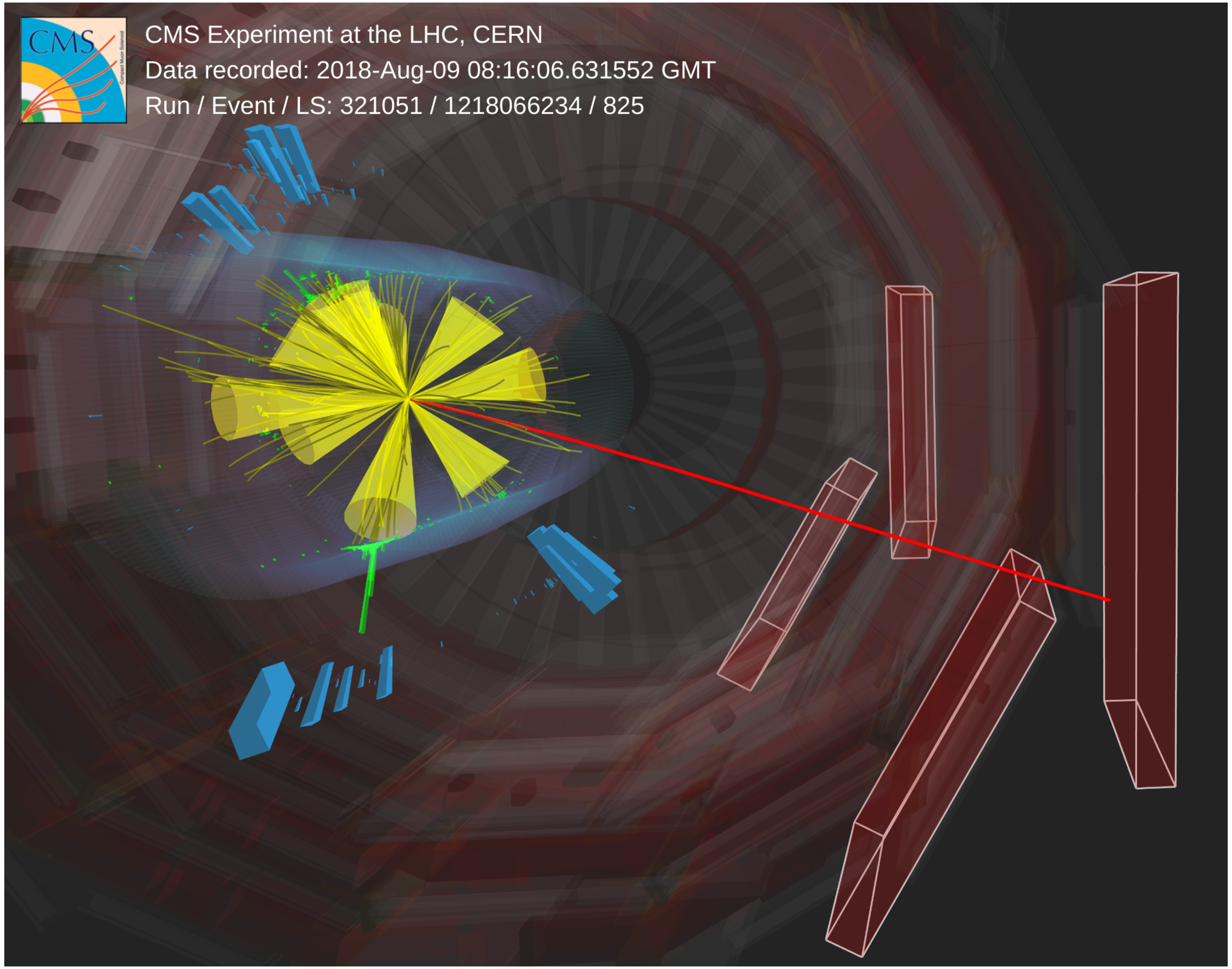
Front.Big Data 4 (2021) 661501

HEP data workflow and ML



Front.Big Data 4 (2021) 661501

Jet tagging 101: definitions

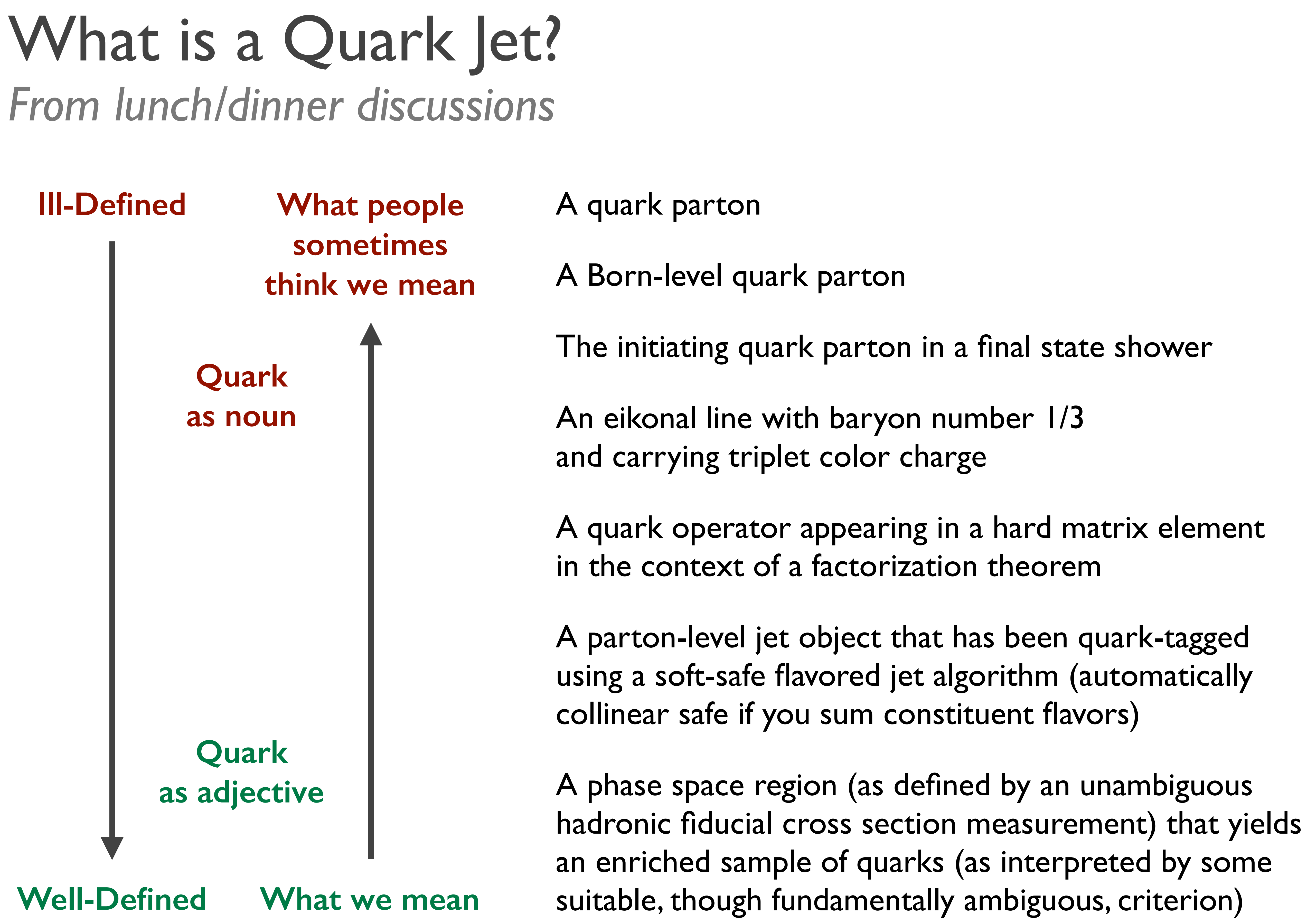


- Jet: a narrow cone of collimated stable particles
- Jet clustering: agglomeration law associating the stable particles into a jet (e.g. anti-kT)
- Jet's origin : Jet tagging
- Jet's properties : jet (energy) regression

Disclaimer: focus on small jet radius, but the discussion generalizes to large radius one too

CMS-PHO-EVENTS-2024-025

Jet tagging 101: definitions



For quark jet: necessary to define the flavor of the originating quark or gluon

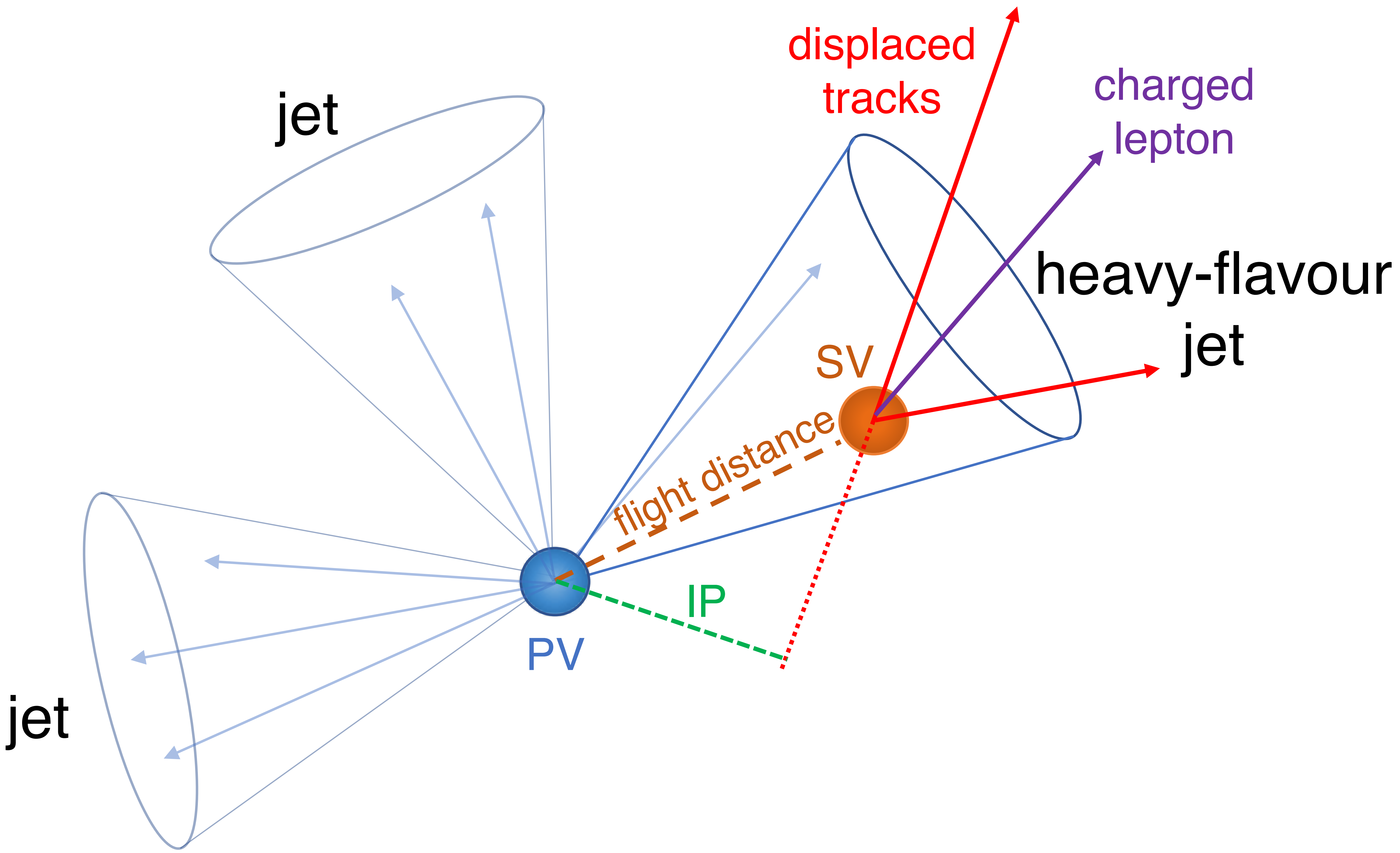
Jet flavor: A long and complex discussion (with still no consensus)

In CMS: defined via ghost hadron/ parton association

Jet Energy Correction (JEC): correct for detector response, pileup effects, and other biases.

Les Houches 1-19 June 2015
JINST 13 (2018) P05011

Jet tagging 101: signatures

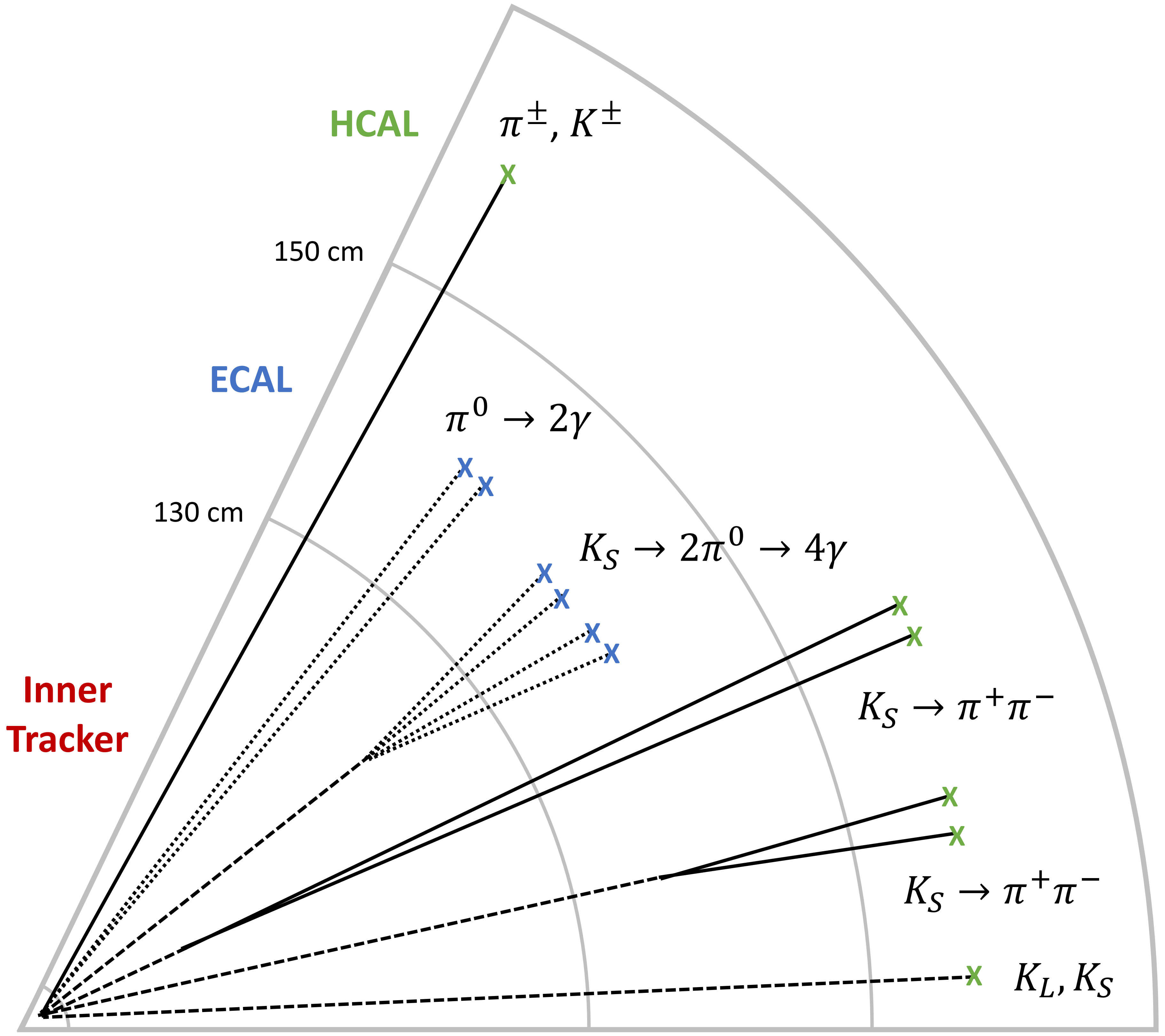


Heavy flavor (b-c) jets: b (c) hadron with a sufficient lifetime, 1.5 (1.0) ps

- Creation of a secondary vertex (SV)
- Displaced tracks
- 20% (10%) of the b (c) jets containing a soft lepton from the heavy hadron decay

JINST 13 (2018) P05011

Jet tagging 101: signatures



Strange jet: weaker signatures vs u-d jets

- Neutral composition of the jet and its energy
- Charged kaon identification not possible at our energy scale for CMS

arXiv:2003.09517

Deep Learning 101: ML

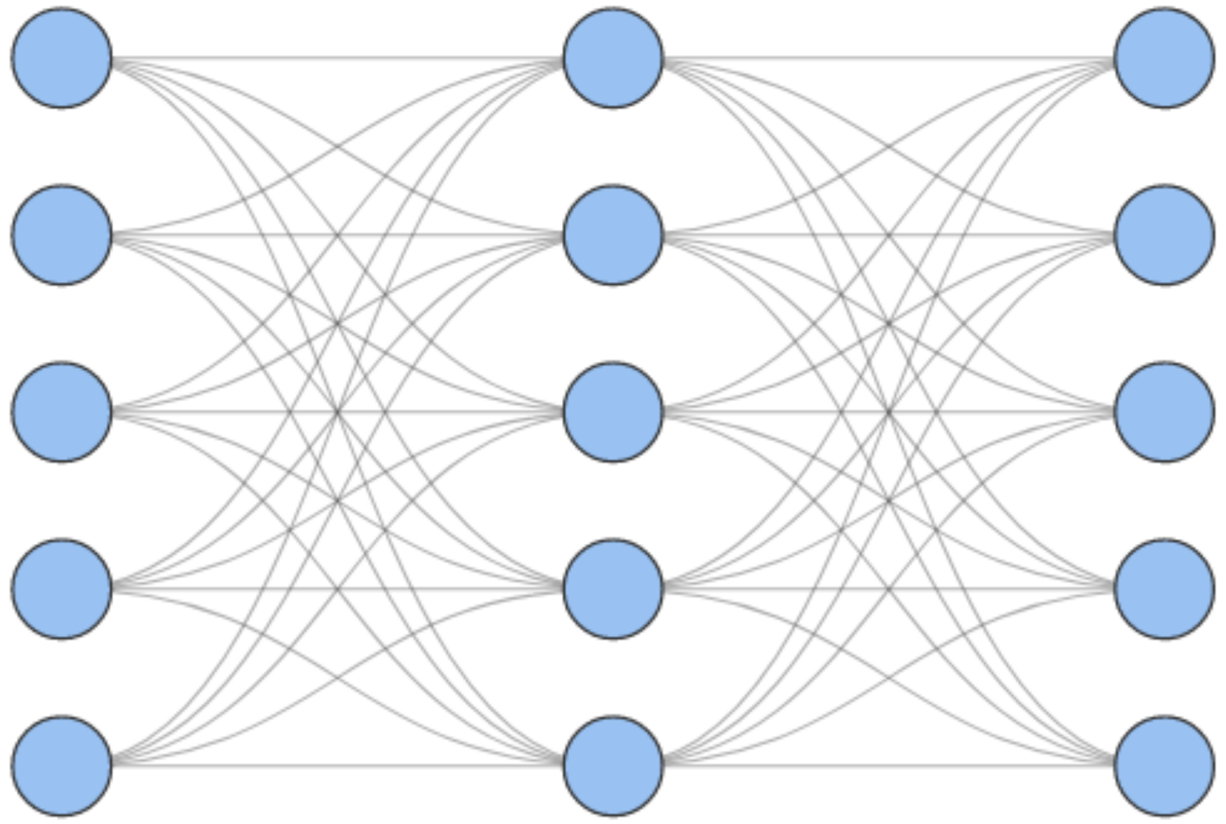
What is Machine Learning ?

"Machine Learning is the science of getting computers to act without being explicitly programmed."
- Arthur Samuel, 1959

Key ingredients:

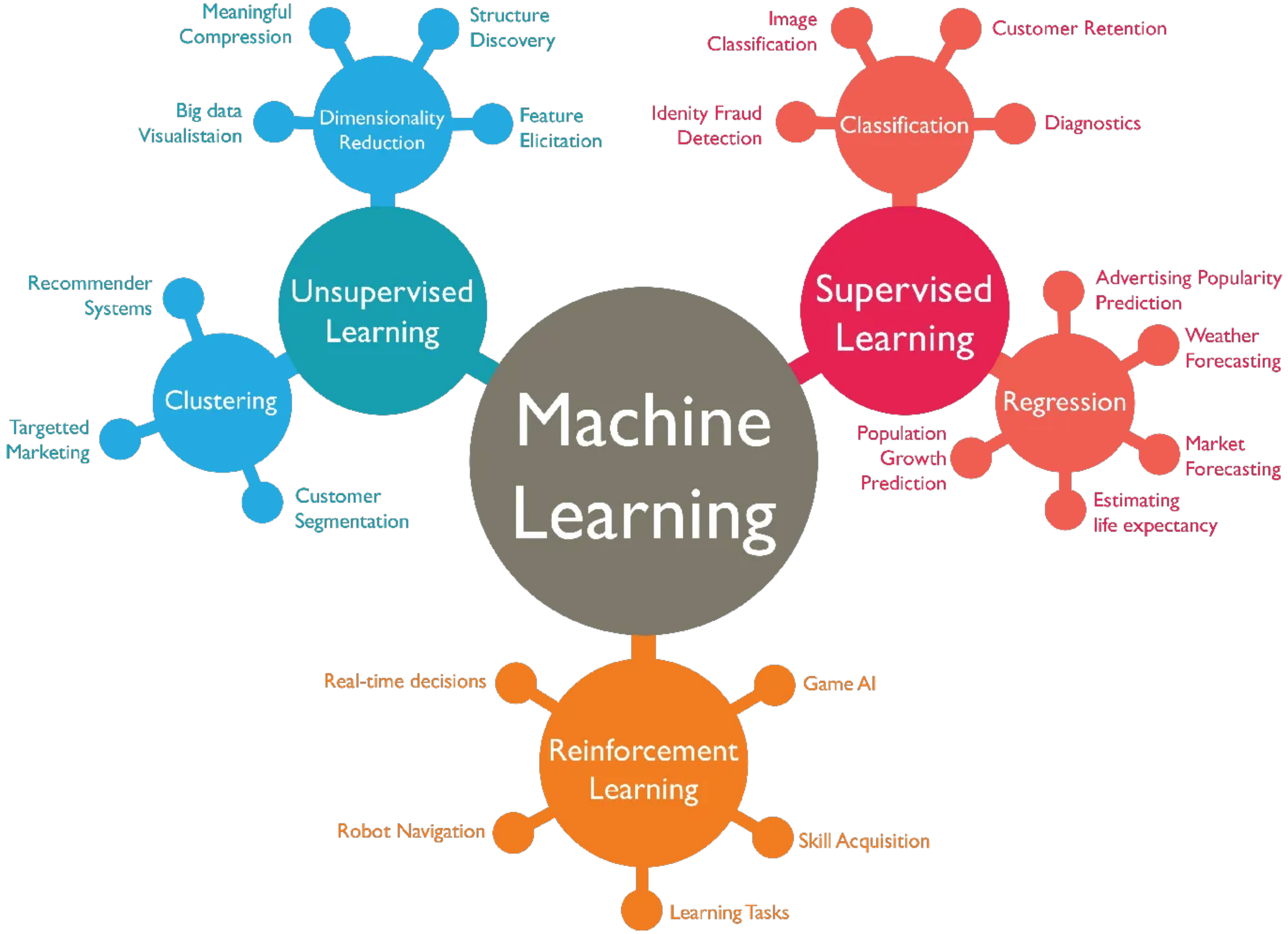
- Data and an objective: what and in which conditions?
- Objective (loss) function: mapping the prediction and the ground truth
- Learning strategies: how to adjust the prediction
- Structure: how to modelize the prediction from the inputs

$$o_j^1(\bar{x}, \bar{W}^1) = a^1\left(\sum_i x_i W_{ij}^1 + b_j^1\right)$$

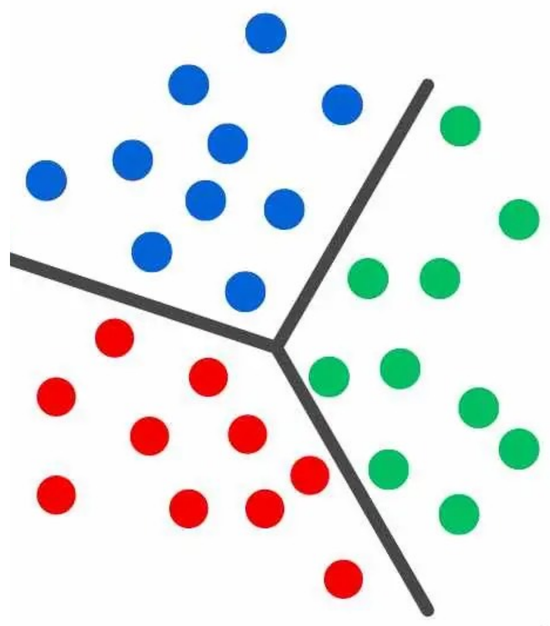


Only focus on ML with artificial neural networks (Deep Learning)

Deep Learning 101: types of learning

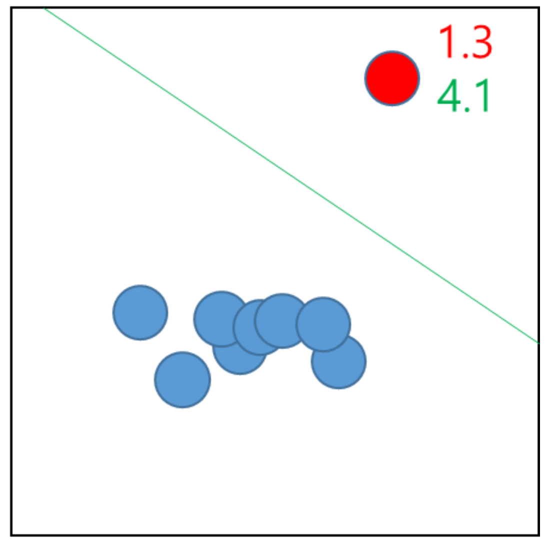


- Supervised learning (task specific model):



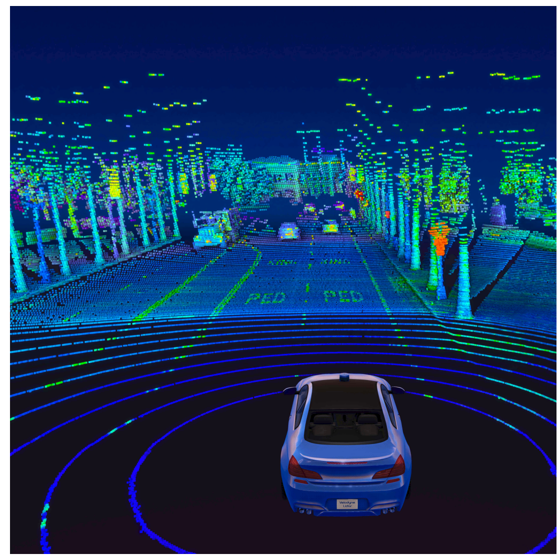
Assigned a task to predict an outcome from labelled data(classification or regression)

- Unsupervised learning (general model):



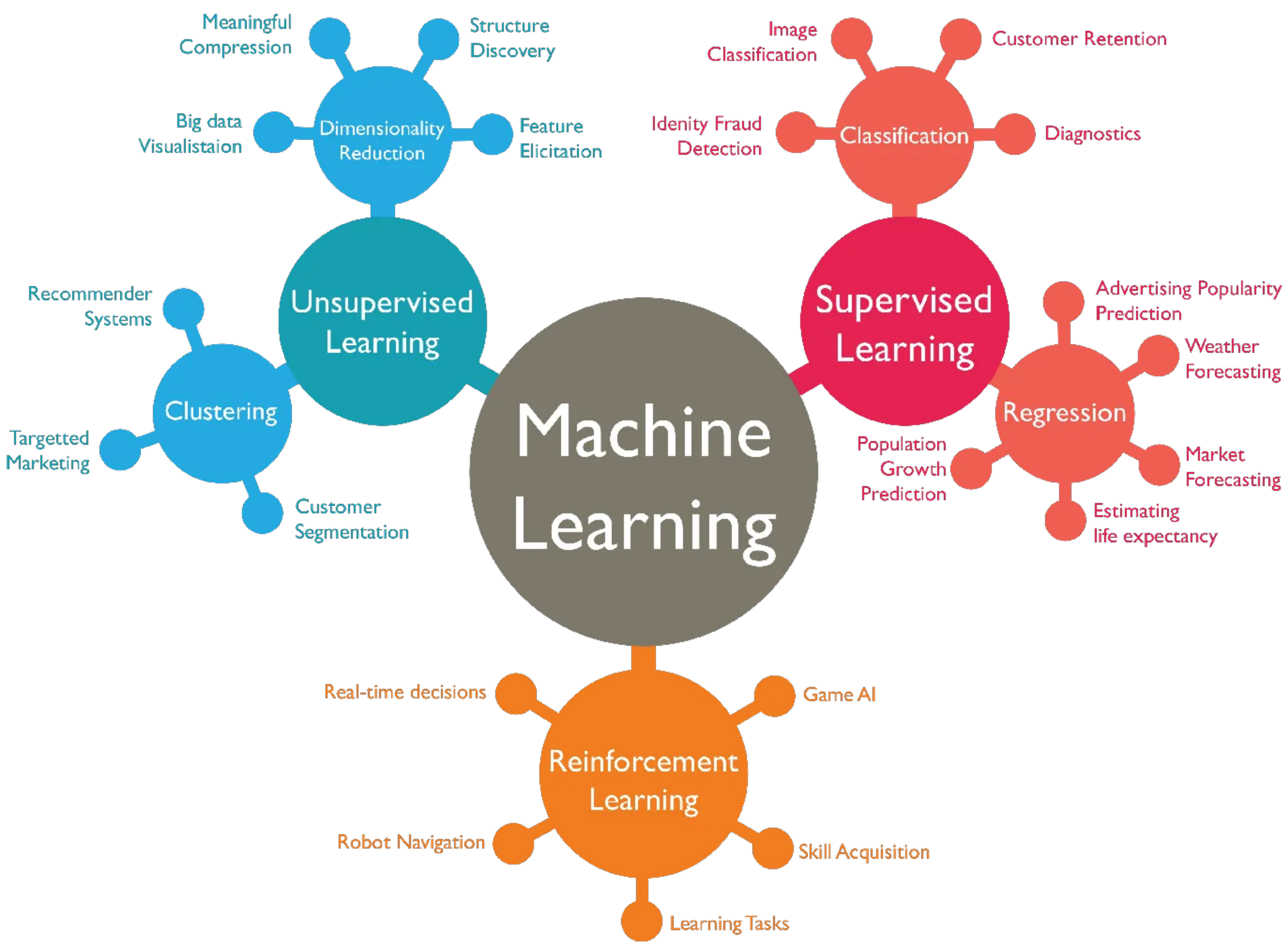
Discover patterns and properties from unlabeled data (LLMs, anomaly detection, etc)

- Reinforcement learning (acting model)



Decision making model, adjusting it's next move from a reward cost function (autonomous driving, Gaming, etc)

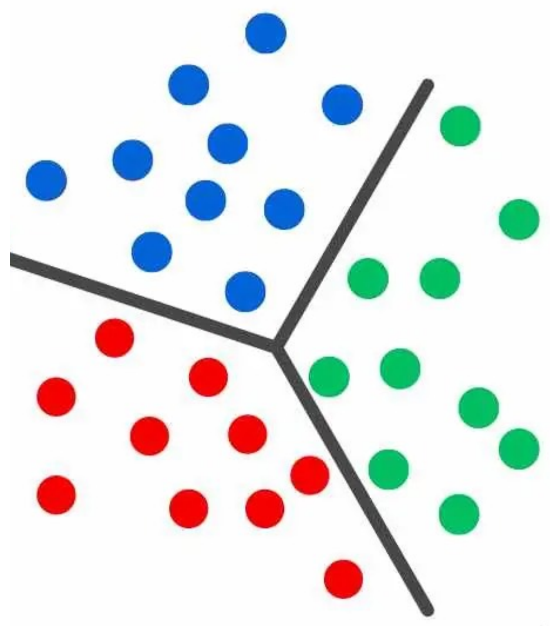
Deep Learning 101: types of learning



Today we stay in this area

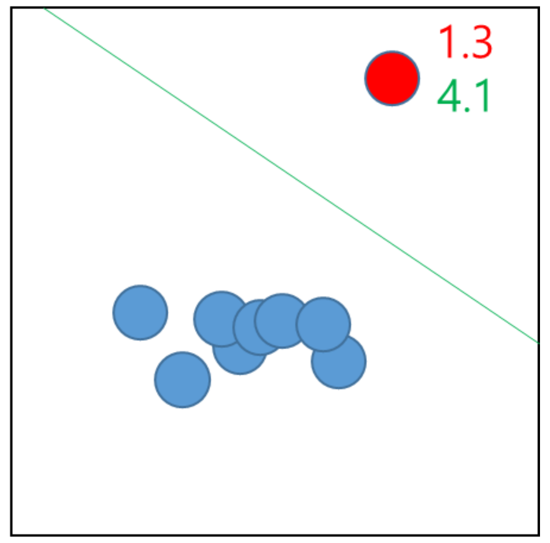
- Supervised learning (task specific model):

Assigned a task to predict an outcome from labelled data(classification or regression)



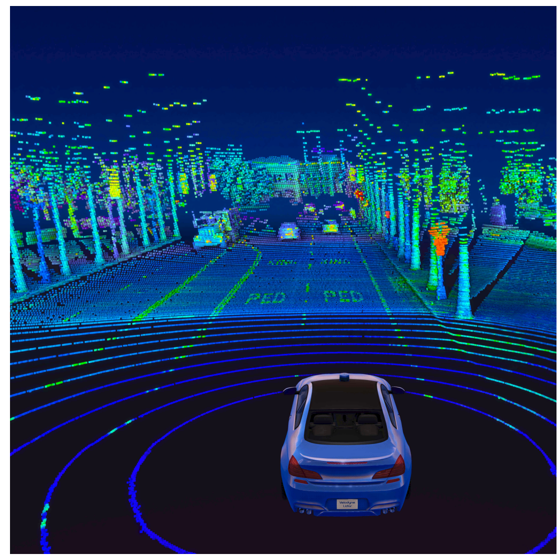
- Unsupervised learning (general model):

Discover patterns and properties from unlabeled data (LLMs, anomaly detection, etc)



- Reinforcement learning (acting model)

Decision making model, adjusting it's next move from a reward cost function (autonomous driving, Gaming, etc)



Deep Learning 101: Gradient descent

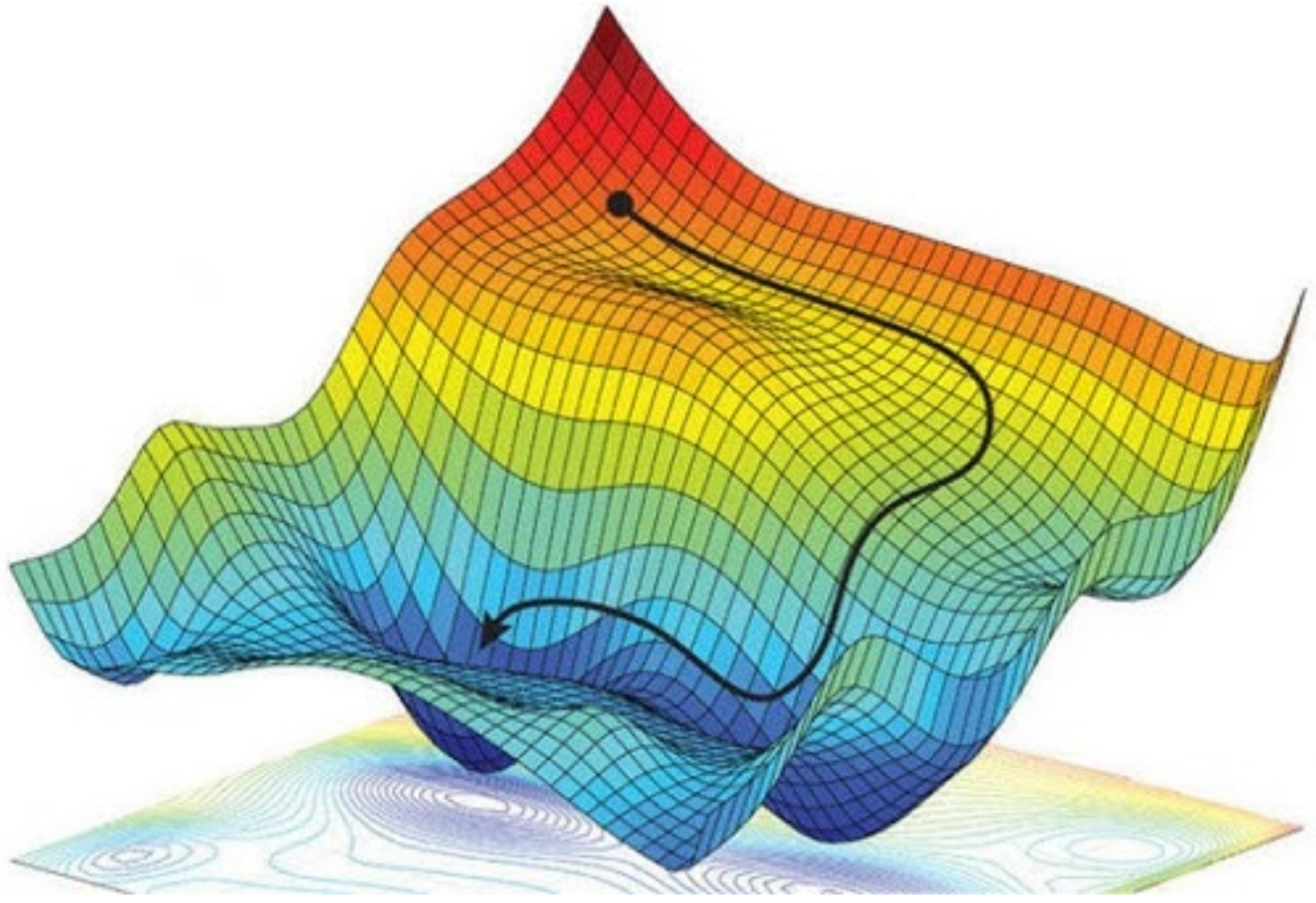
Core Concept: An iterative optimization algorithm for finding the minimum of a function by following the direction of steepest descent

Physical Analogy: Imagine releasing a ball on a mountainous landscape. The ball naturally rolls downhill, following the steepest slope at each point, until it reaches a valley—a local minimum

In gradient descent, the "landscape" is defined by a cost function *L*, and we iteratively update parameters by moving opposite to the gradient:

$$W_t = W_{t-1} - \lambda \cdot \frac{\partial \mathcal{L}}{\partial W_{t-1}}$$

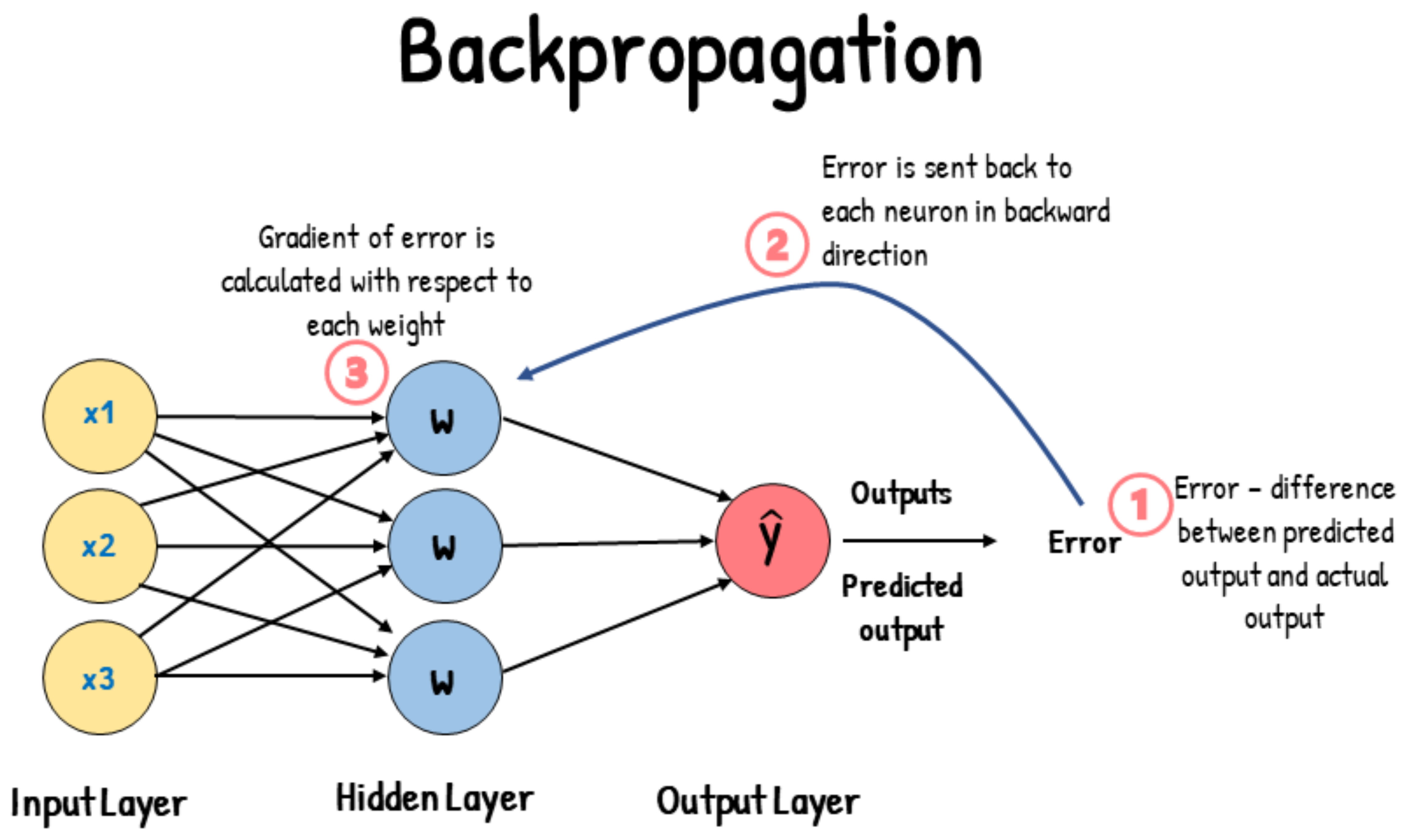
Key Insight: The learning rate λ controls step size—too large and we overshoot; too small and convergence is slow. The gradient provides local directional information, analogous to measuring the slope beneath the ball.

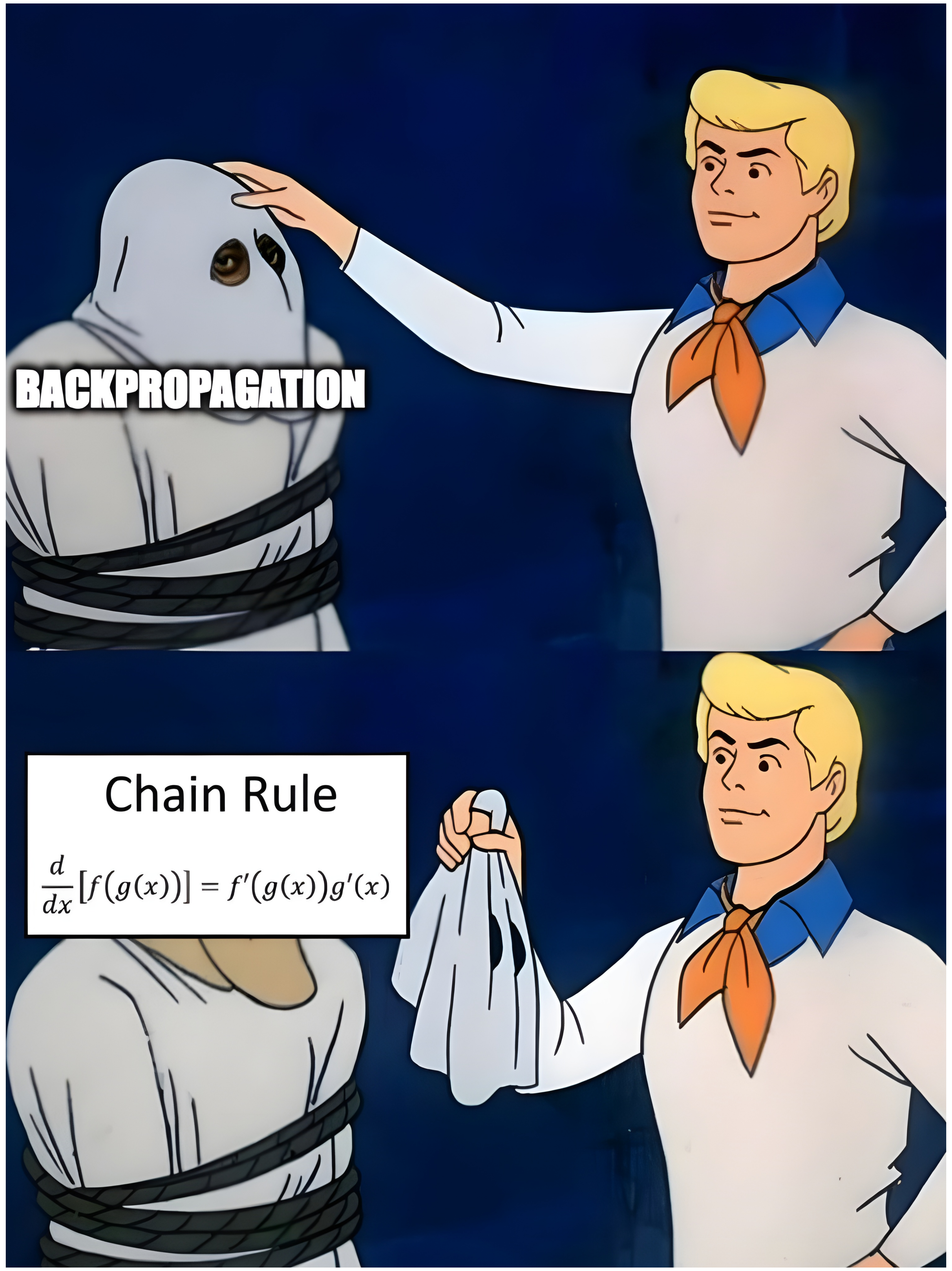


Foundation paper: *[Nature 323, 533–536 (1986)]*

Deep Learning 101: Gradient descent

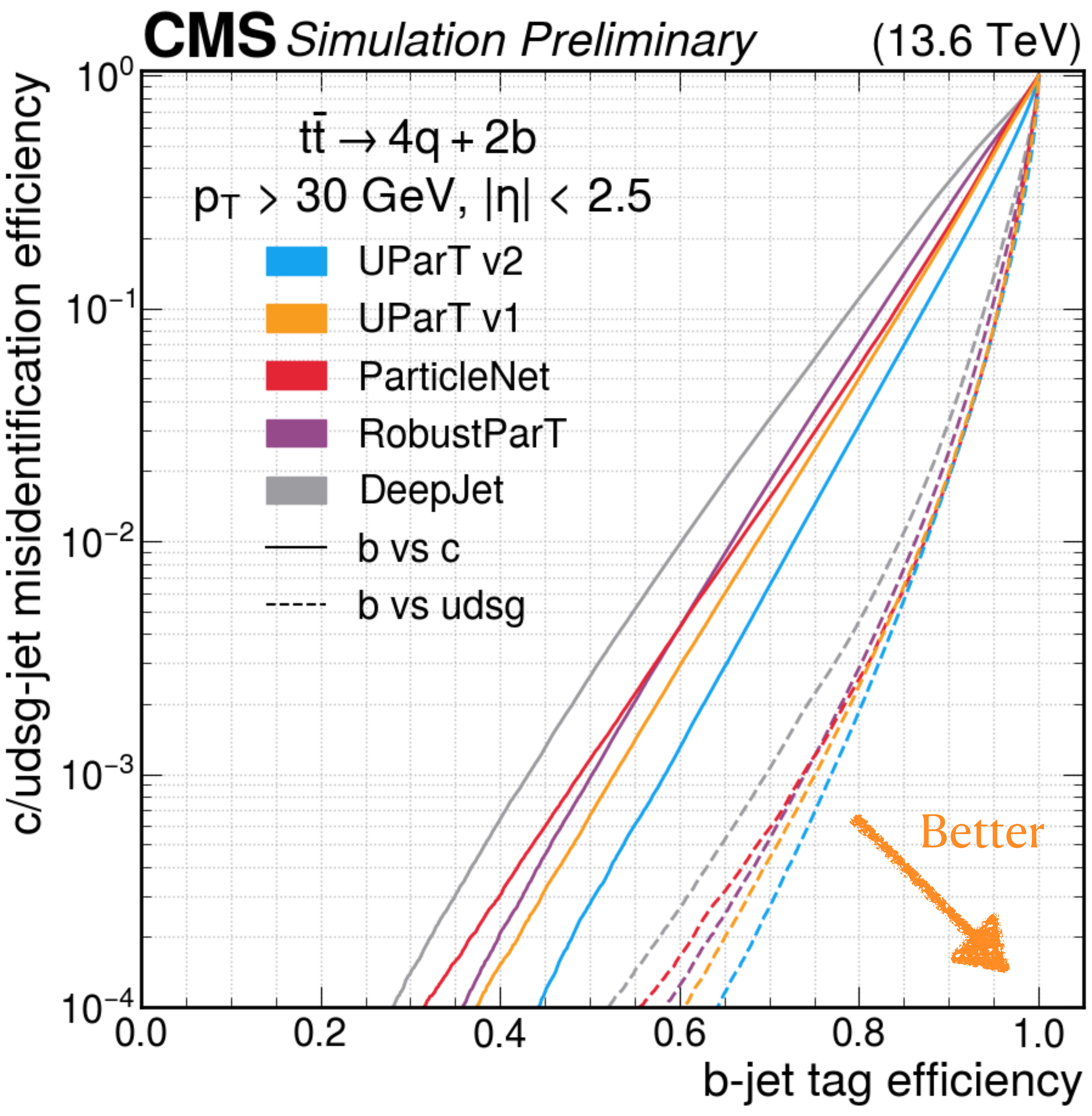
Update of all the layers: Chain rules across the weights matrices and functions of the neural network structure - Backpropagation





The BTV rosetta stone

Type I error

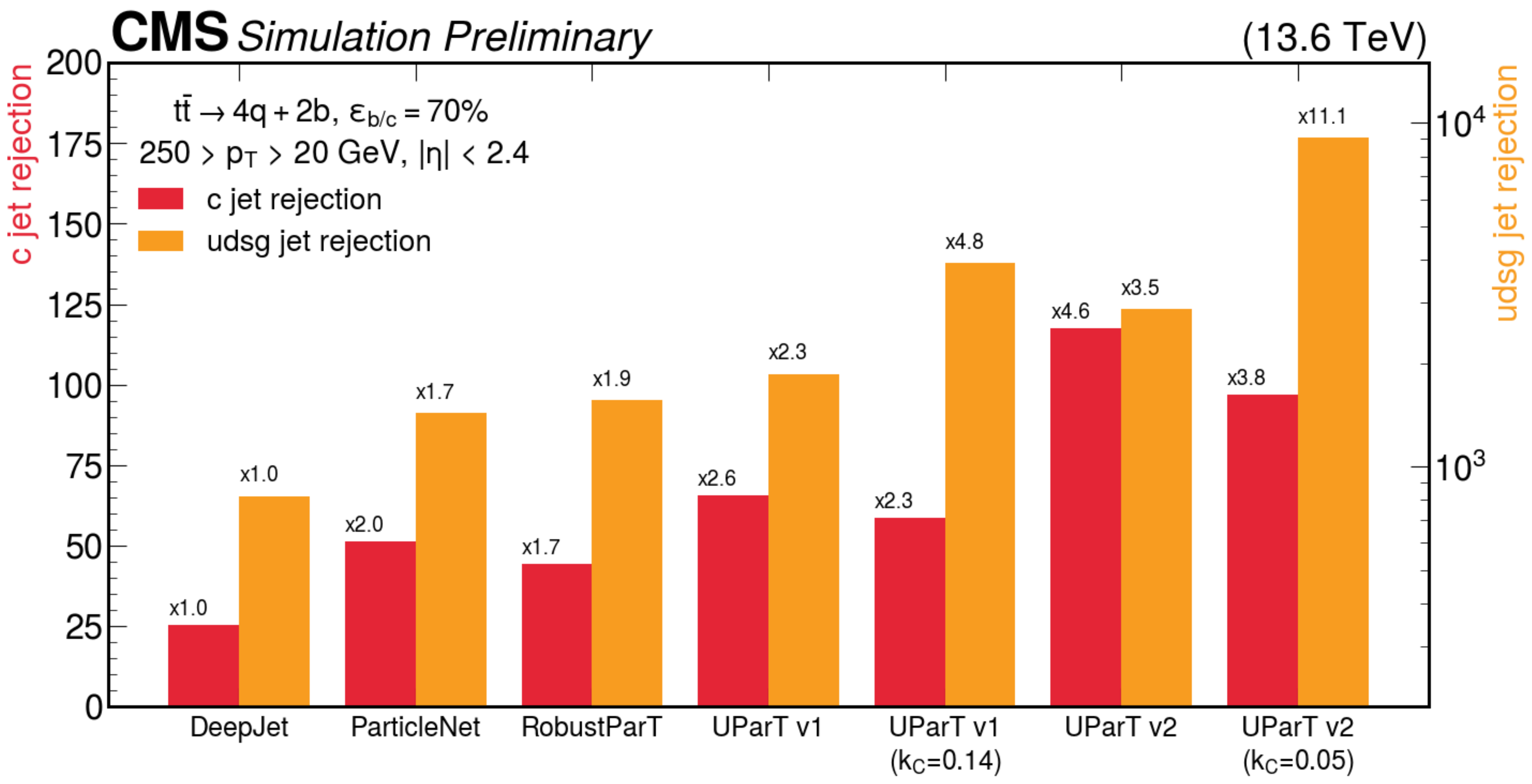


Sensitivity (1-Type II error)

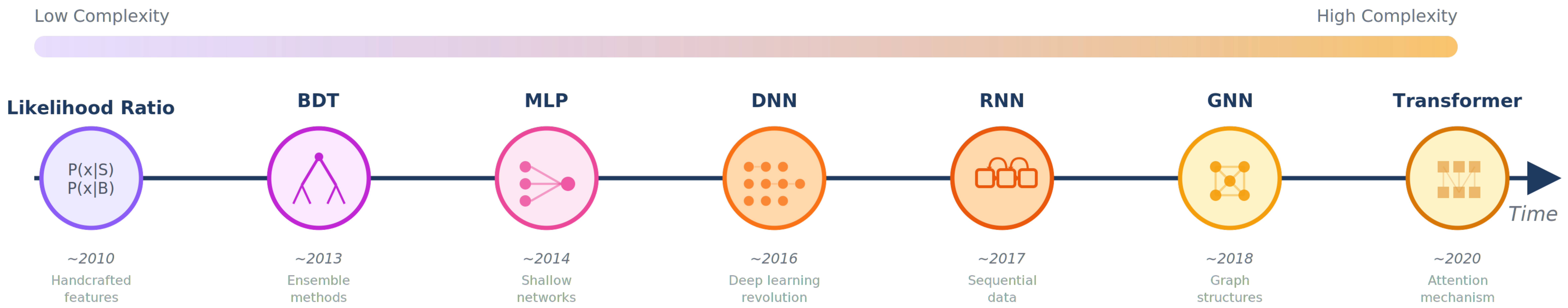
[CMS-DP-2025-081](CMS-DP-2025-081)

Two types of metrics:

- Working points (WPs): signal efficiency at fixed background mis.id. rate
- Rejection rate: invert of the background mis.id. rate at a fixed signal efficiency

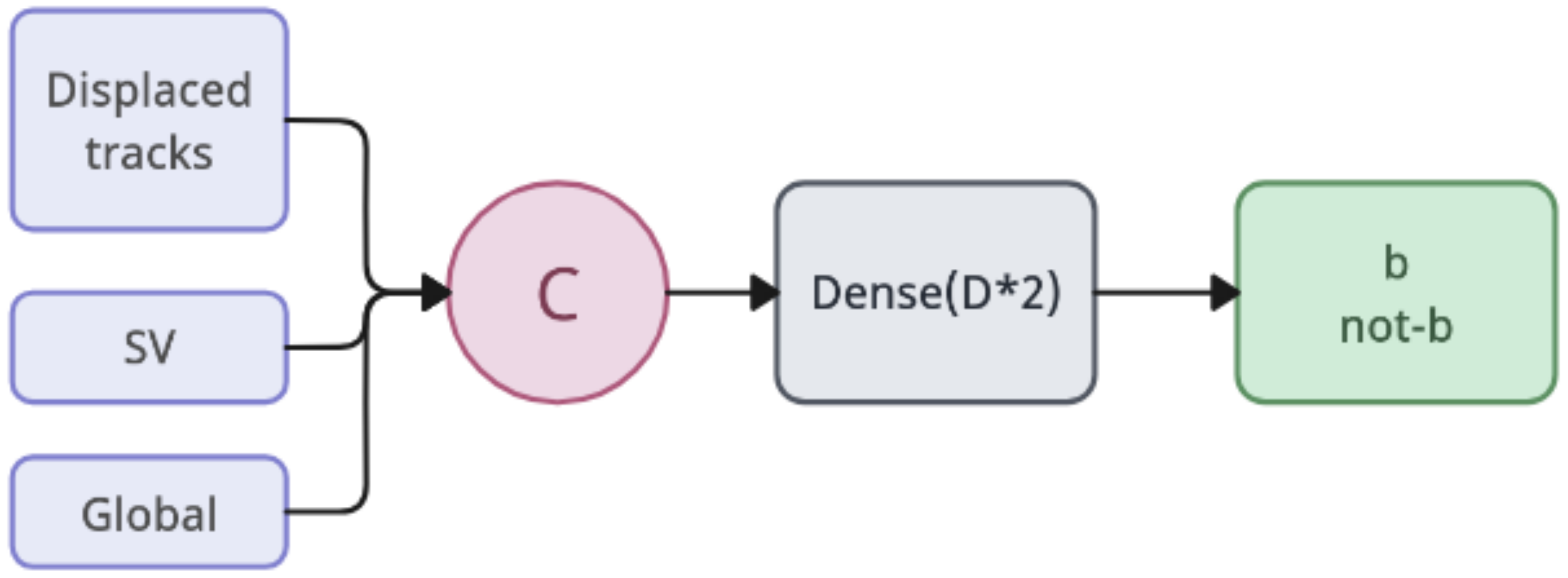


Jet algorithm evolution: a matter of representation

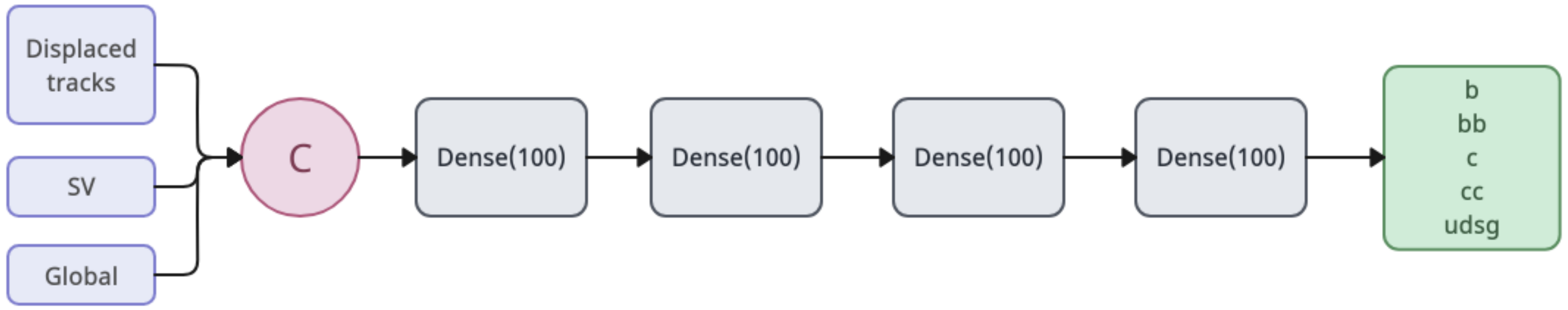


Jet algorithms: a matter of representation

- The evolution of algorithms followed how we represented our jet inputs.
- First naïve approach use jet level inputs + handcrafted displaced tracks.
- ML methods used evolved from likelihood ratio (JetProbability algorithm) to Boosted Decision Tree/Shallow networks (Combined Secondary Vertex algorithms - CSVs)
- First multi-layer DNN arrived at the end of this era: DeepCSV



CSVv2 architecture



DeepCSV architecture

[JINST 13 (2018) P05011]

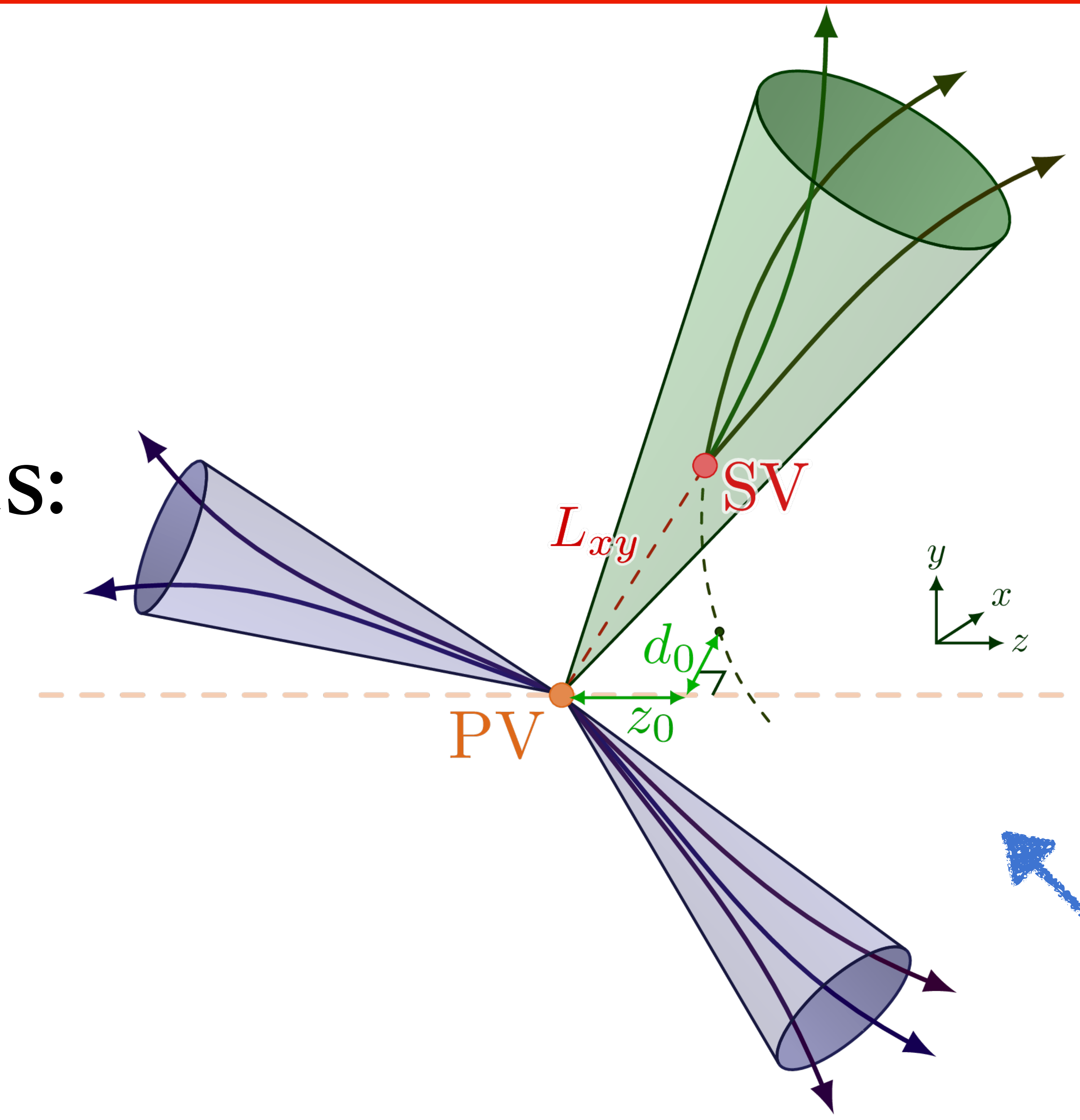
Jet as an image

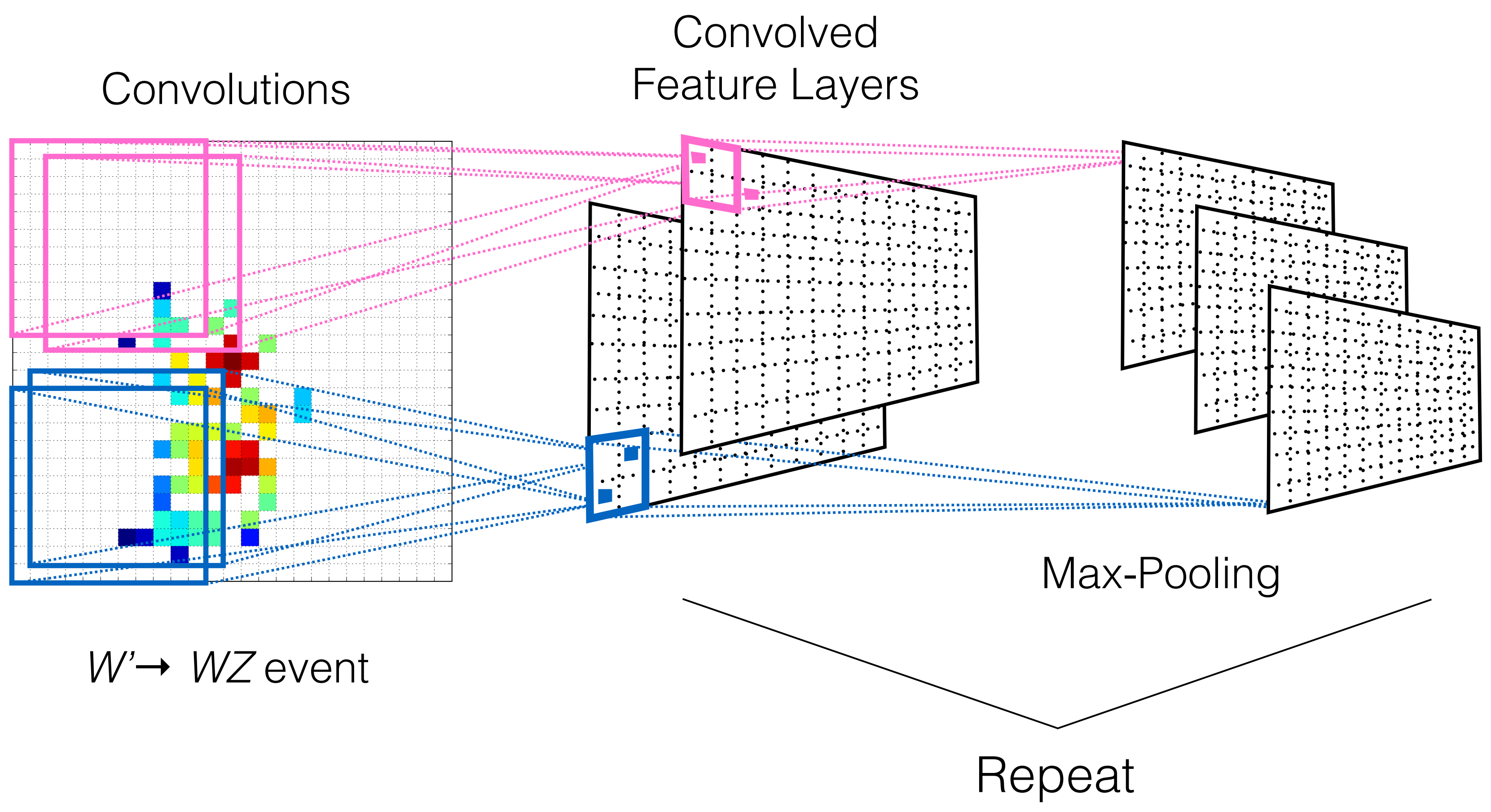
arXiv:1511.05190
arXiv:2012.09719

Jet seen as an image made of pixel hits:

- Jets are ‘sparse’ structures in an inhomogeneous medium
- Pixel representation unadapted to the physics meaning

Based on Convolutional neural networks (CNN)



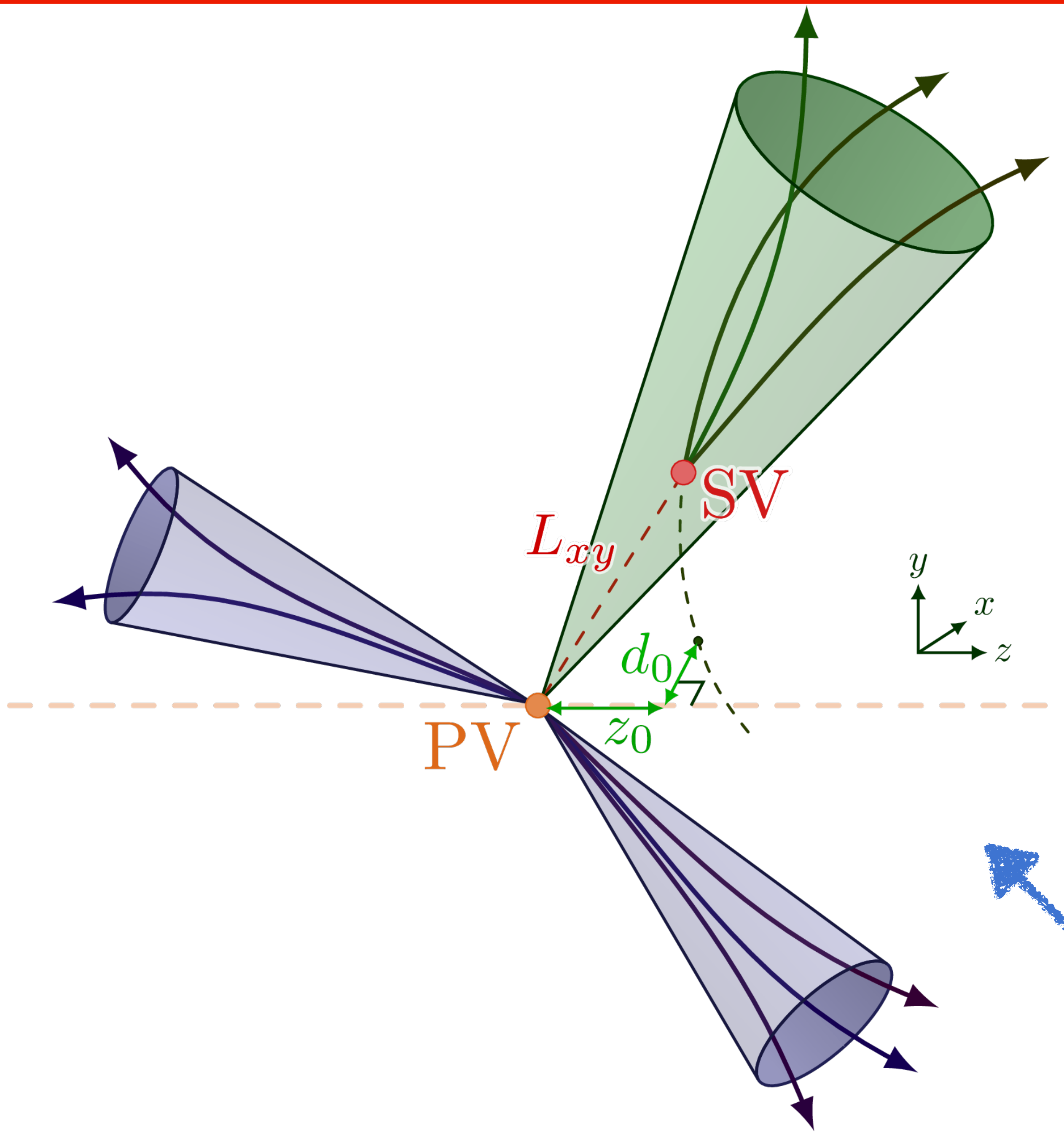


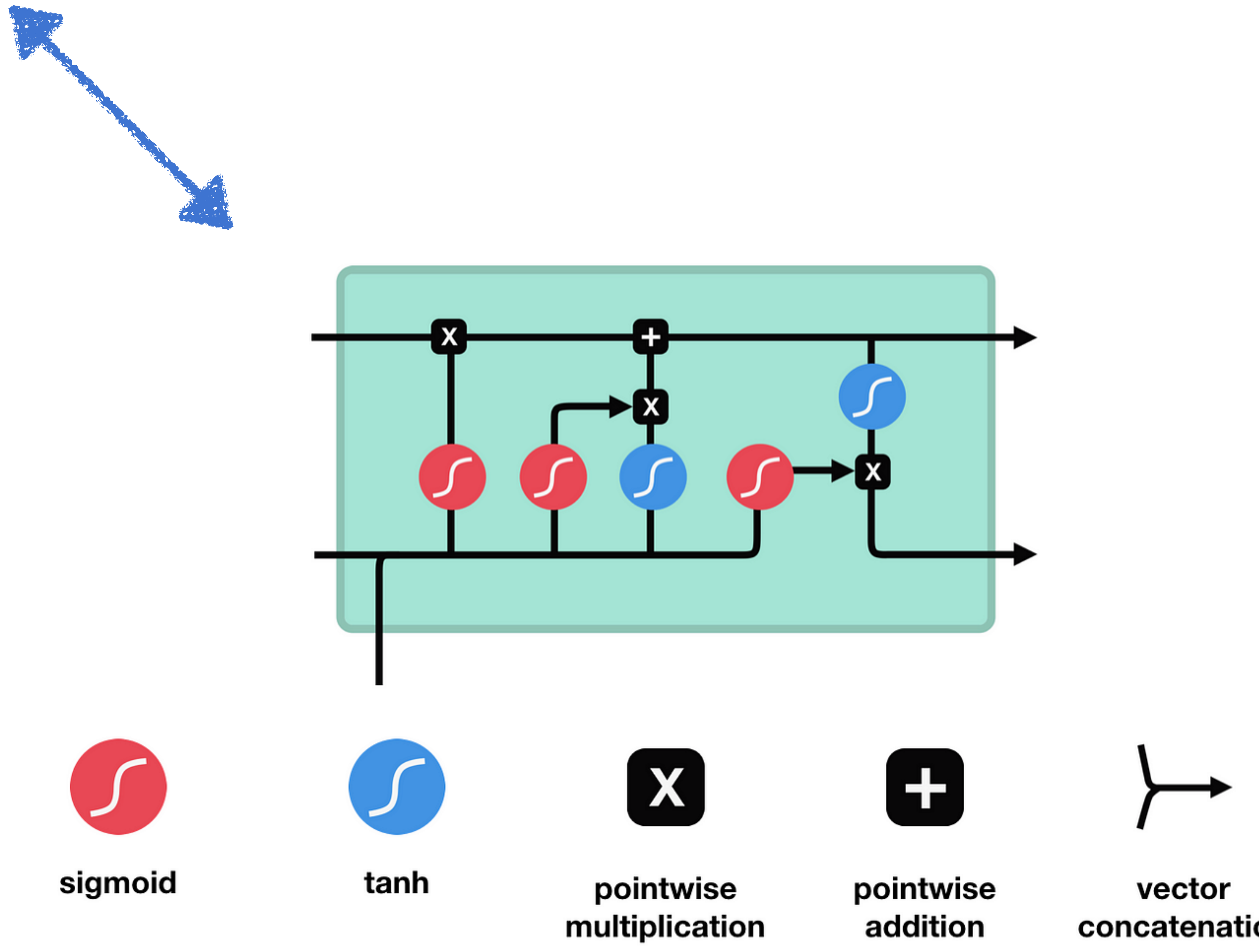
Jet as a sequence

[JINST 15 (2020) P06005](#)
[JINST 15 P12012](#)

First turning point of jet algorithms :

- Exploits the substructure of the jets (constituent based approach)
- Processes constituents one by one in order to obtain a sequence-based latent space
- First 'Deep' neural networks
- Parametrization O(100) to O(10,000) and inputs O(10) to O(500)





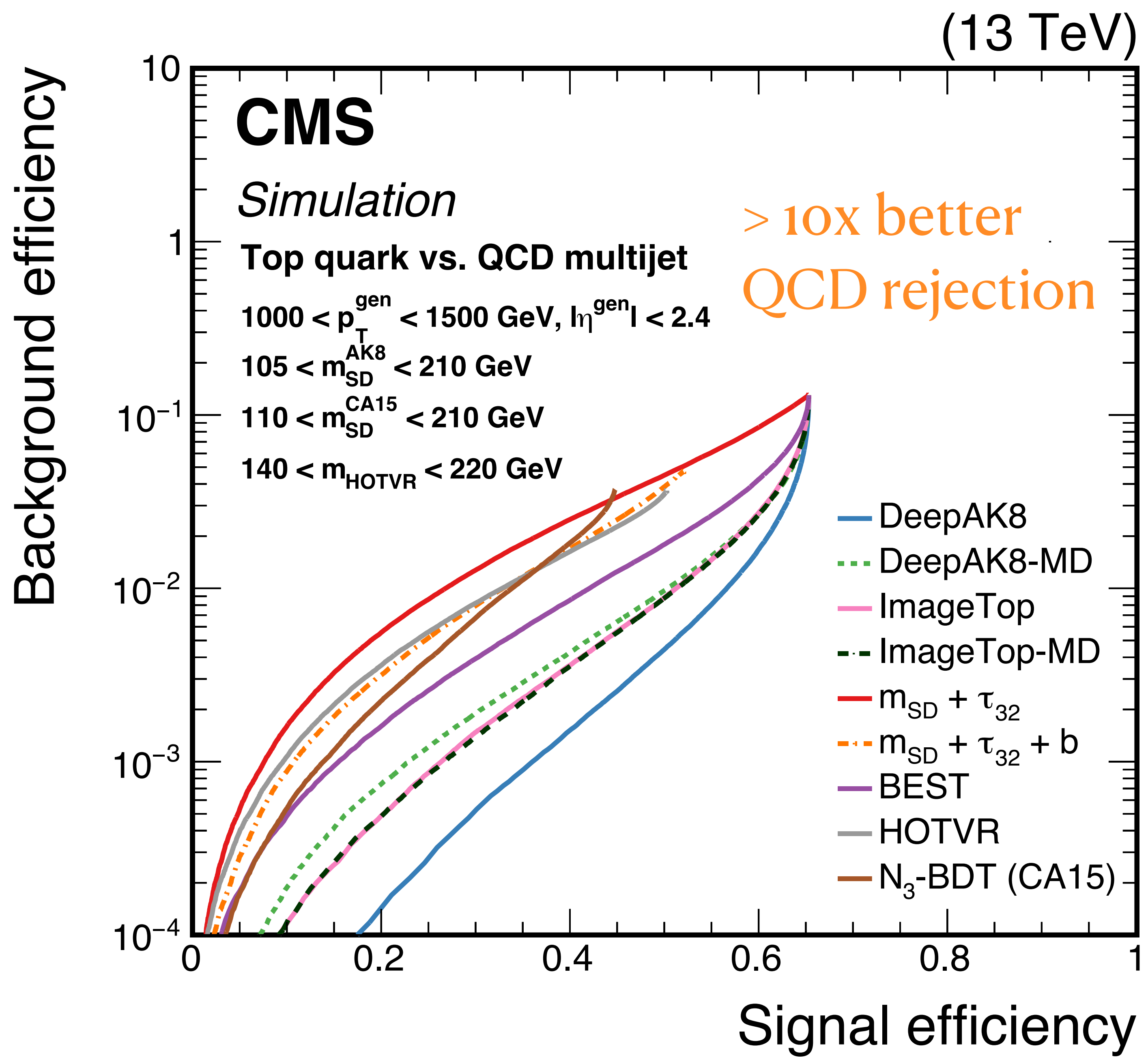
A complex and hard to train architecture

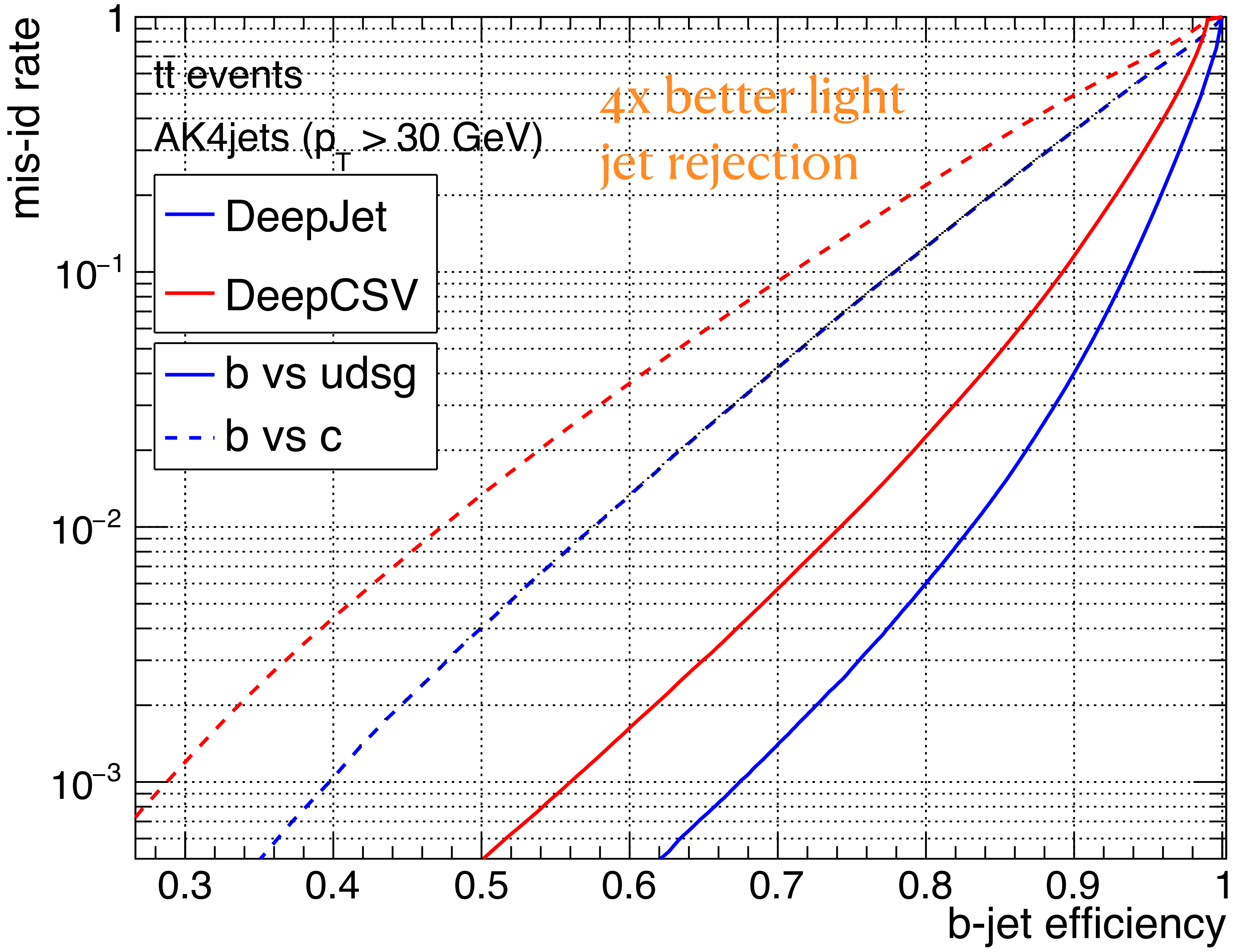
Jet as a sequence: DeepAK8 and DeepJet

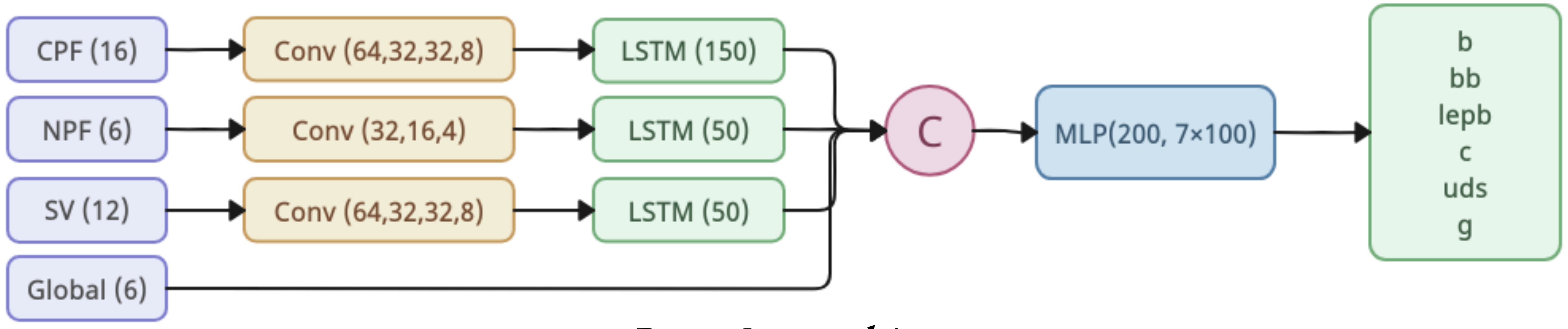
JINST 15 (2020) P06005
JINST 15 P12012

Widely adopted as the state-of-the-art at the end of Run2 at CMS.

Two models: DeepAK8 for large jet radius and DeepJet for small jet radius







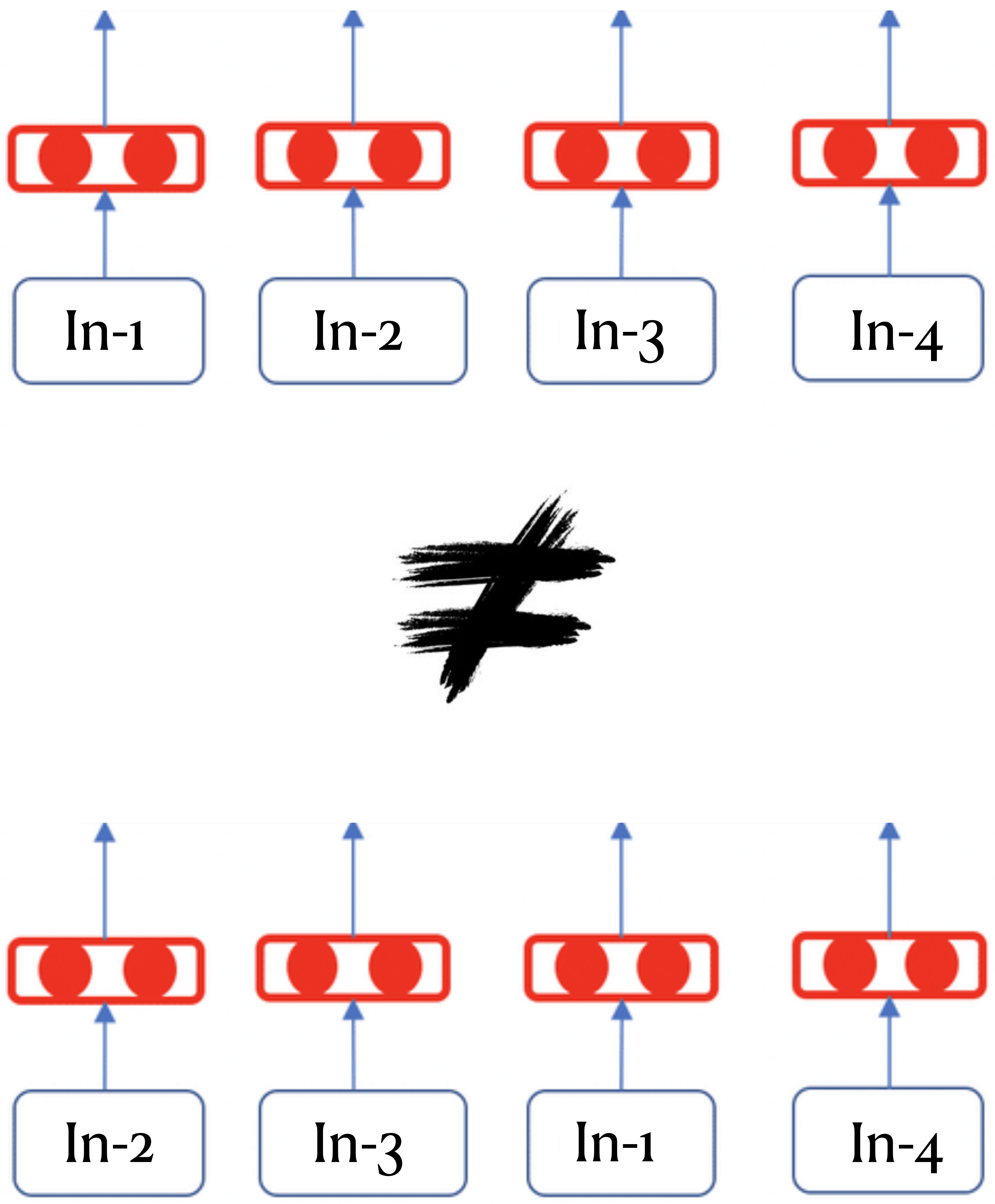
DeepJet architecture

Jet as a sequence: limit

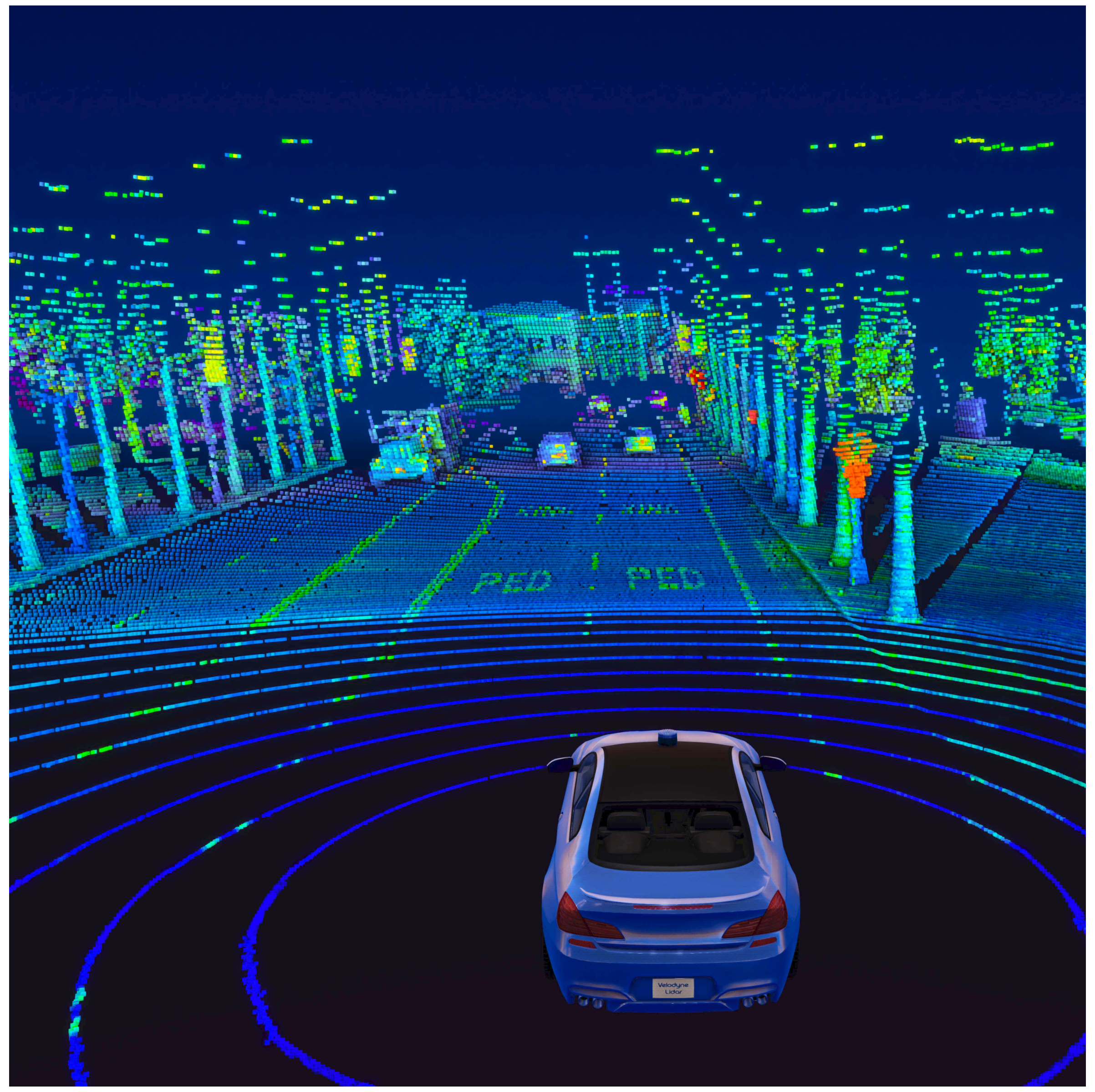
Sequence based model depends on the ordering:

- Shuffle your input list and the prediction changes
- Jets have no hierarchical ordering (not a sentence)

Need to impose a new representation that is permutation equivariant in the latent space

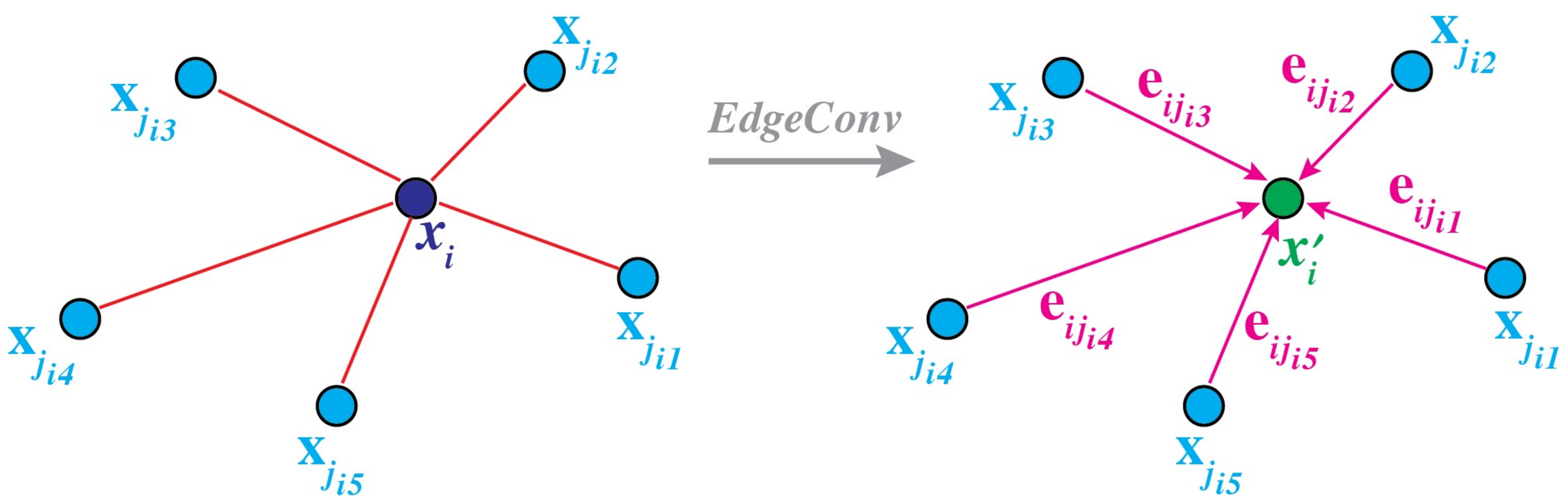


Jet as a Particle Cloud



Analogous representation to the Point Cloud approach in 3D imaging (such as LIDAR):

- Each constituent is an element of a cloud in spatial coordinates
- Cloud is processed and represented as a graph
- Each constituent represent a nodes/'point' in the feature space, we can connect constituents via edges/'lines' representing features of a pair of nodes

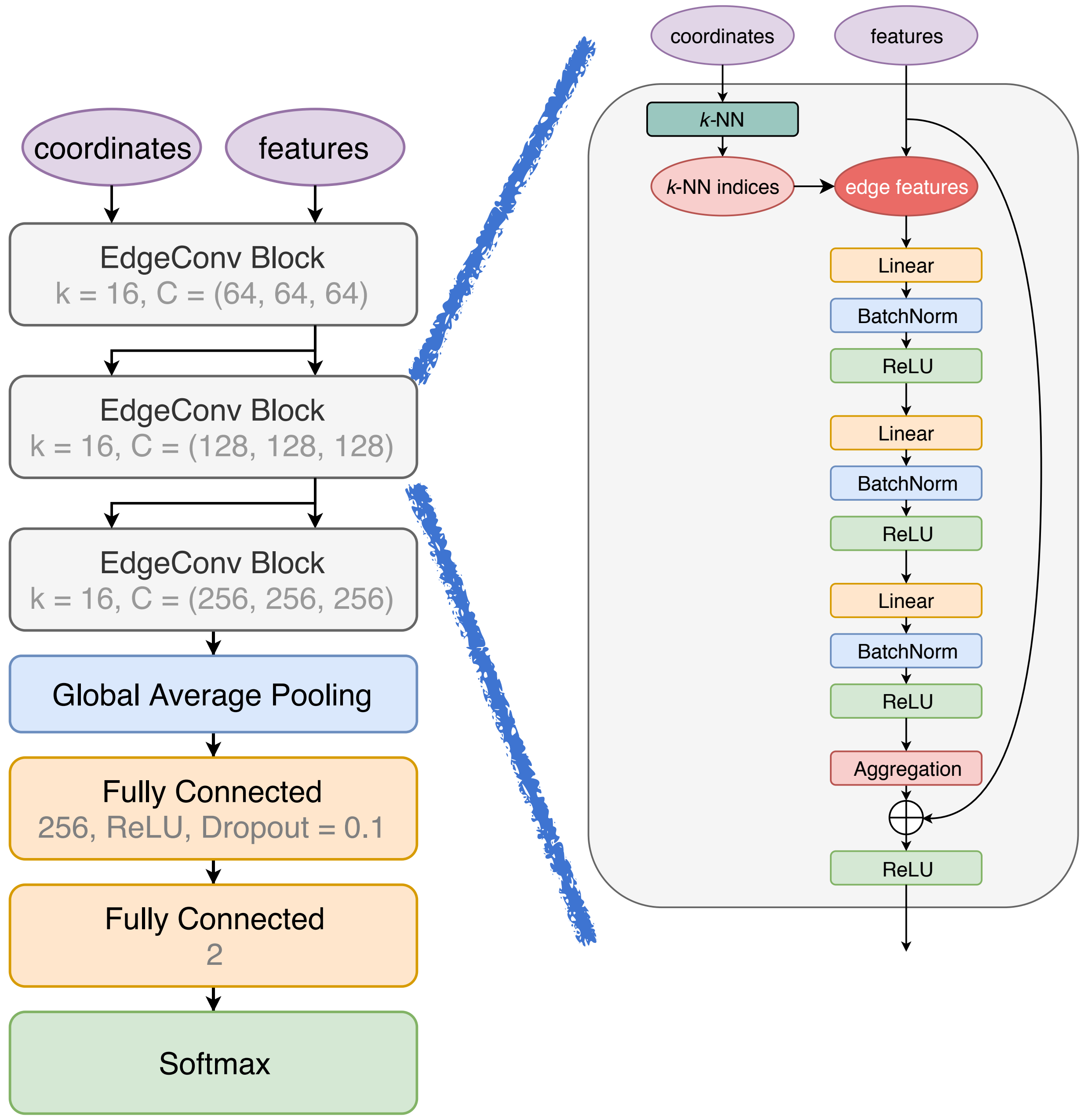


arXiv:1801.07829

ParticleNet

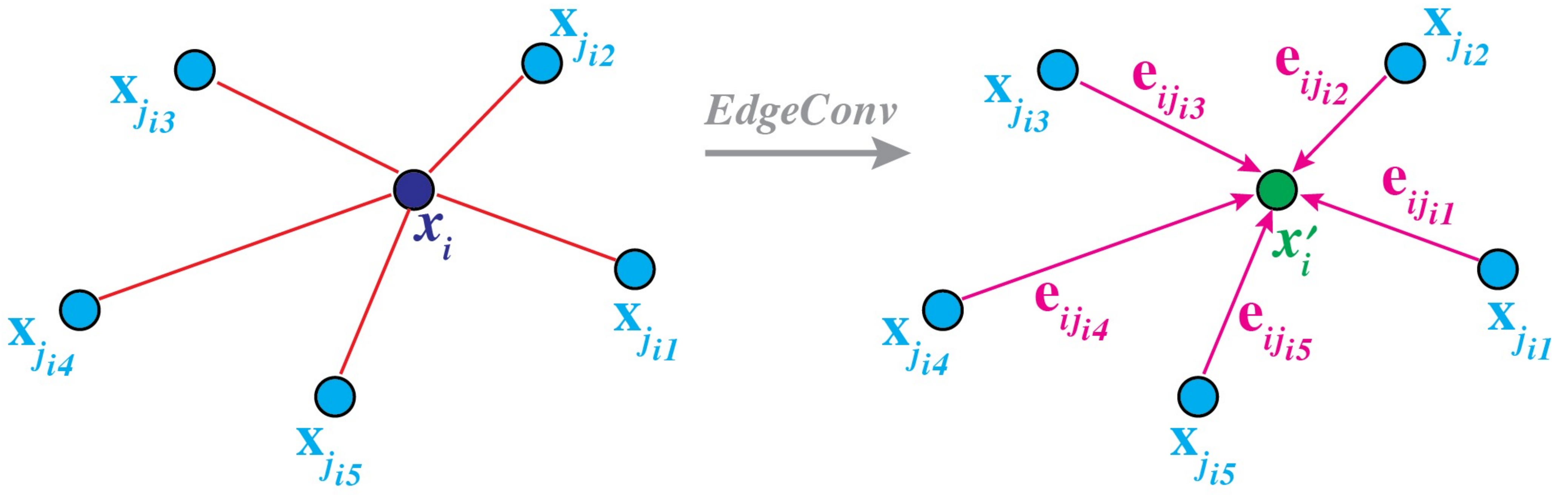
ParticleNet: jet tagging via particle clouds

- Consider the jet from its set of constituents in the $\eta - \phi$ coordinates
- Graph Network structure inherited from the Edge Convolution block
  - Build the edges from the k-nearest neighbors of each nodes
  - Uses permutation equivariant structure; convolutional layers and mean aggregation



	$1/\varepsilon_b$ at $\varepsilon_s = 30\%$
ResNeXt-50	1147 ± 58
P-CNN	759 ± 24
PFN	888 ± 17
ParticleNet-Lite	1262 ± 49
ParticleNet	$\mathbf{1615 \pm 93}$

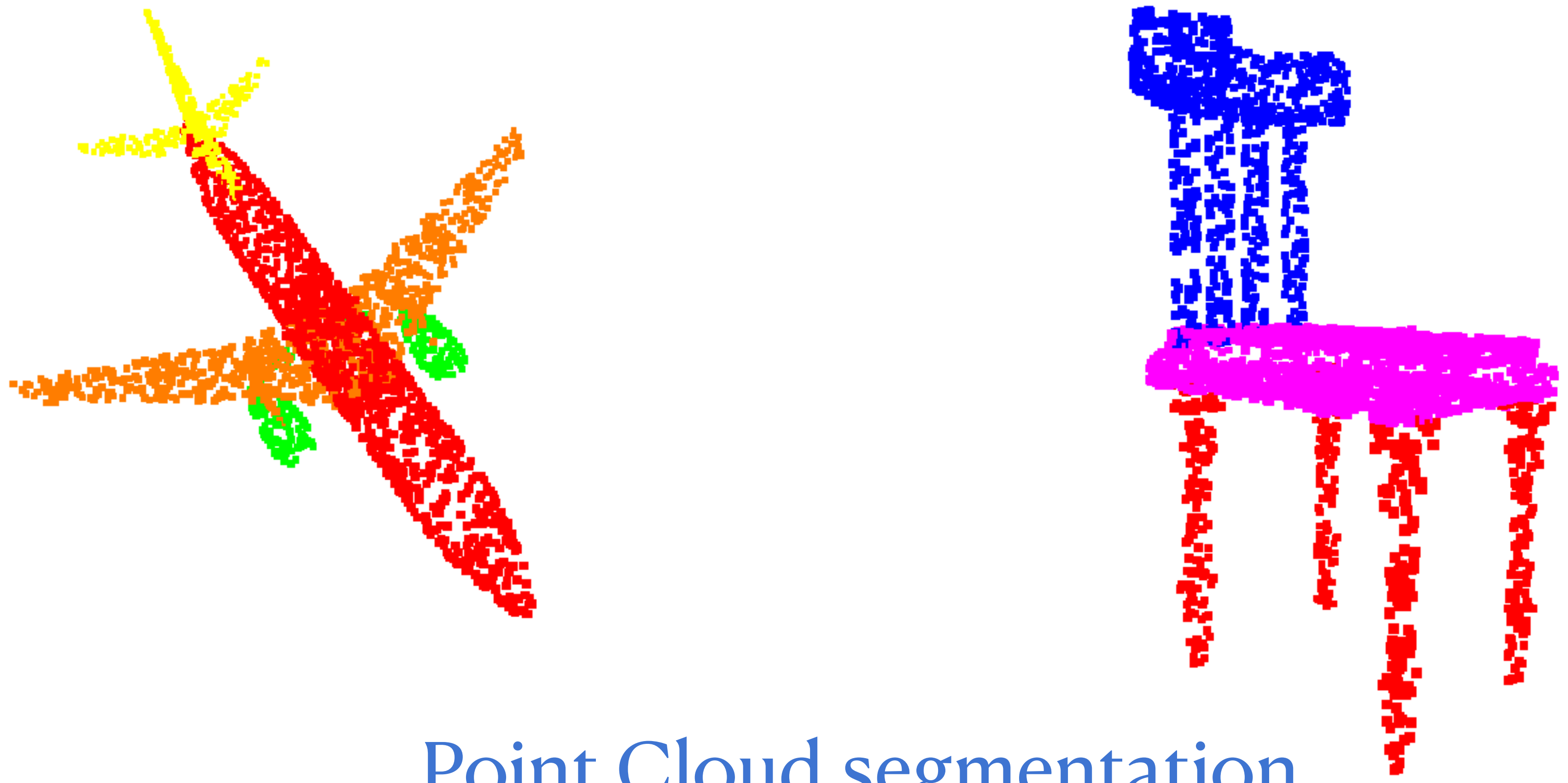
40% better bkg rejection



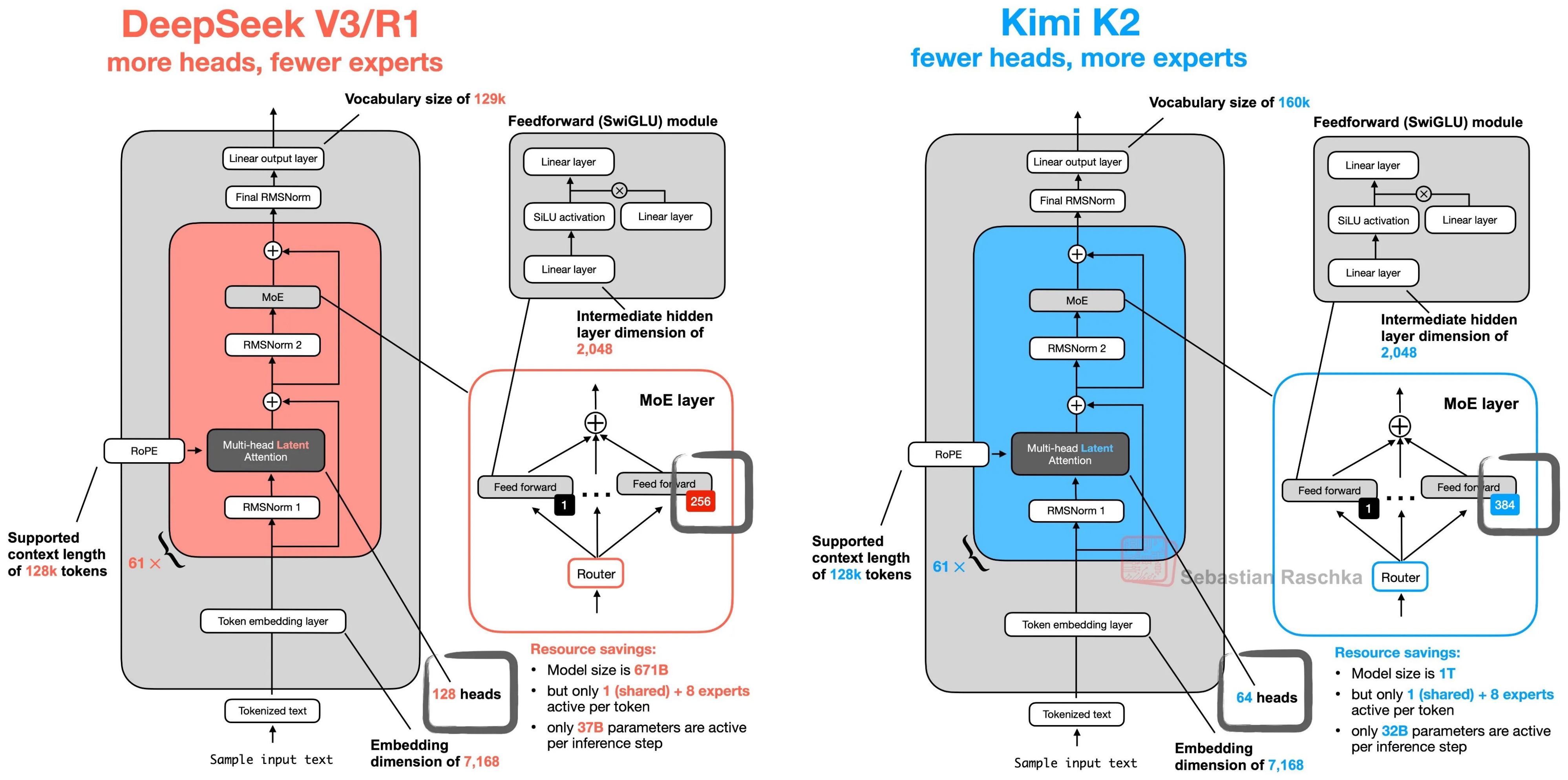
arXiv:1801.07829

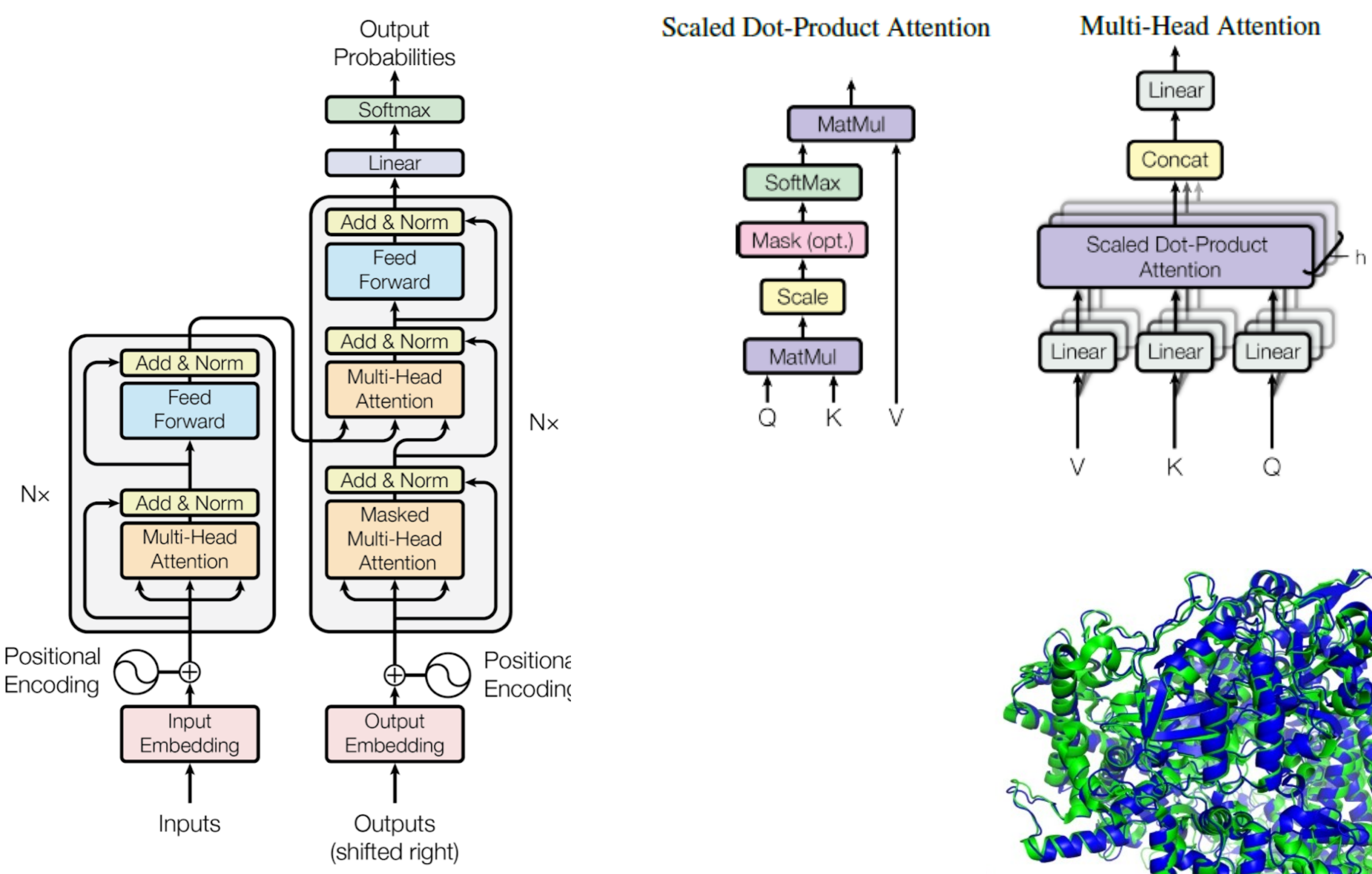
Phys. Rev. D 101, 056019

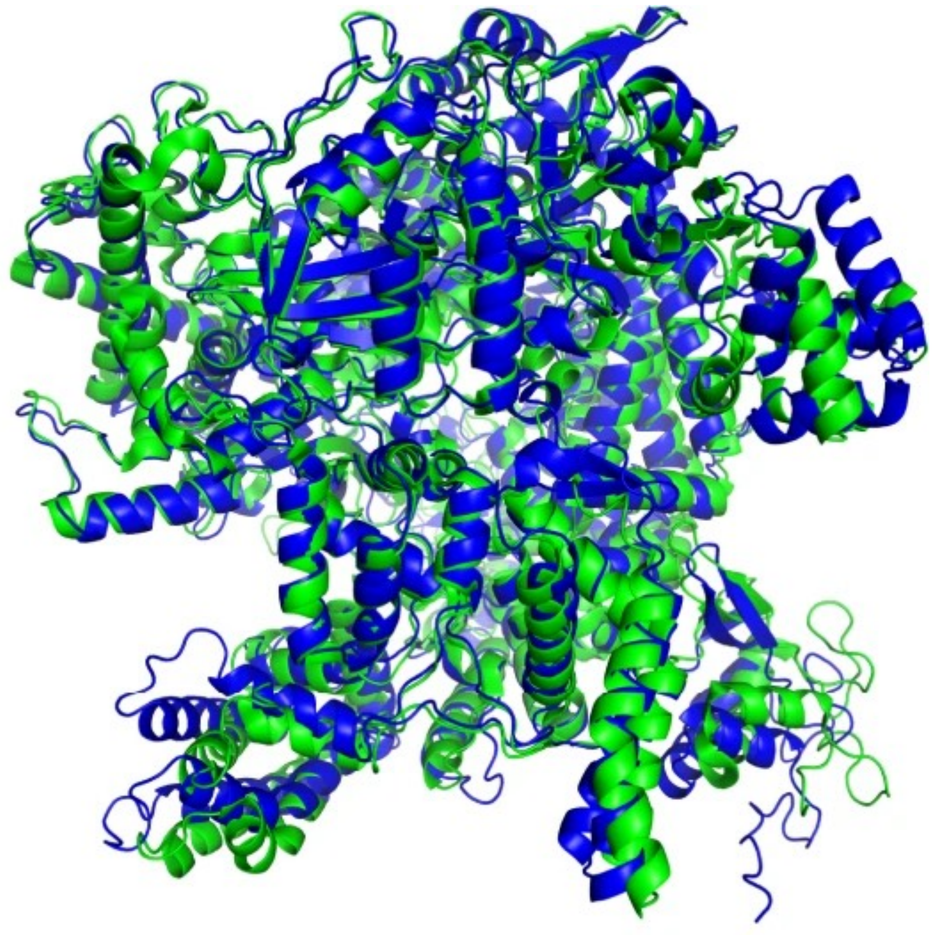
Transformer models



Point Cloud segmentation
(e.g. arXiv:2012.09164)





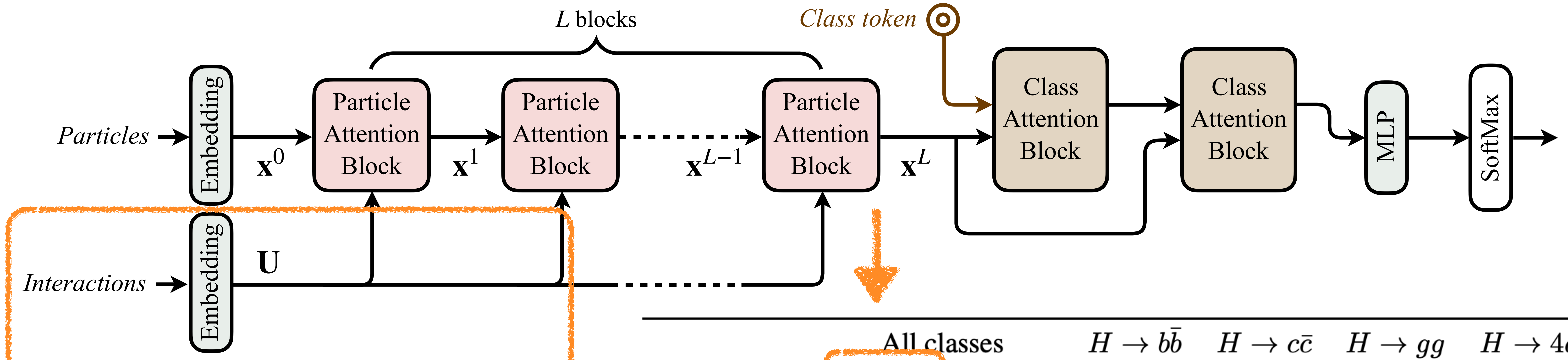


AlphaFold Experiment
r.m.s.d.$_{95}$ = 2.2 Å; TM-score = 0.96

Protein Structure Prediction
(e.g. Nature 596 (2021) 583)

Particle Transformer

HQ, C. Li, S. Qian, ICML 2022



$$\Delta = \sqrt{(y_i - y_j)^2 - (\phi_i - \phi_j)^2}$$

$$k_{\mathrm{T}} = \min(p_{\mathrm{T},i}, p_{\mathrm{T},j})\Delta$$

$$z = \frac{\min(p_{\mathrm{T},i}, p_{\mathrm{T},j})}{p_{\mathrm{T},i}, p_{\mathrm{T},j}}$$

$$m^2 = (E_i + E_j)^2 - |\vec{p_i} + \vec{p_j}|^2$$

Physics inspired pairwise bias (ParT is a Graph Transformer model)

$$P\text{-}MHA(Q, K, V, U) = \mathrm{SoftMax}\left(\frac{QK^T}{\sqrt{d_k}} + U\right) V$$

	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	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{99\%}$	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{99.5\%}$	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{50\%}$
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
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

Better

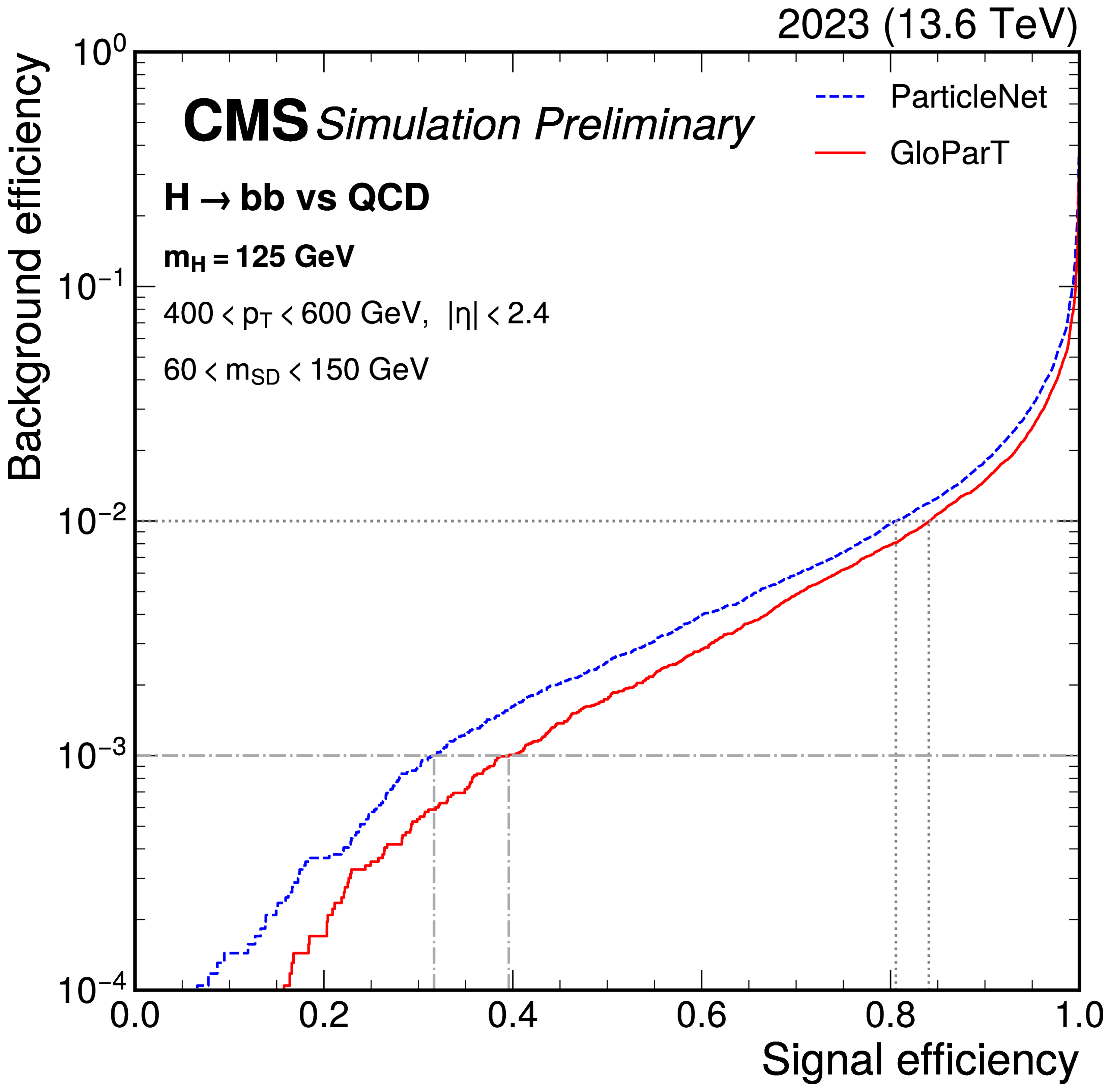
Up to 3x in bkg rejection for an even faster architecture!

	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
ParT	**0.861**	2.14 M	340 M
ParT (plain)	0.849	2.13 M	260 M

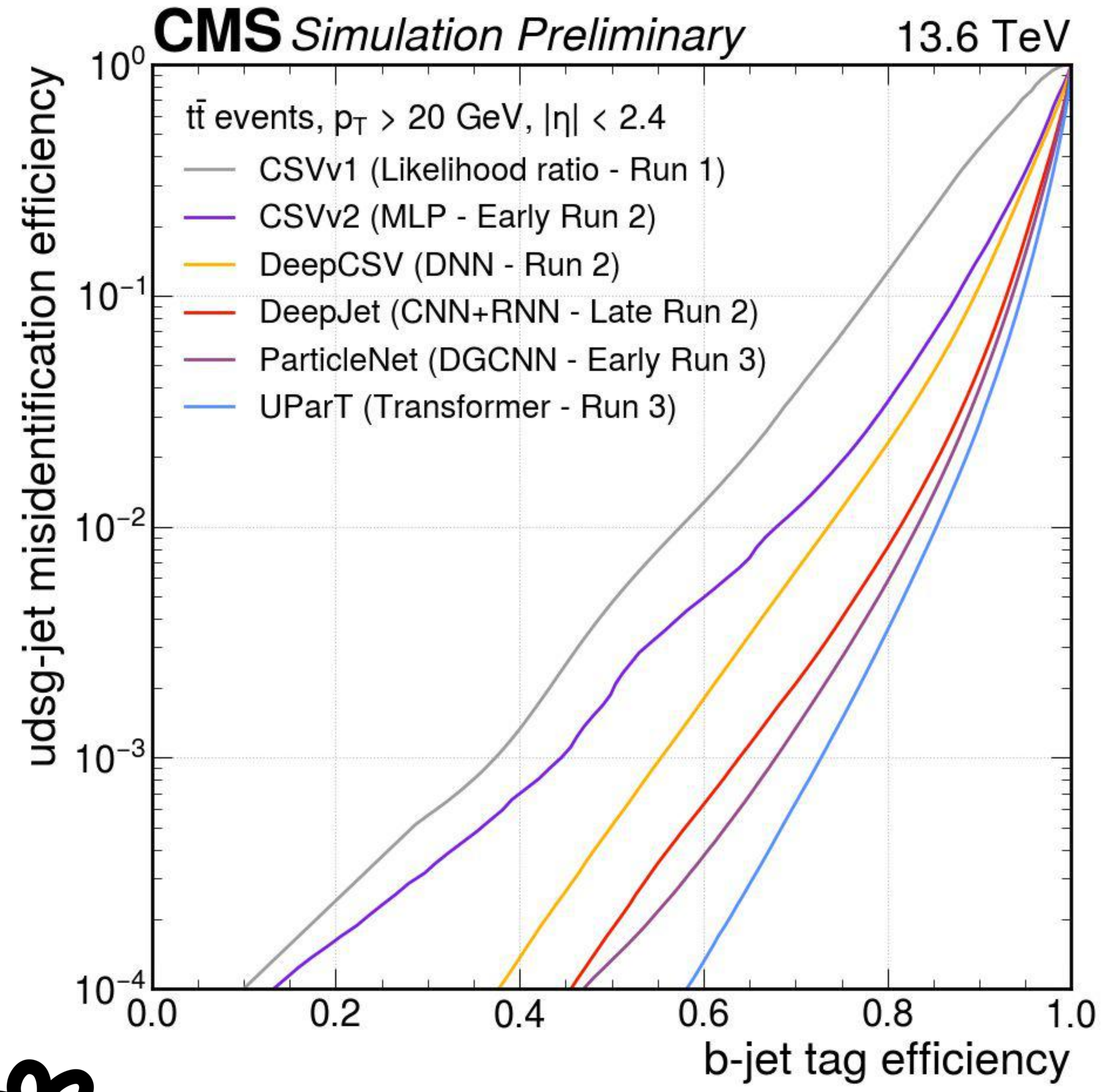
Faster

In CMS: the new state-of-the-art

Large radius tagging



Small radius tagging



Particle Transformer demonstrated state-of-the-art performance.

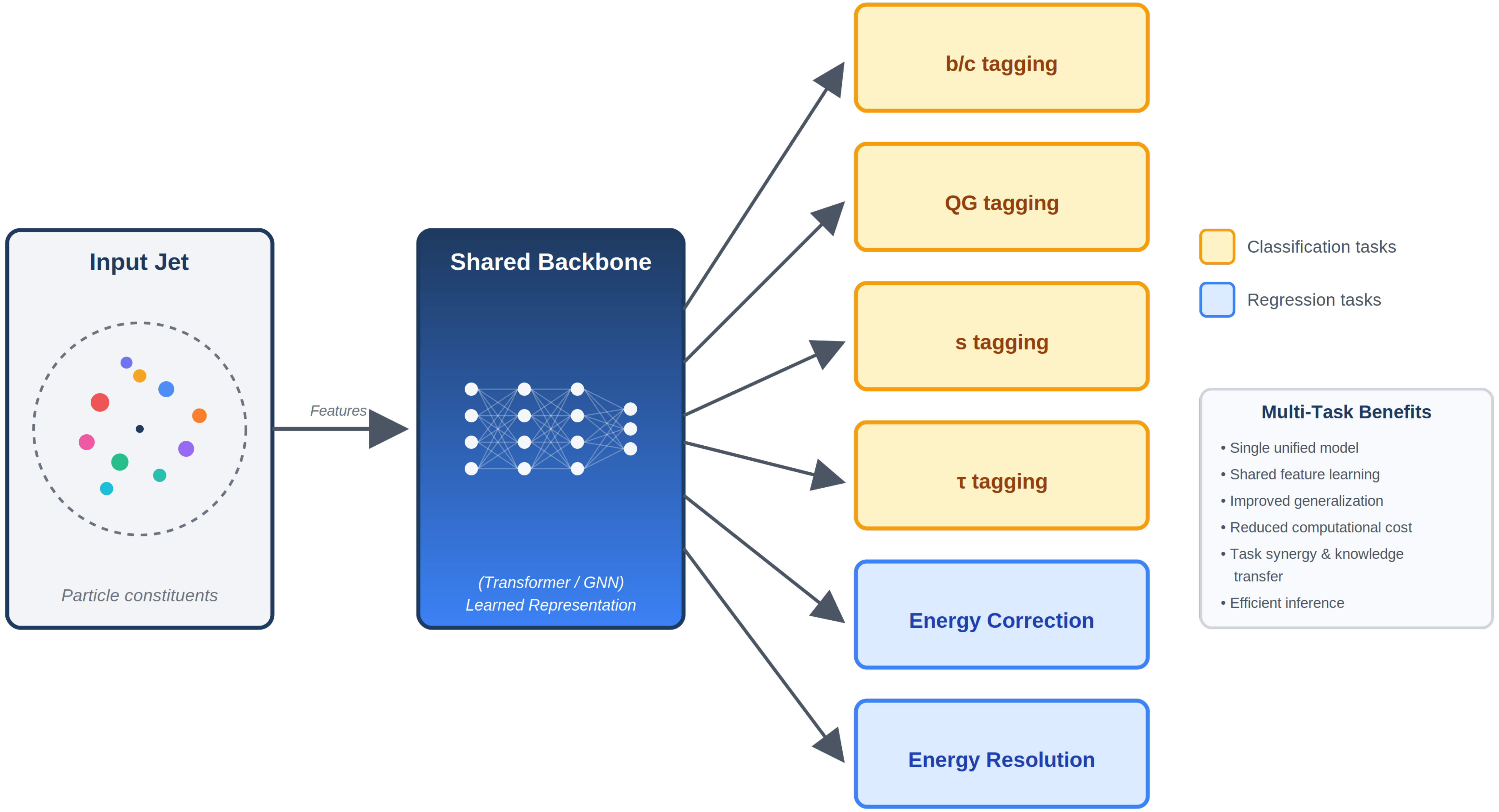
It is now widely adopted as the standard algorithm for jet tasks

CMS-PAS-HIG-24-010

Question: can we beat Transformers ?

CMS DP -2025/081

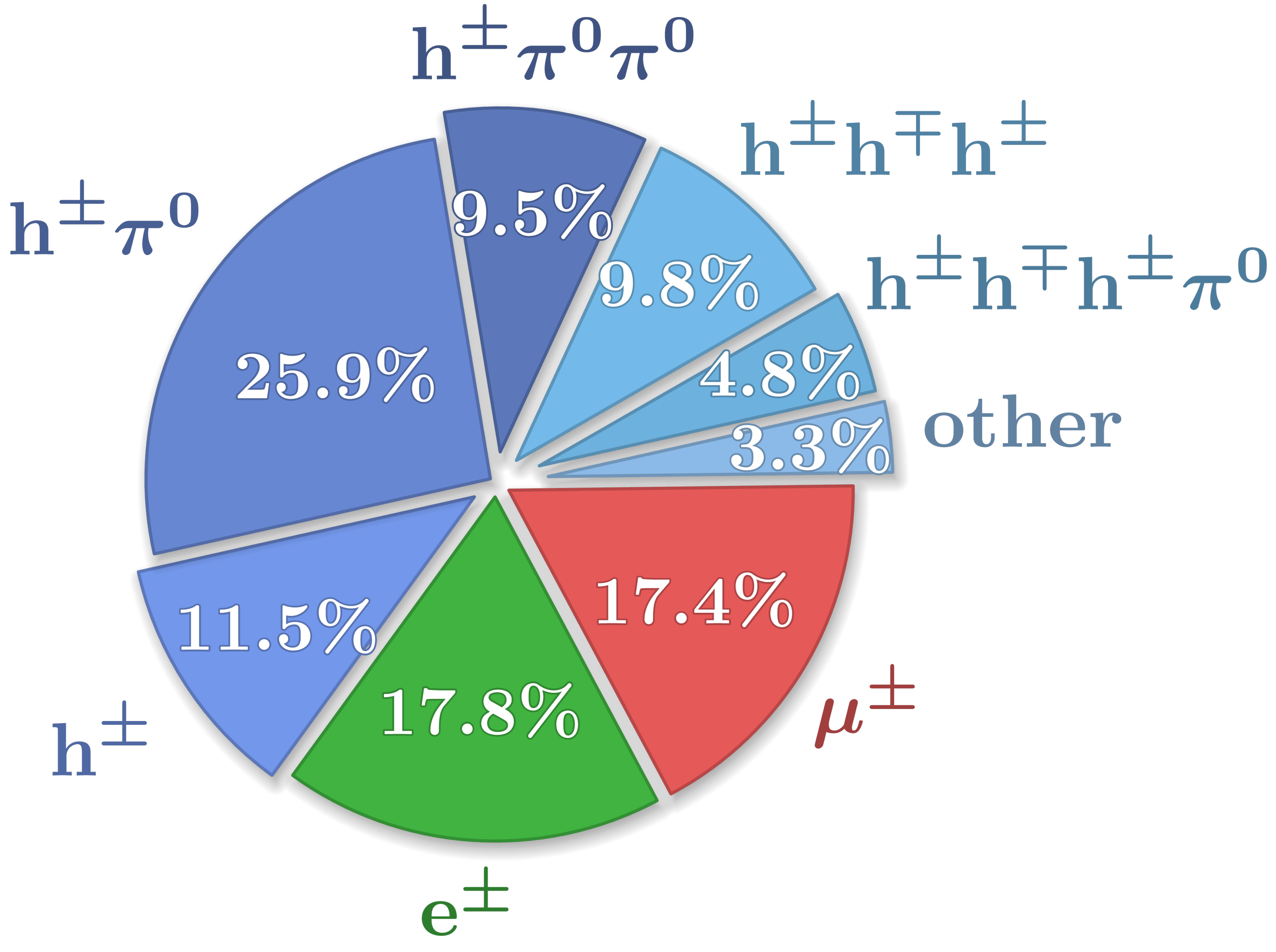
Towards a unified jet approach



Towards a unified jet approach

Goal: combine the latest efforts for developing a unified jet algorithm

- First attempt with ParticleNet
- Extended classification for s-tagging and hadronic tau tagging
- Include a flavor-aware jet energy regression and resolution
- SOTA architecture: Particle Transformer
- New inclusive loss function for this purpose



Tau lepton decay modes

New model: Unified Particle Transformer (UParT)

Towards a unified jet approach

Goal: combine the latest efforts for developing a unified jet algorithm

- First attempt with ParticleNet
- Extended classification for s-tagging and hadronic tau tagging
- Include a flavor-aware jet energy regression and resolution
- SOTA architecture: Particle Transformer
- New inclusive loss function for this purpose

$$L_{cat} = \mathrm{CrossEntropy}(\mathrm{x}, \mathrm{x}_{\mathrm{truth}})$$

+

$$L_{reg} = \lambda \times [\log(\cosh(y^{vis} - y^{vis}_{target}) + \log(\cosh(y^{\nu} - y^{\nu}_{target})]$$

+

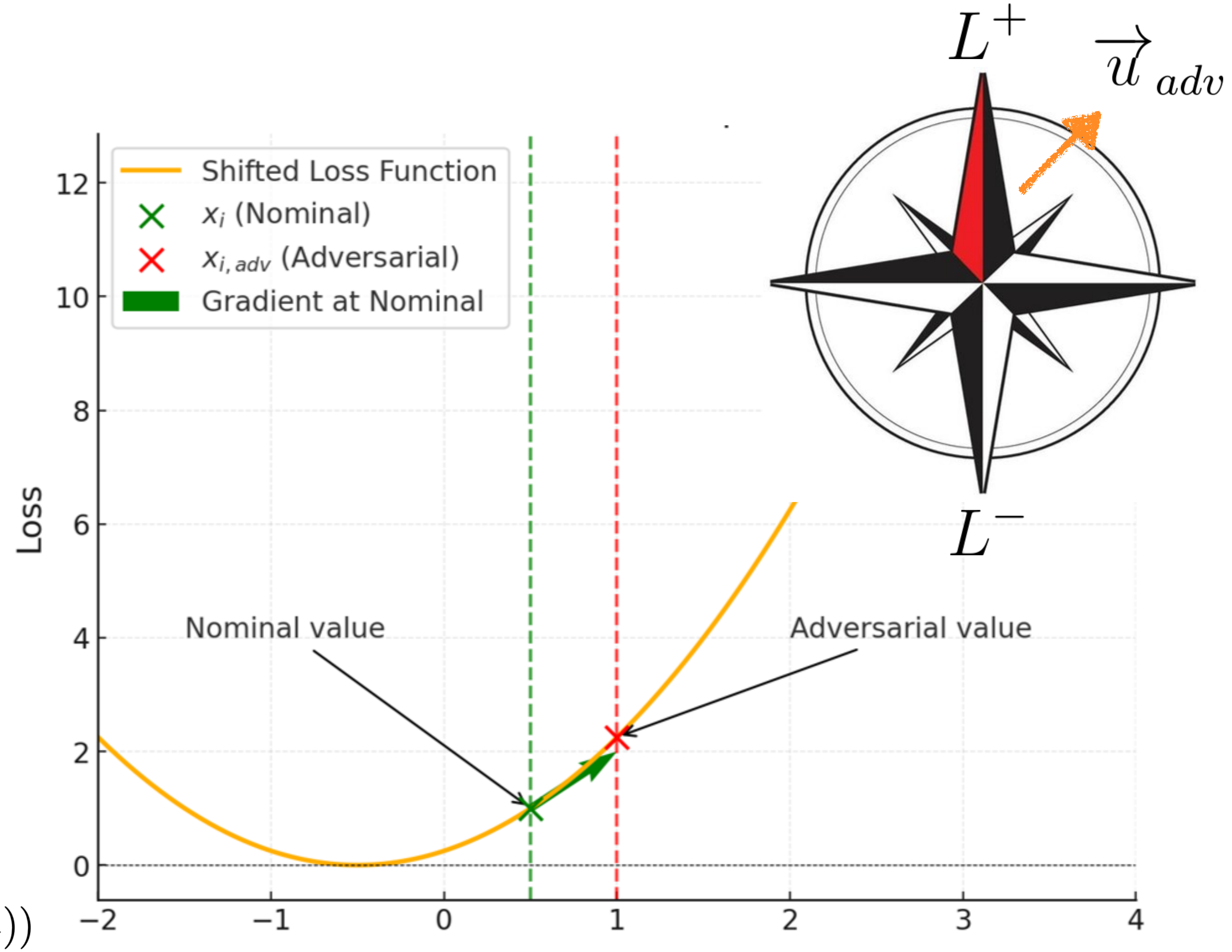
$$L_{res} = \gamma \times [\rho_{0.16}(\mathrm{z}^{\mathrm{vis}} - \mathrm{y}^{\mathrm{vis}}_{\mathrm{target}}) + \rho_{0.84}(\mathrm{k}^{\mathrm{vis}} - \mathrm{y}^{\mathrm{vis}}_{\mathrm{target}})]$$

New model: Unified Particle Transformer (UParT)

New adversarial training

Targeted perturbations to jet input features to fool the network:

- Standard attacks (FGSM & co) are not ideal for heterogeneous jet features.
- Design jet-specific attacks and training.
- Adversarial training: improve robustness to mismodelling in an agnostic way while keeping nominal performance.



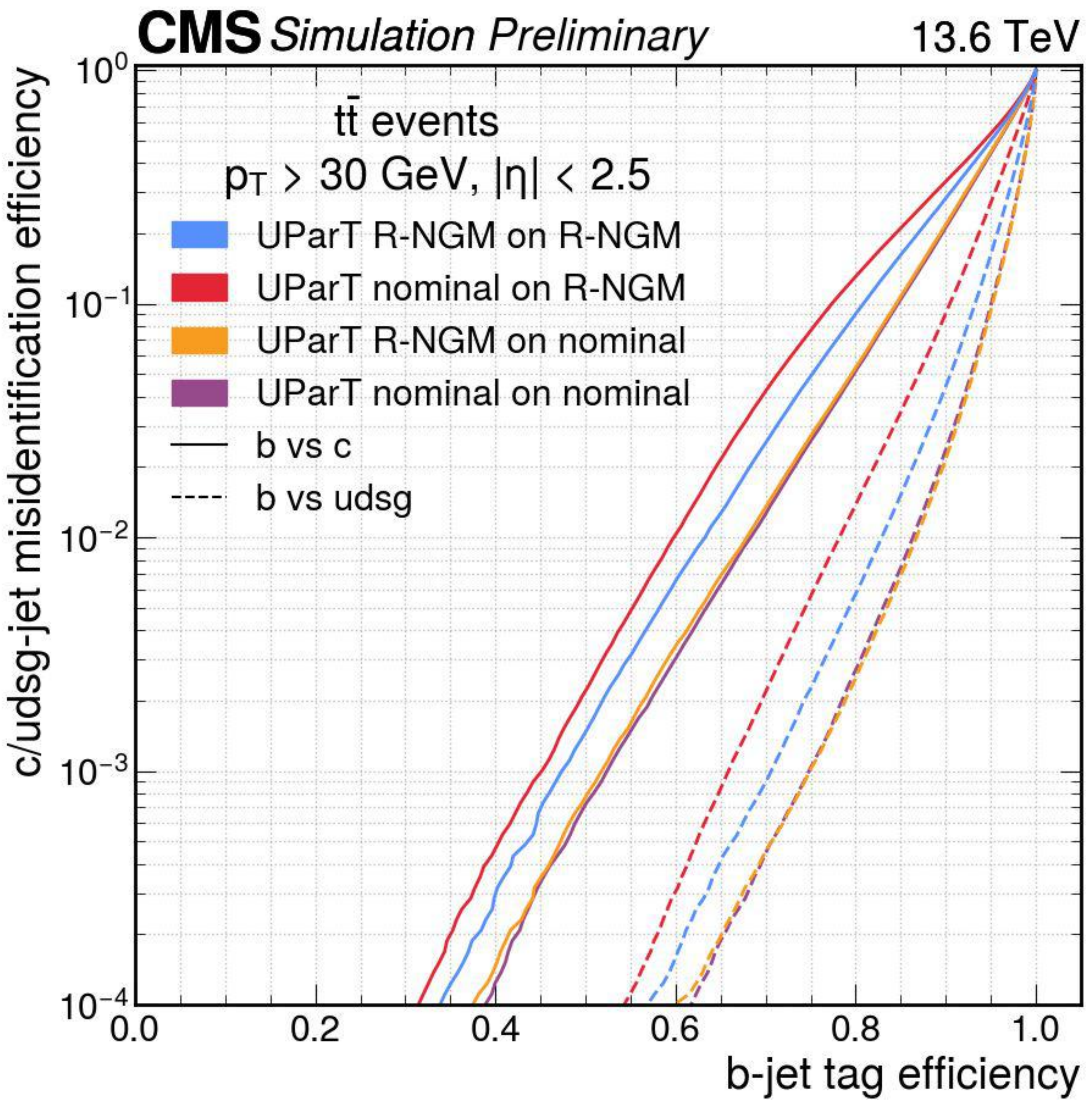
$$L_{\text{adv}}(x, x_{\text{adv}}, y, \theta) = \begin{cases} CE(\theta(x), y) \text{ if nominal mode} \\ CE(\theta(x_{\text{adv}}), y) + \lambda \cdot KL(\theta(x), \theta(x_{\text{adv}})) \\ \quad \text{otherwise} \end{cases}$$

Rectified Normed Gradient Method

R-NGM: uses full input gradient normalized

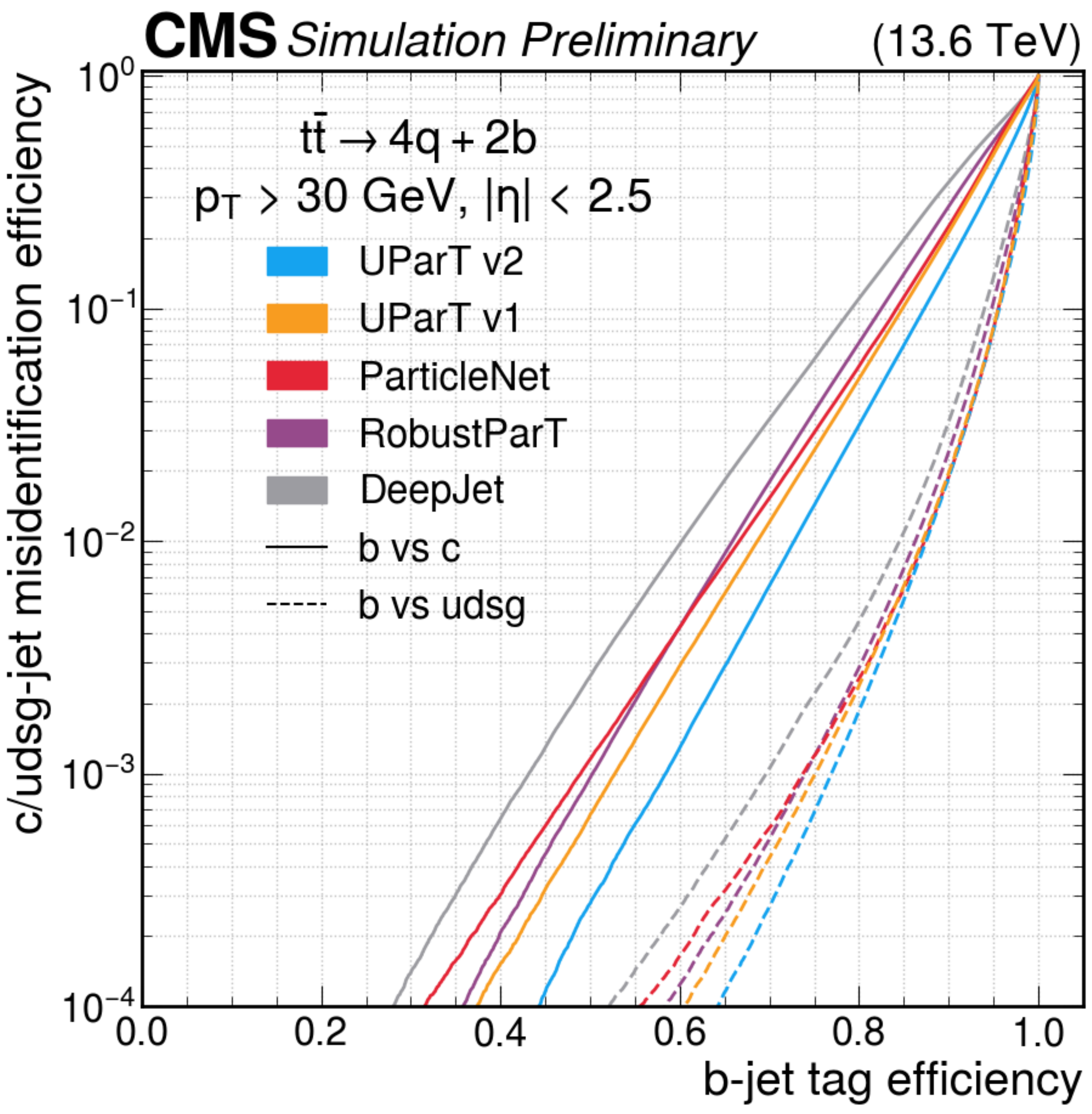
- Adversarial training with these attacks preserves nominal performance
- R-NGM achieves the best robustness
- Result: best MC performance with minimization of the sensitivity to data/MC disagreement

$$x_{\text{adv}} = x + \epsilon \cdot \frac{\nabla_x L(\theta, x, y)}{||\nabla_x L(\theta, x, y)||_{L_2}}$$



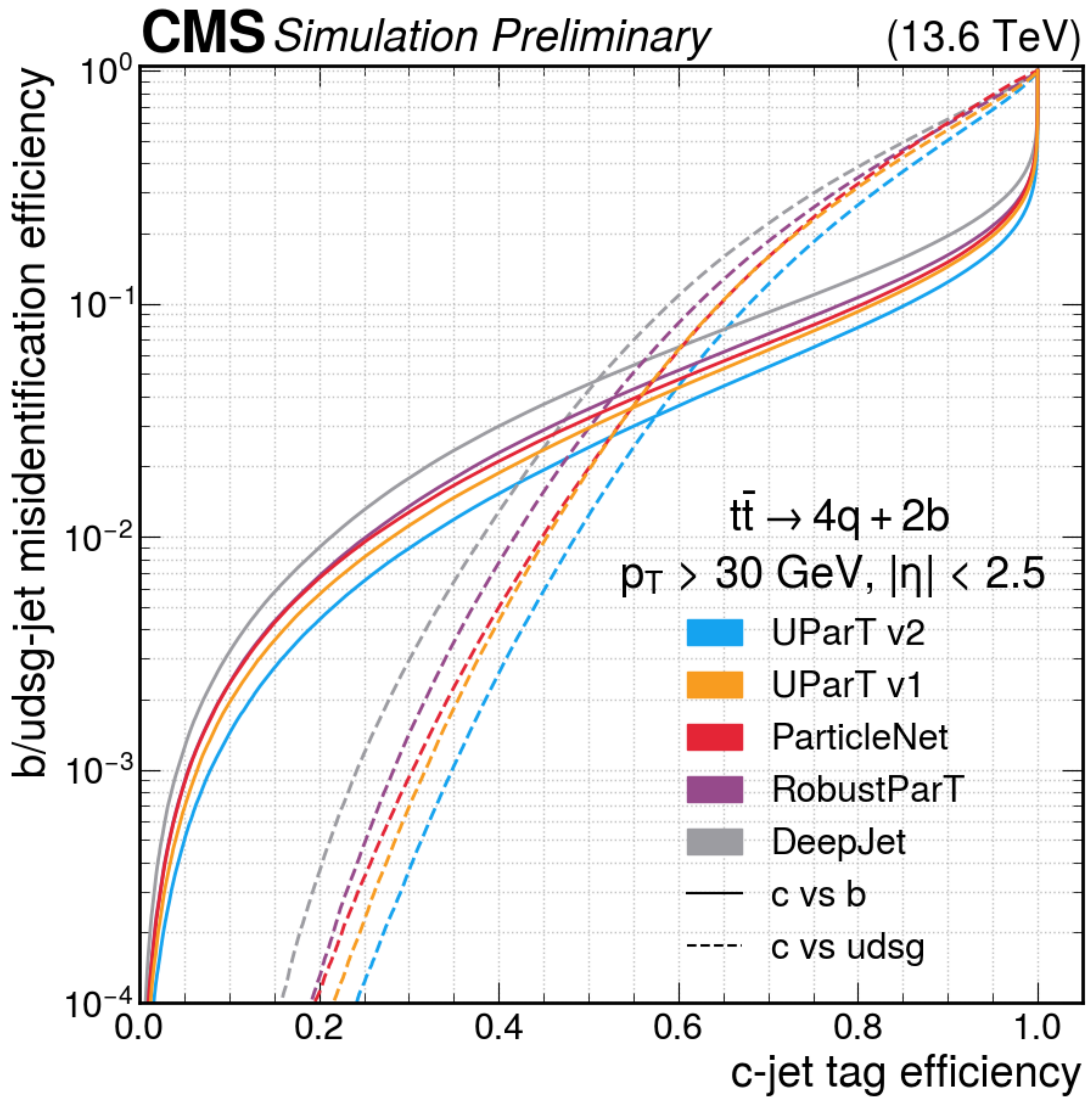
CMS DP -2025/081

UParT: flavor tagging result



b-tagging

About 20-60% improvement in bkg rejection for b/c tagging compared to the ParticleNet models



c-tagging

CMS DP -2025/081

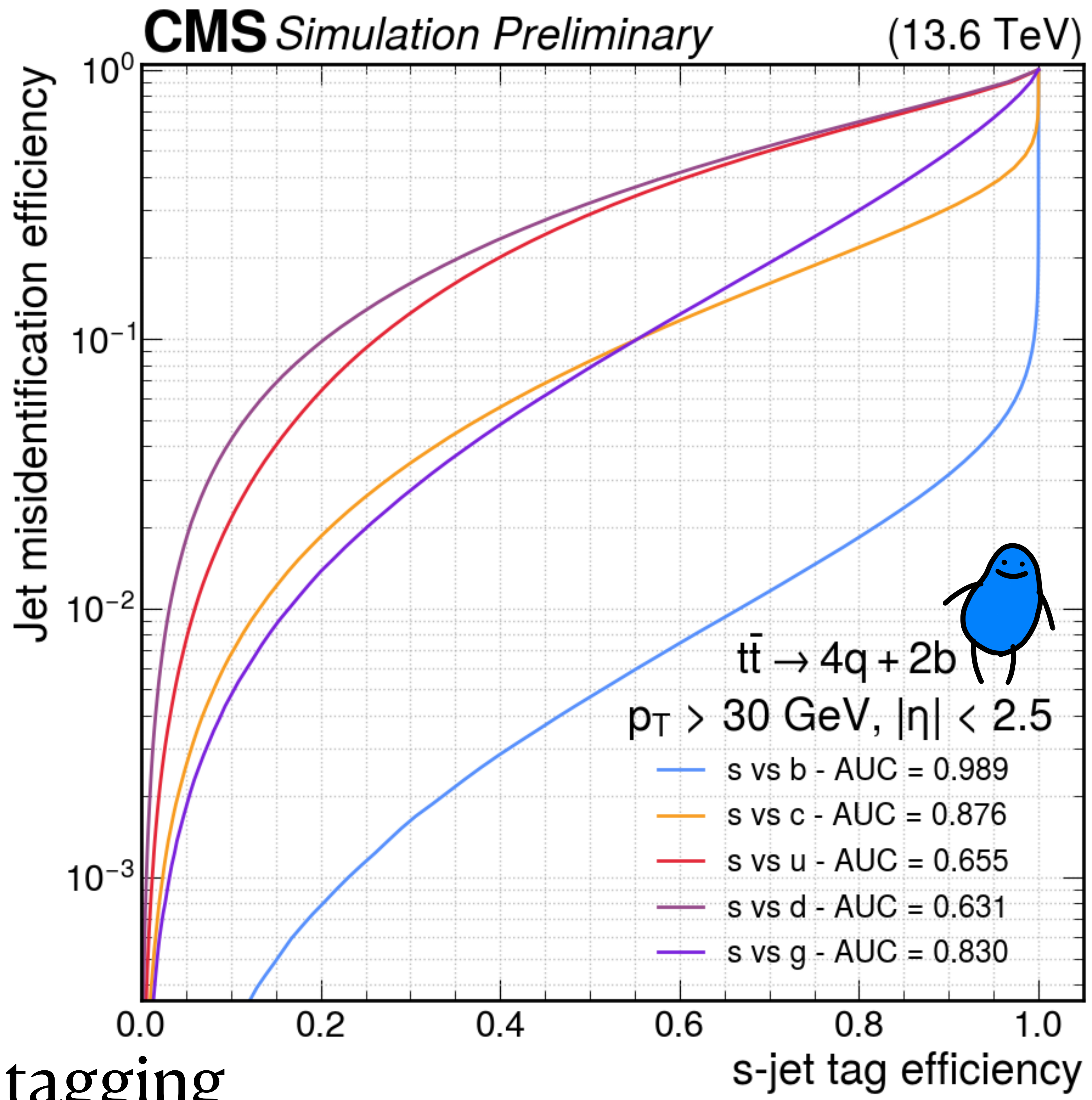
UParT: flavor tagging result

CMS DP -2025/081

Very first s-tagging model at LHC. We can achieve a low efficiency s-tagger.

Future: work on the calibration of the s-tagging node

Question: Can you do u vs d tagging ?



s-tagging

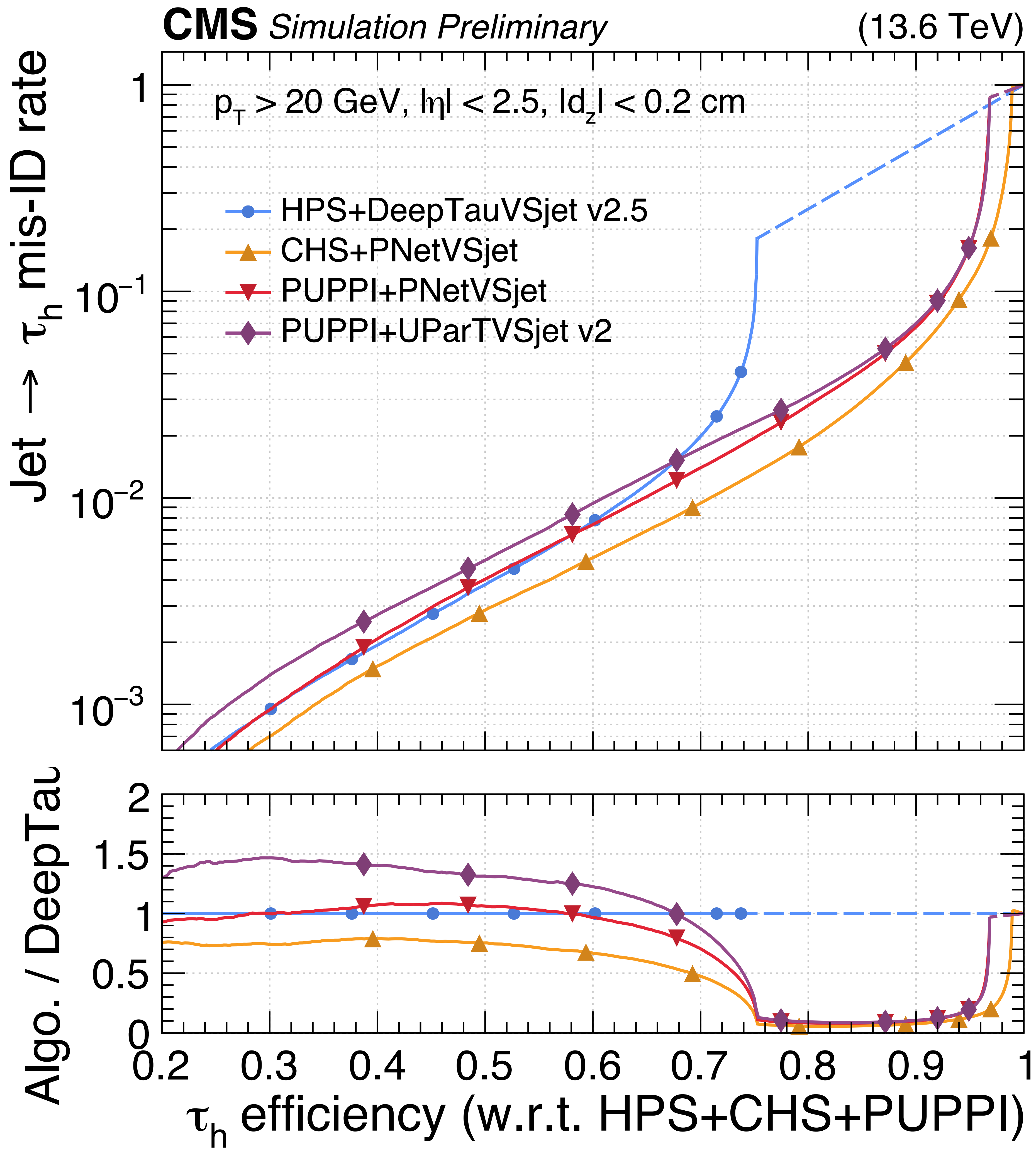
UParT: flavor tagging result

CMS DP -2025/073

Tau performance not fully under control

High dependence of the pile-up mitigation algorithm plus the reconstruction of the tau

Future: include reconstruction task in the jet algorithm

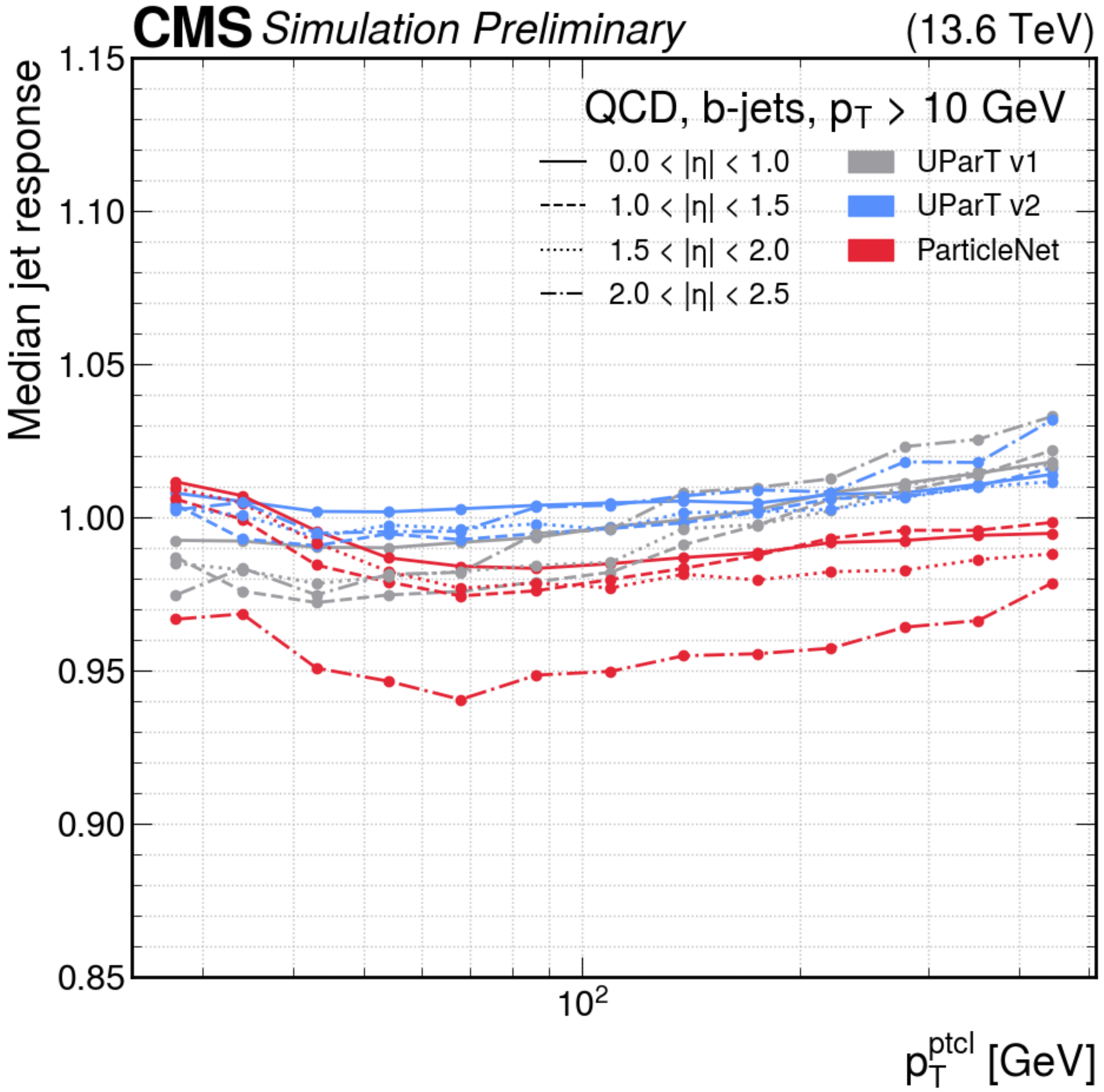


UParT: jet regression result

CMS DP -2025/081

Improved median jet response across model versions.

Observed per year overall sensitivity (small HCAL response changes can dramatically impact the models)

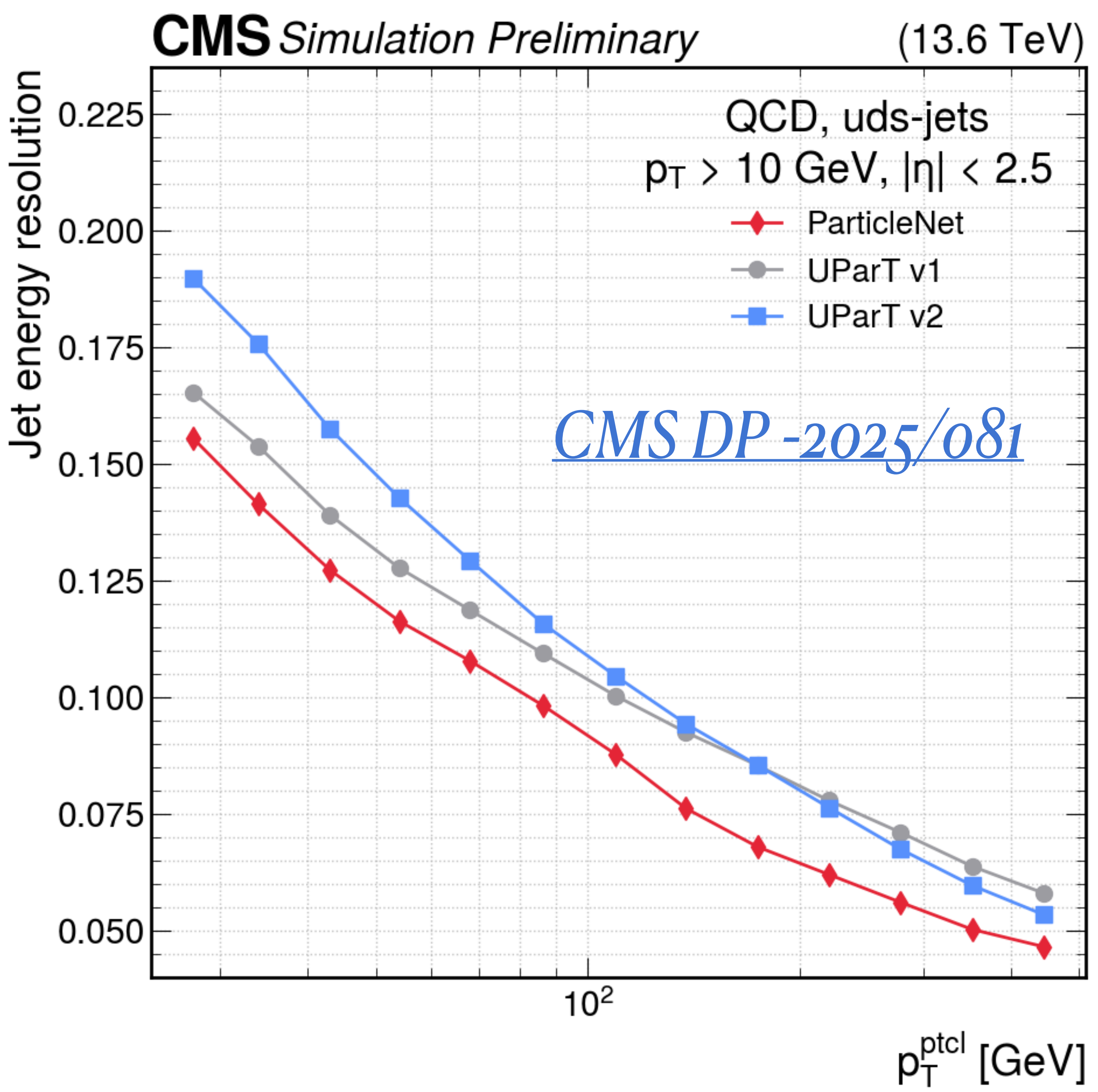


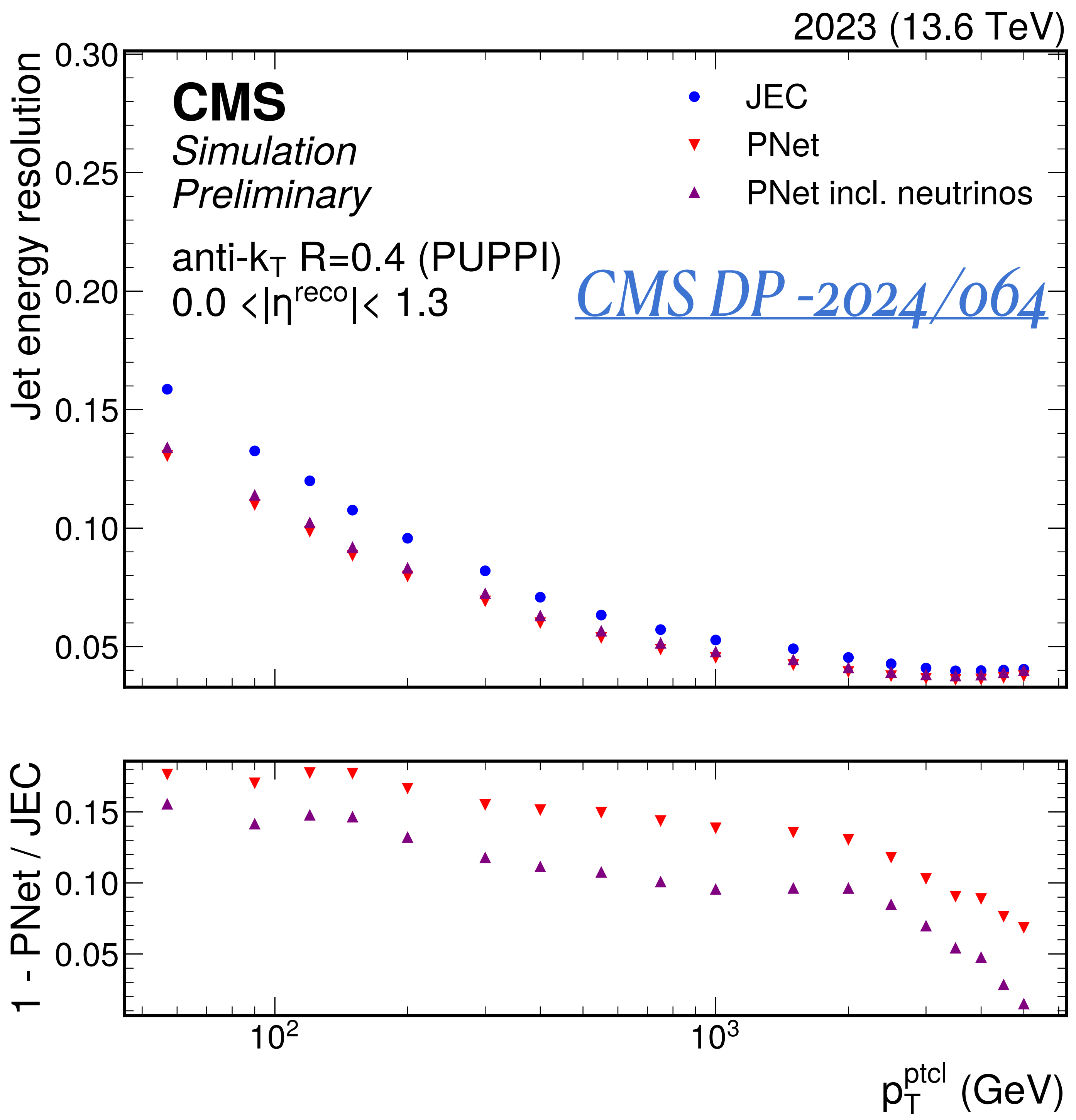
UParT: jet resolution result

Resolution improved compared to the usual JEC by O(10-15%)

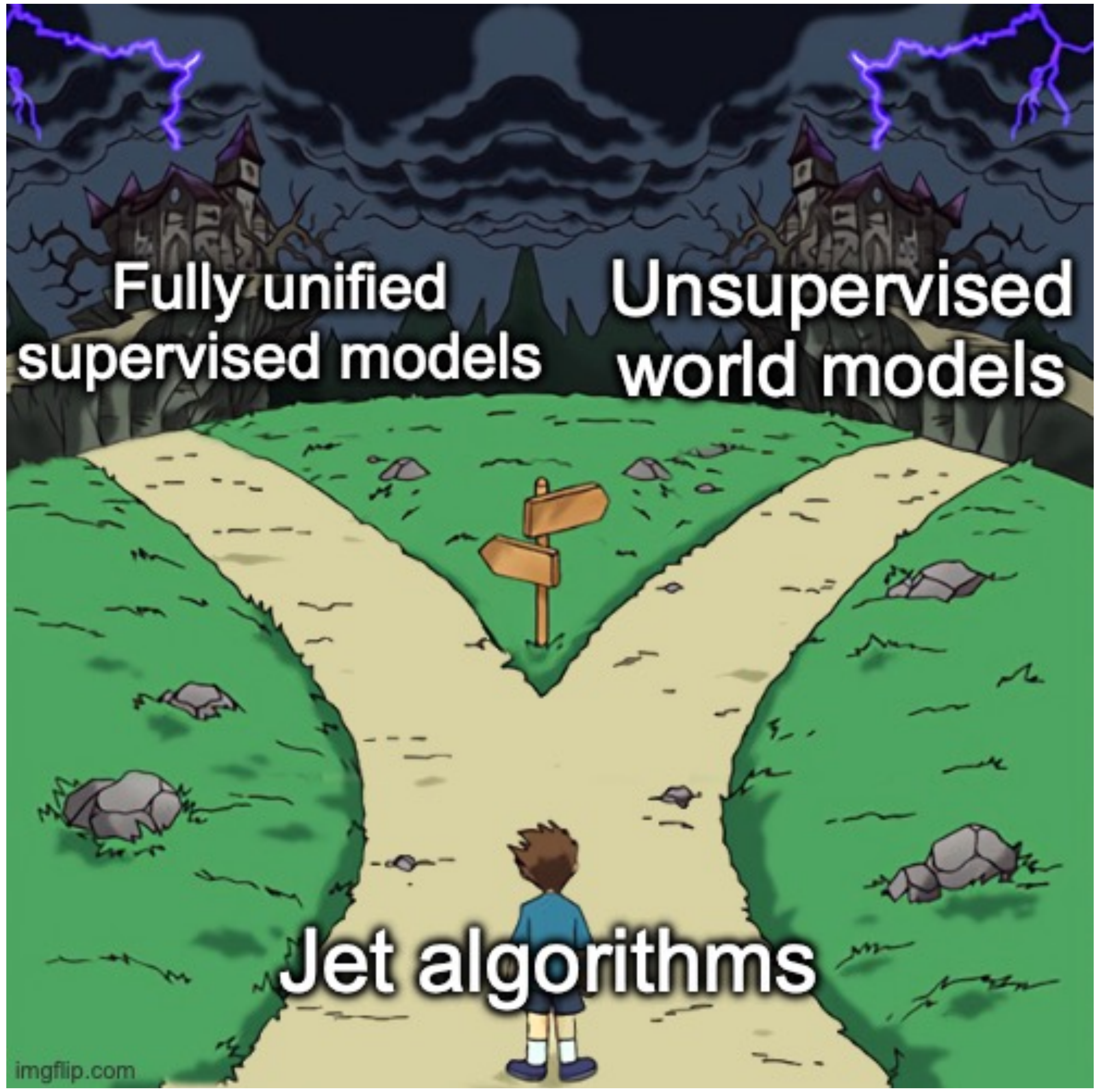
Degradation in UPArT models: problem seems to be now fully understood

- Simulation and training tuning issue





Towards fully unified world models



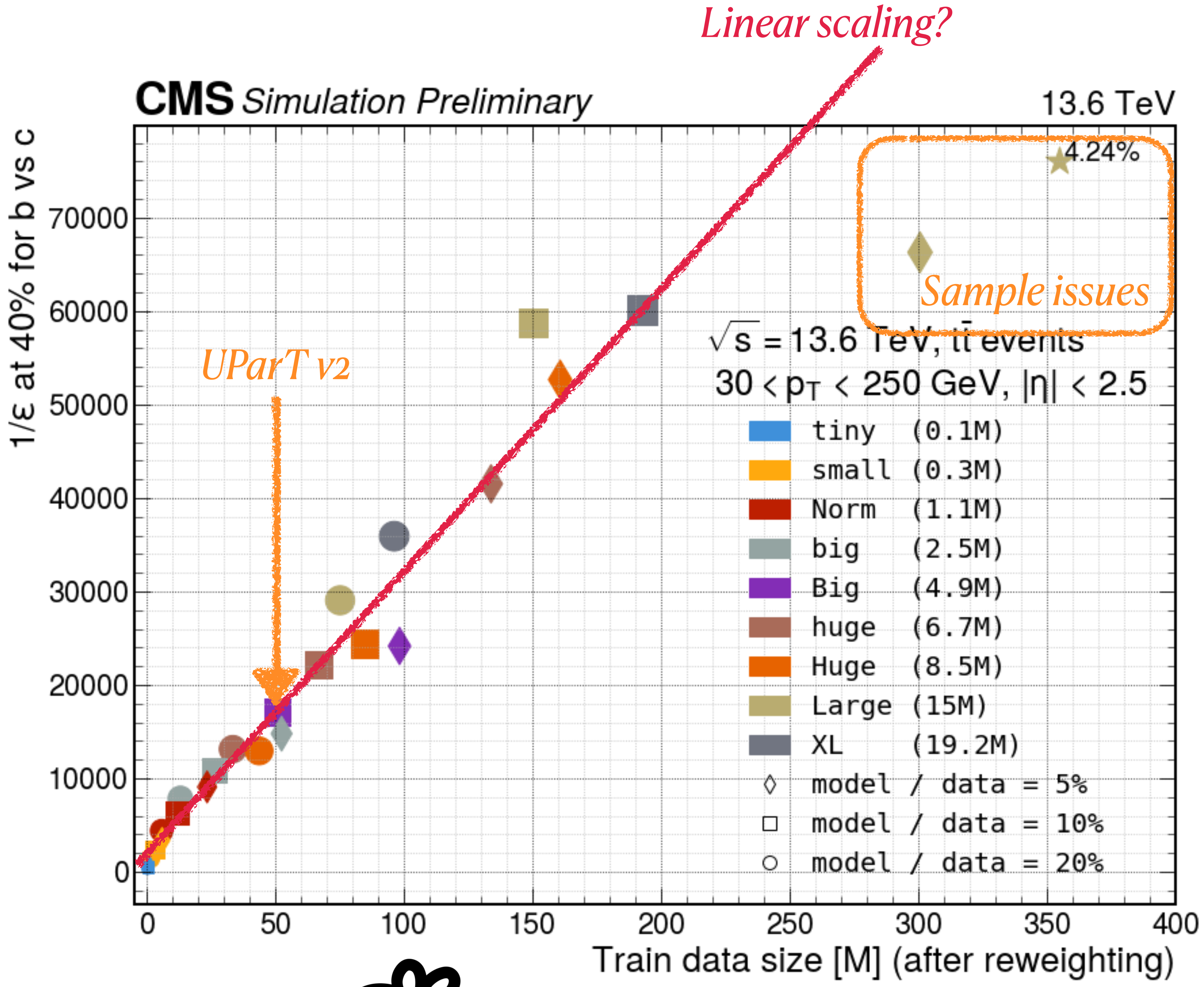
Prelude: scaling laws

[Pavlo Kashko's talk]

Scaling laws: the laws extrapolating the performance of neural networks at larger size (model/datasets)

We have found scaling laws of flavor tagging:

- Huge room of improvement
- Challenges: fit the inference in our software and stabilize the training
- To come: even better scaling laws

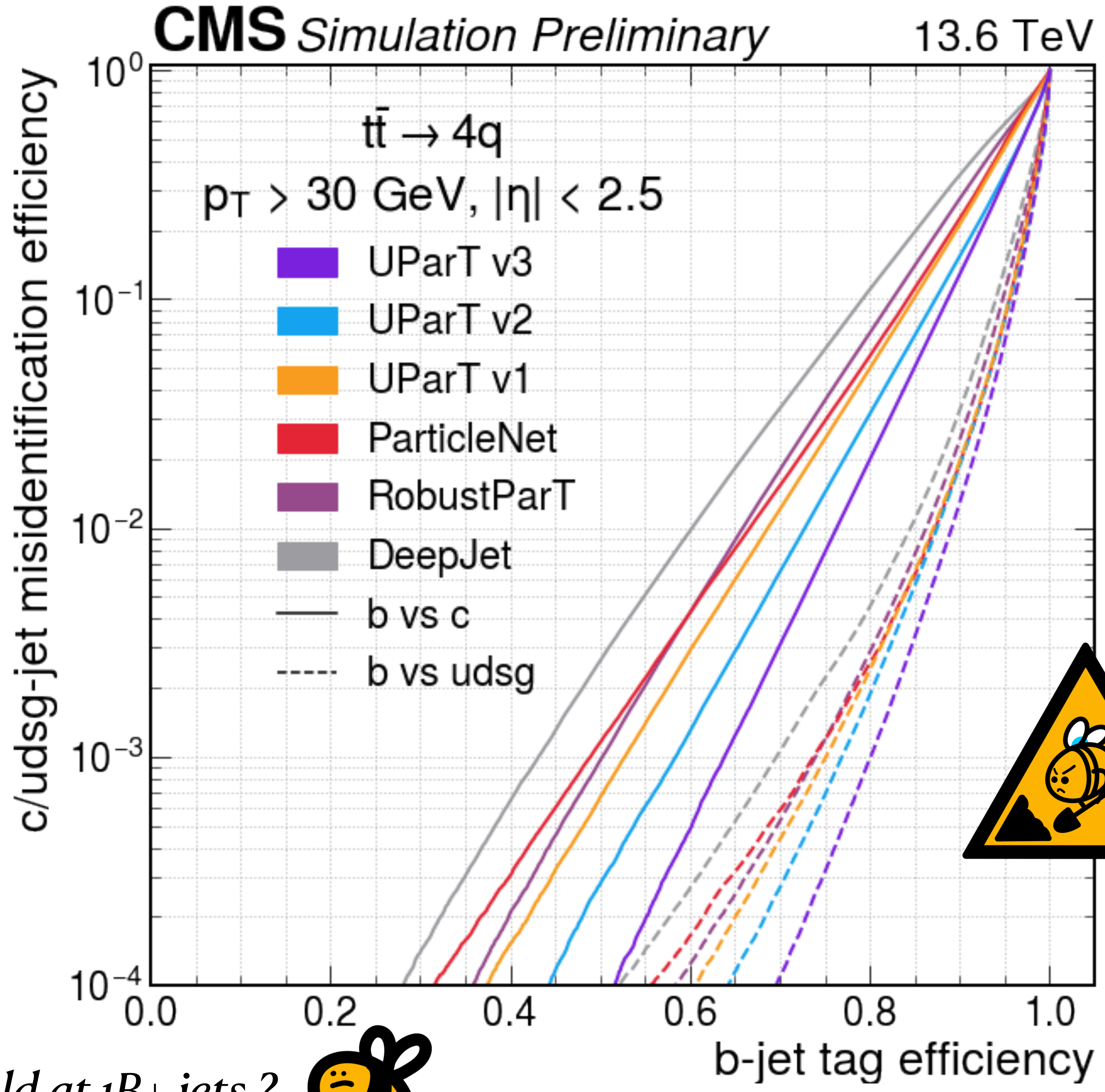


Question: Does it hold at 1B+ jets ?

Prelude: scaling law

We have extrapolated the scaling laws of flavor tagging:

- Huge room of improvement
- Challenges: fit the inference in our software and stabilize the training
- To come: even better scaling laws



W.I.P.

Warning: this is a first attempt!

Question: Does it hold at 1B+ jets ?

Full supervised unification

We could extend the unification towards more tasks:

- Full object reconstructions such as hadronic tau or b/c hadrons
- Jet charge tagging
- In-training ML based calibration

Recent VUB+Vanderbilt effort

gen τ_h - HPS matching efficiency

Reco/Gen	TAU1H0P (16.6%)	TAU1H1P (40.9%)	TAU1H2P (15.9%)	TAU3H0P (17.7%)	TAU3H1P (8.5%)	TAUother (0.3%)
0h_0p	14.54	22.53	21.23	9.65	11.24	20.54
0h_1p	0.00	0.30	0.32	0.00	0.02	0.33
0h_2p	0.00	0.00	0.26	0.00	0.00	0.50
0h_3p	0.00	0.00	0.00	0.00	0.00	0.41
1h_0p	85.46	13.07	7.65	4.40	1.15	5.69
1h_1p		64.10	28.85	0.00	4.60	12.46
1h_2p			41.68	0.00	0.00	15.68
1h_3p				0.00	0.00	18.07
2h_0p				19.62	9.16	1.73
2h_1p				0.00	7.95	1.24
2h_2p				0.00	0.00	1.24
2h_3p				0.00	0.00	0.00
3h_0p				66.33	25.49	8.58
3h_1p					40.40	4.21
3h_2p						8.33
3h_3p						0.00
other						0.99

Old method (cut based - low efficiency and high fake rates)

gen τ_h- jet matching efficiency

Reco/Gen	TAU1H0P (16.6%)	TAU1H1P (40.9%)	TAU1H2P (15.9%)	TAU3H0P (17.7%)	TAU3H1P (8.5%)	TAUother (0.3%)
0h_0p	1.33	0.04	0.03	0.07	0.01	0.08
0h_1p	0.00	8.49	0.40	0.00	0.29	0.08
0h_2p	0.00	0.00	7.64	0.00	0.00	0.41
0h_3p	0.00	0.00	0.00	0.00	0.00	3.88
1h_0p	98.67	8.14	2.01	1.28	0.10	0.66
1h_1p		83.33	12.62	0.00	1.61	4.13
1h_2p			77.30	0.00	0.00	8.83
1h_3p				0.00	0.00	45.30
2h_0p				12.58	1.41	0.33
2h_1p				0.00	11.07	0.58
2h_2p				0.00	0.00	3.05
2h_3p				0.00	0.00	0.00
3h_0p				86.07	13.82	1.40
3h_1p					71.68	3.63
3h_2p						18.23
3h_3p						0.00
other						9.41

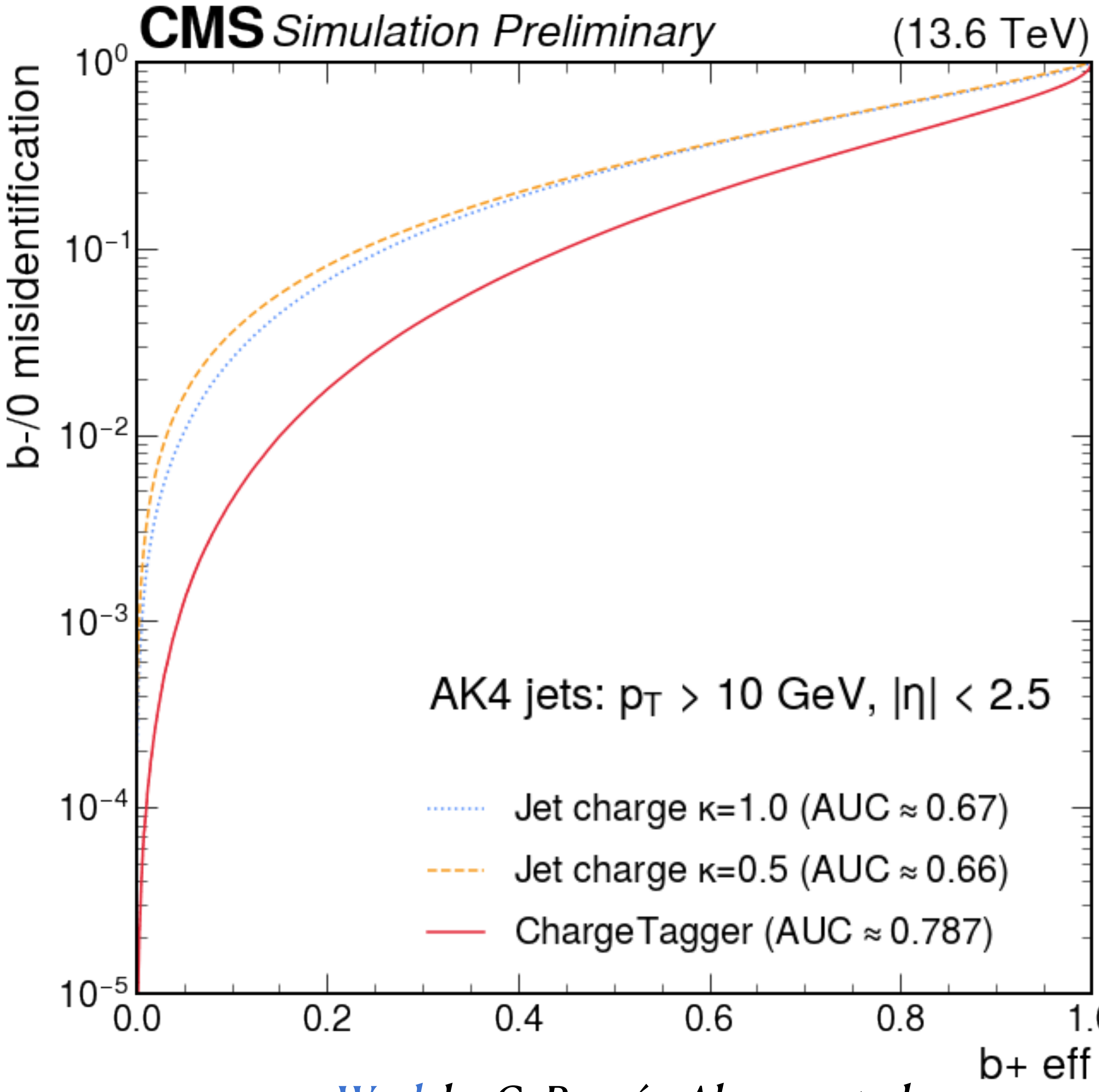
New study (huge room of improvement - gen matching only)

Full supervised unification

We could extend the unification towards more tasks:

- Full object reconstructions such as hadronic tau or b/c hadrons
- Jet charge tagging
- In-training ML based calibration

CMS-DP-2025-071



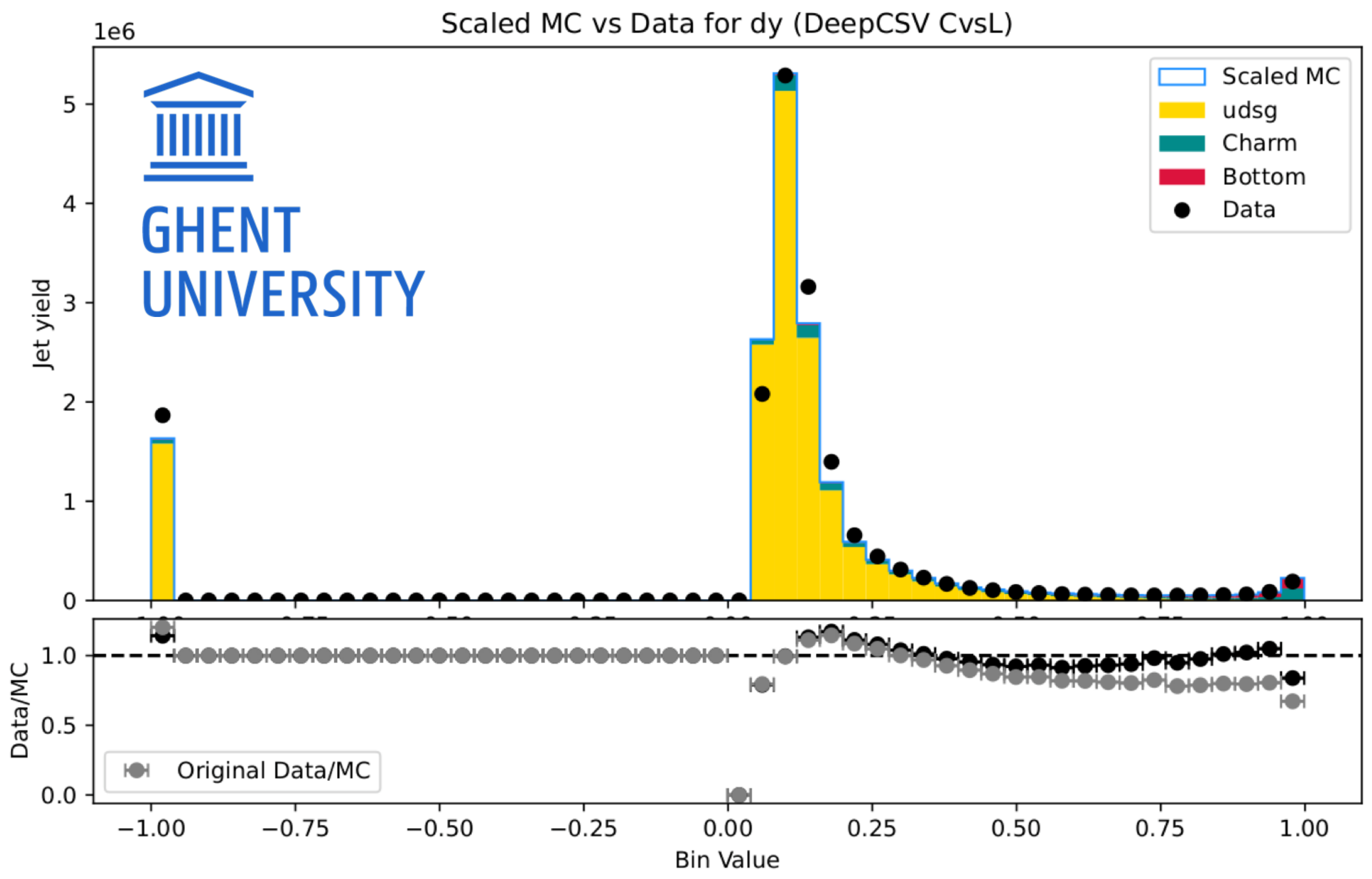
Work by C. Ramón Alvarez et al.

Now working on calibration method for b and c jet charge tagging

Full supervised unification

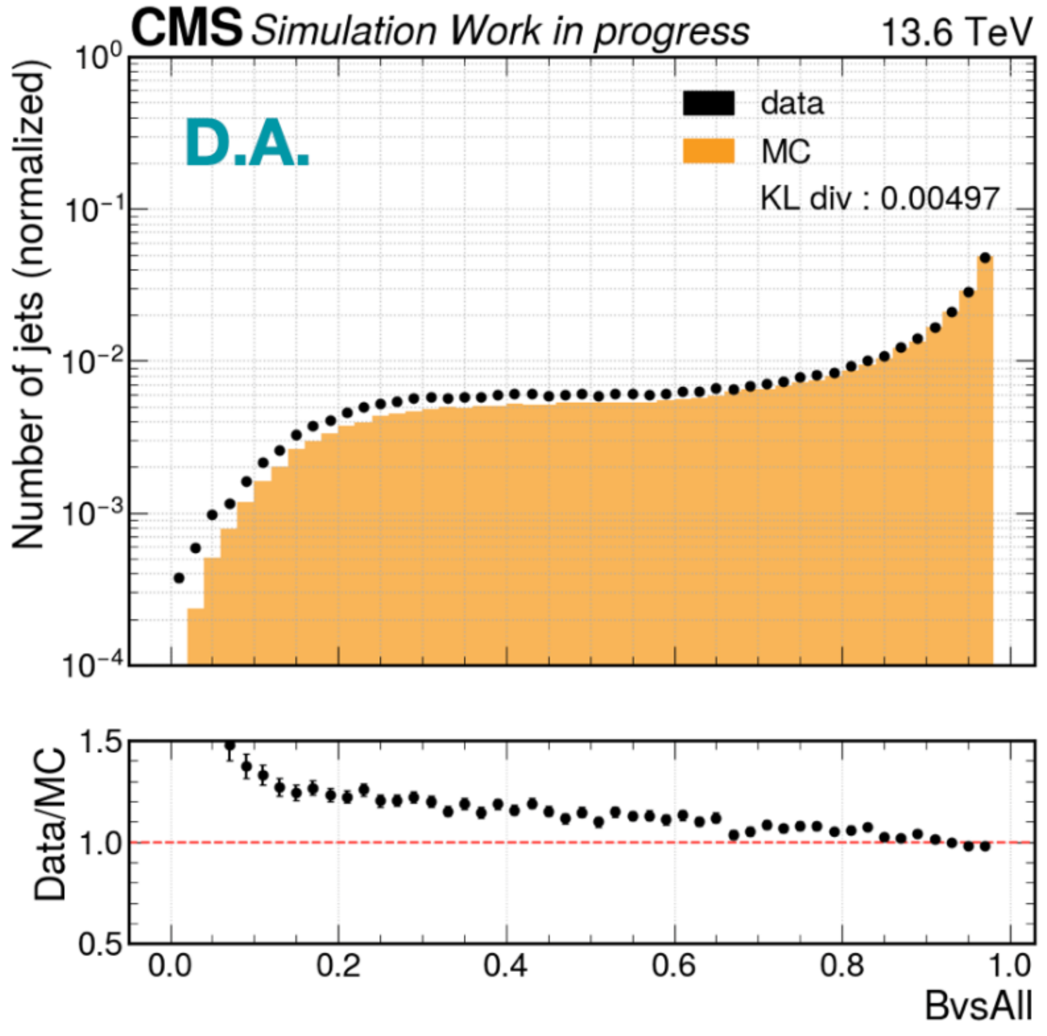
We could extend the unification towards more tasks:

- Full object reconstructions such as hadronic tau or b/c hadrons
- Jet charge tagging
- In-training ML based calibration

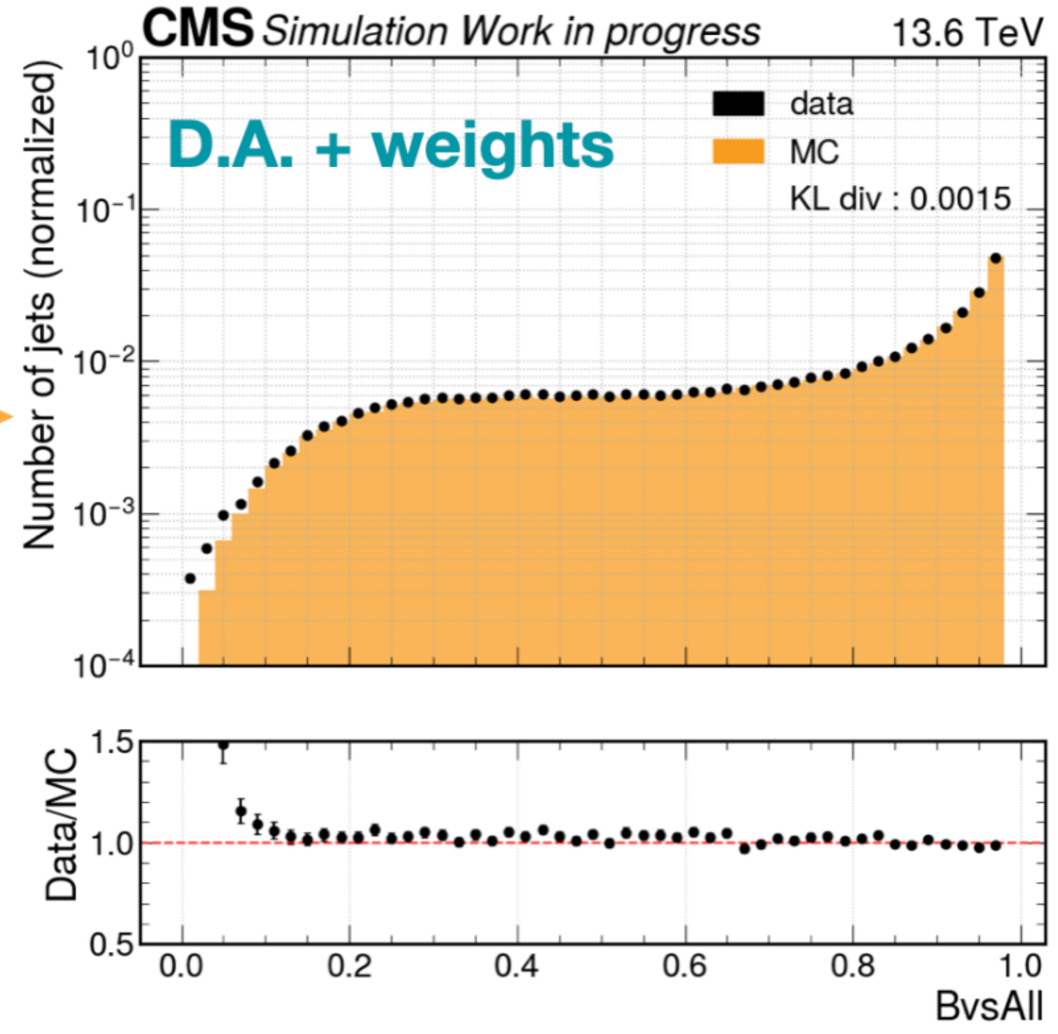


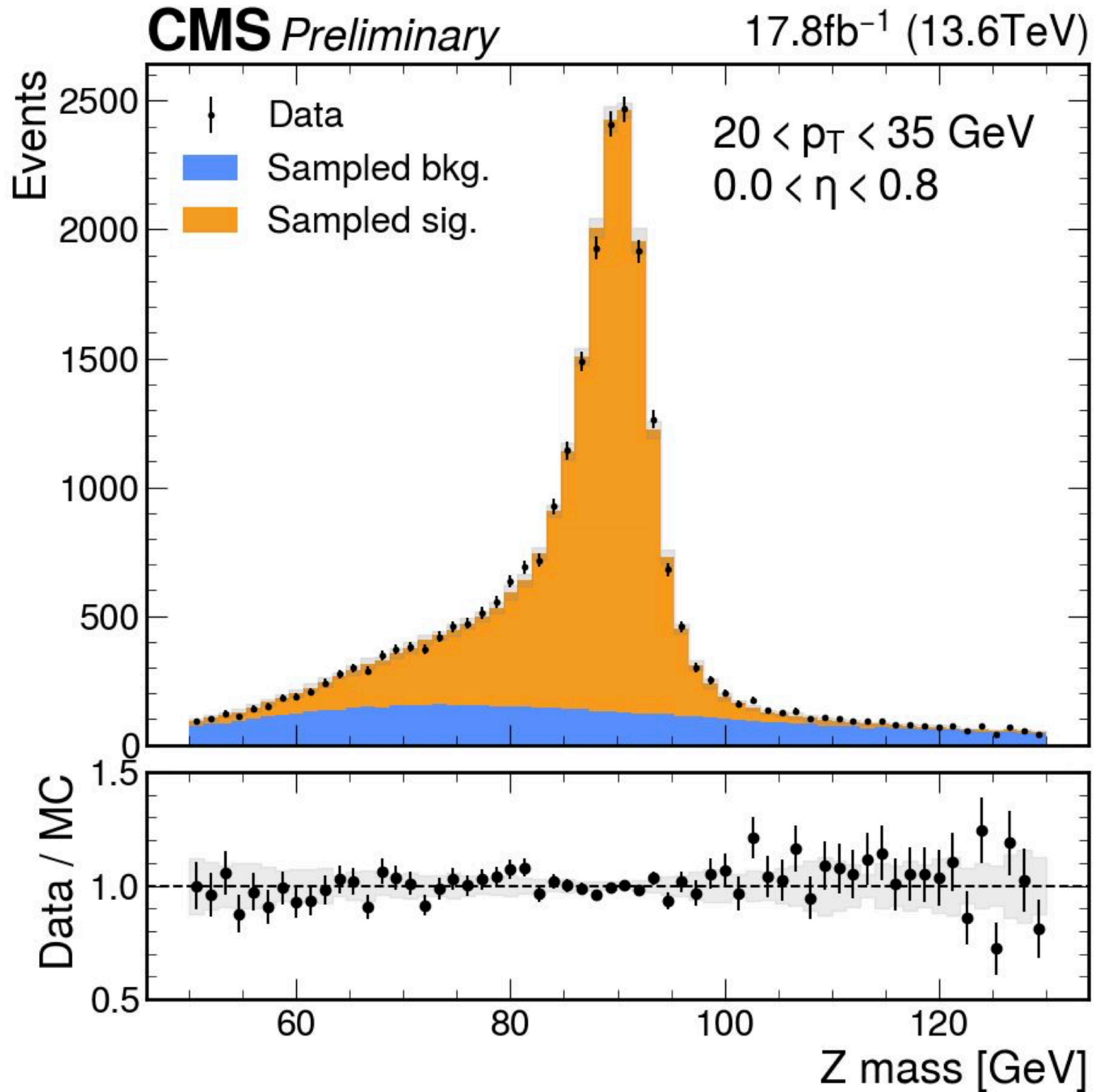
DeepSF

Domain adaptation



Very good fit except the low prob (other flavor contamination ?)





Also *CMS-DP-2025-053*: public result on lepton id. calib. but will be tested on b-tagging soon

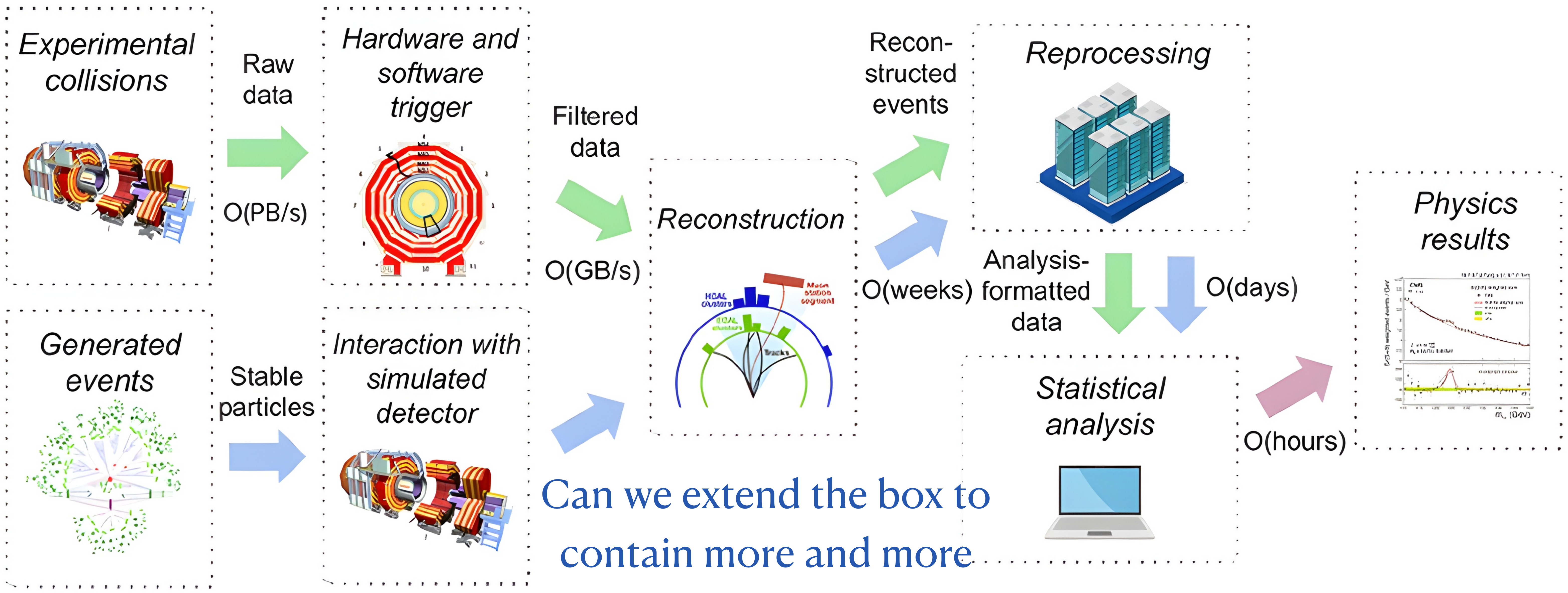
Towards the unsupervised world model

Front.Big Data 4 (2021) 661501

Fully supervised model will always be focused only on the loss function tasks

Need to go to unsupervised for self discover:

- Fully understand and discover underlying properties
- One single large model (Jet/CMS GPT) you can fine-tune for any downstream tasks
- Probe on data: get rid of most of the mismodeling ?



Can we extend the box to contain more and more elements?

Towards the unsupervised world model

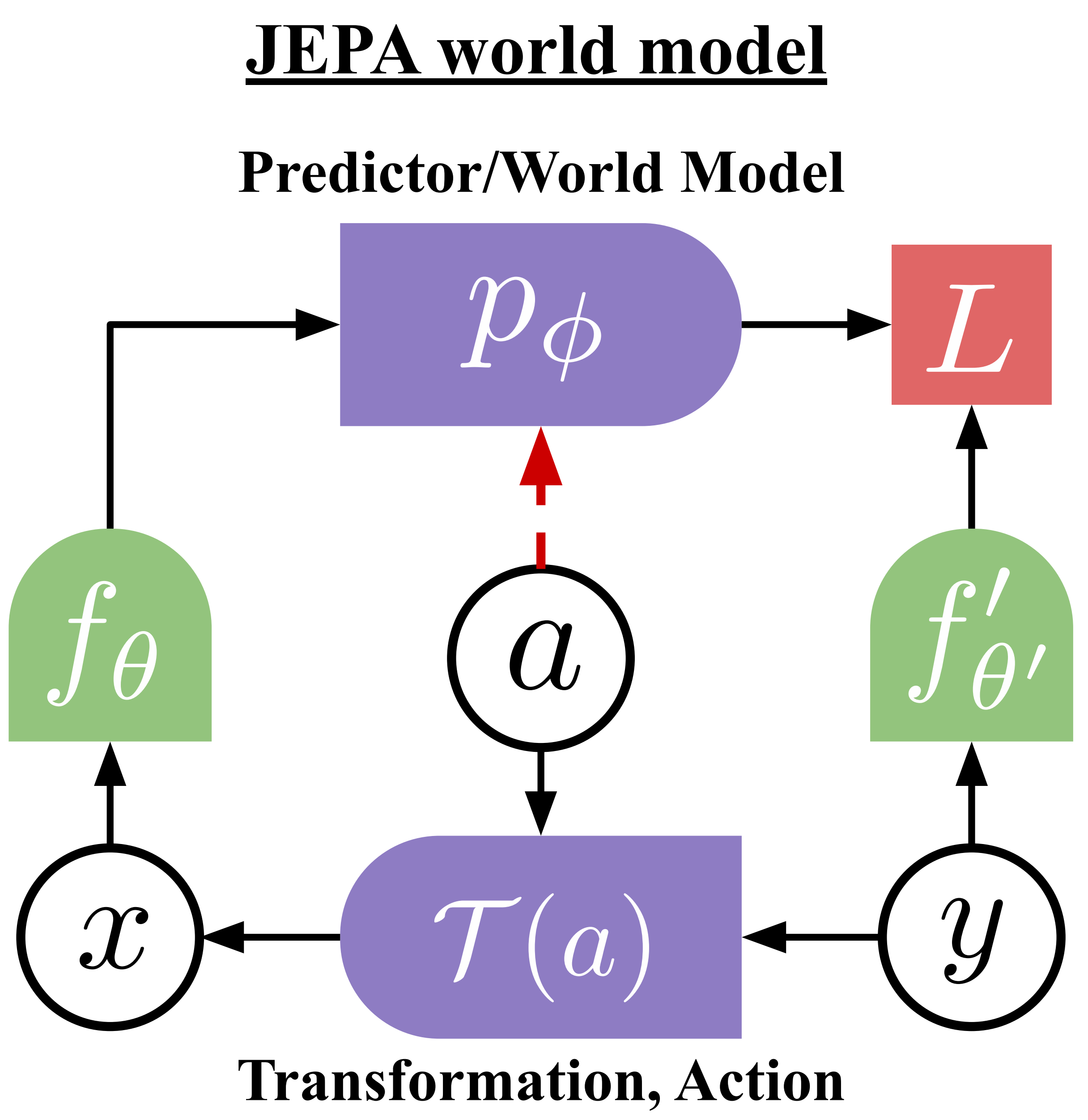
Joint-Embedding Predictive Architecture (JEPA):

Learn by predicting representations, not raw features
- Mask jet subregions: predict their latent embeddings from remaining context
- Train with MSE loss in embedding space

Advantages for HEP:
- Learns physics-informed representations automatically
- More robust than constituent-level prediction

Cons: rely on a predictor bias
- How can you make this reliable for HEP ?
- Can you get rid of the bias and avoid human-level knowledge intervention

arXiv:2403.00504

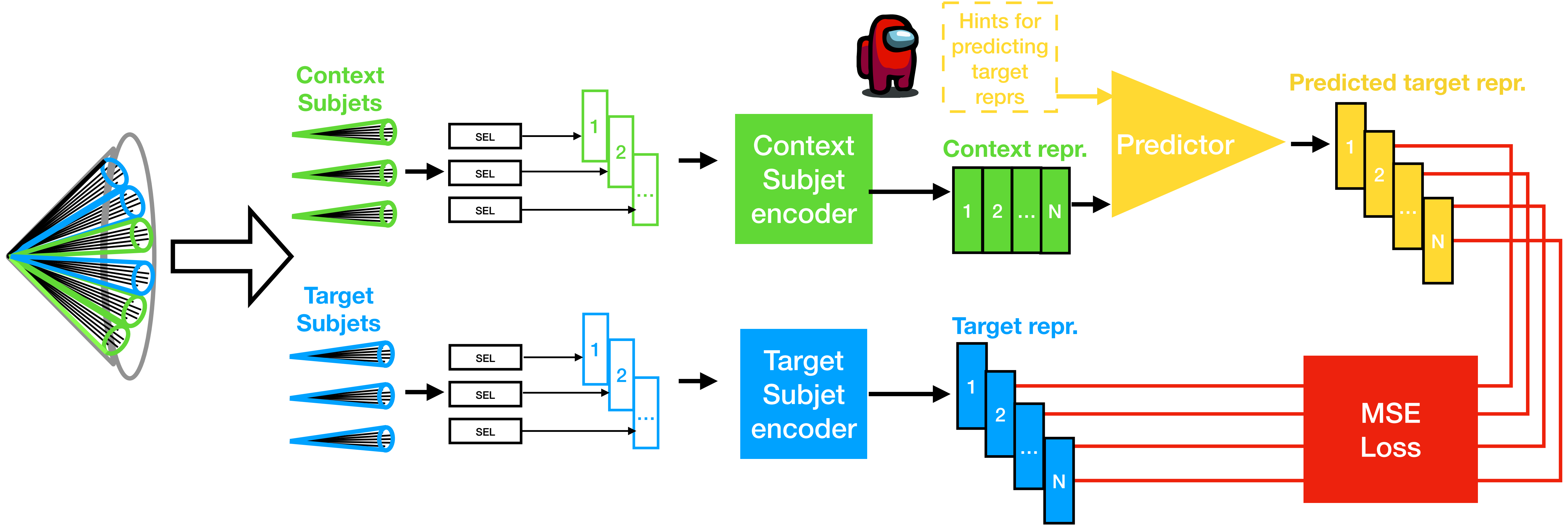


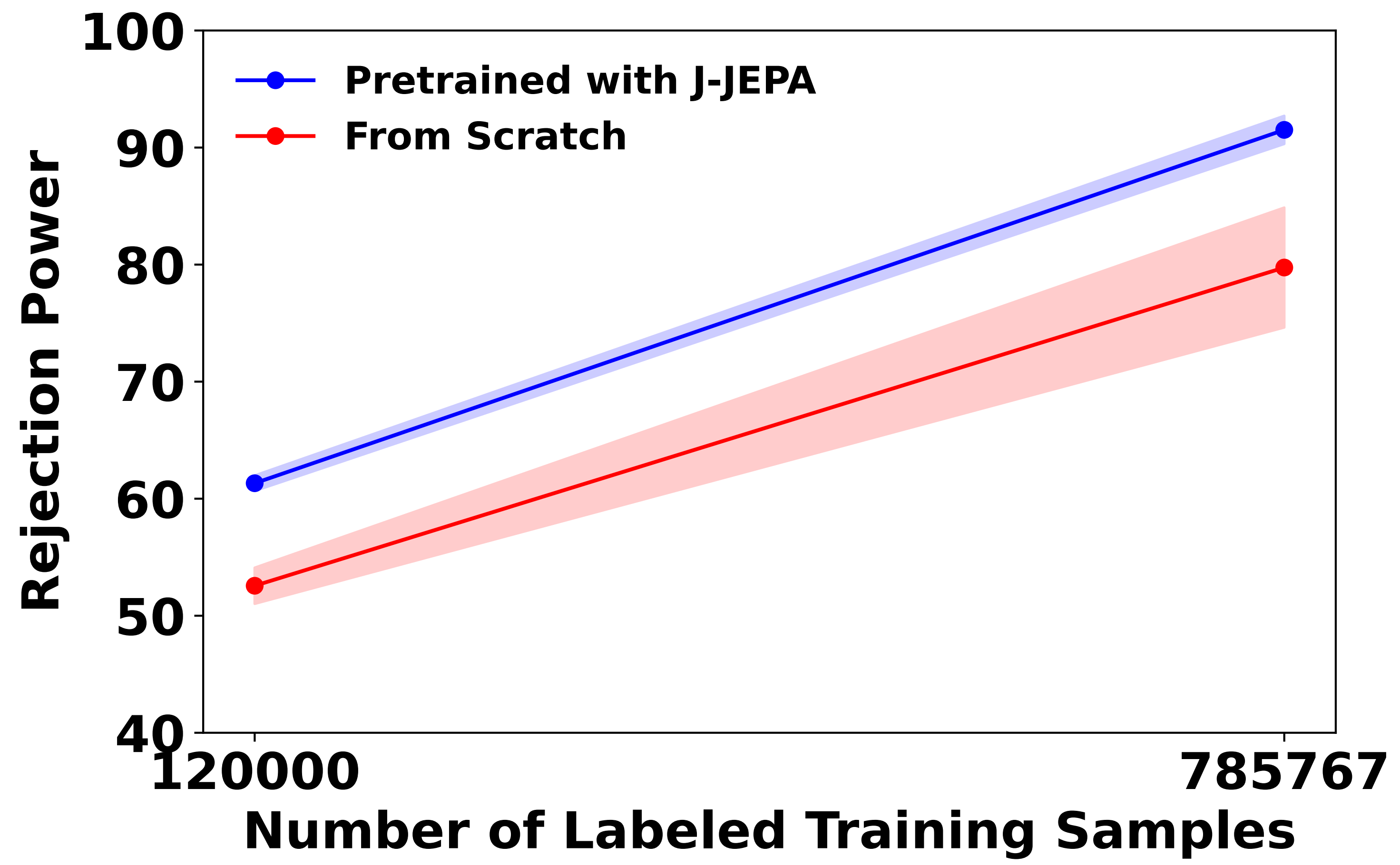
Towards the unsupervised world model

Advantages for HEP:

- Learns physics-informed representations automatically
- No labelling needed: can we achieve a training on data?
- Single pre-trained model: fine-tune for multiple tasks
- More robust than constituent-level prediction

First JEPA attempts in HEP: 2412.05333 and 2502.03933



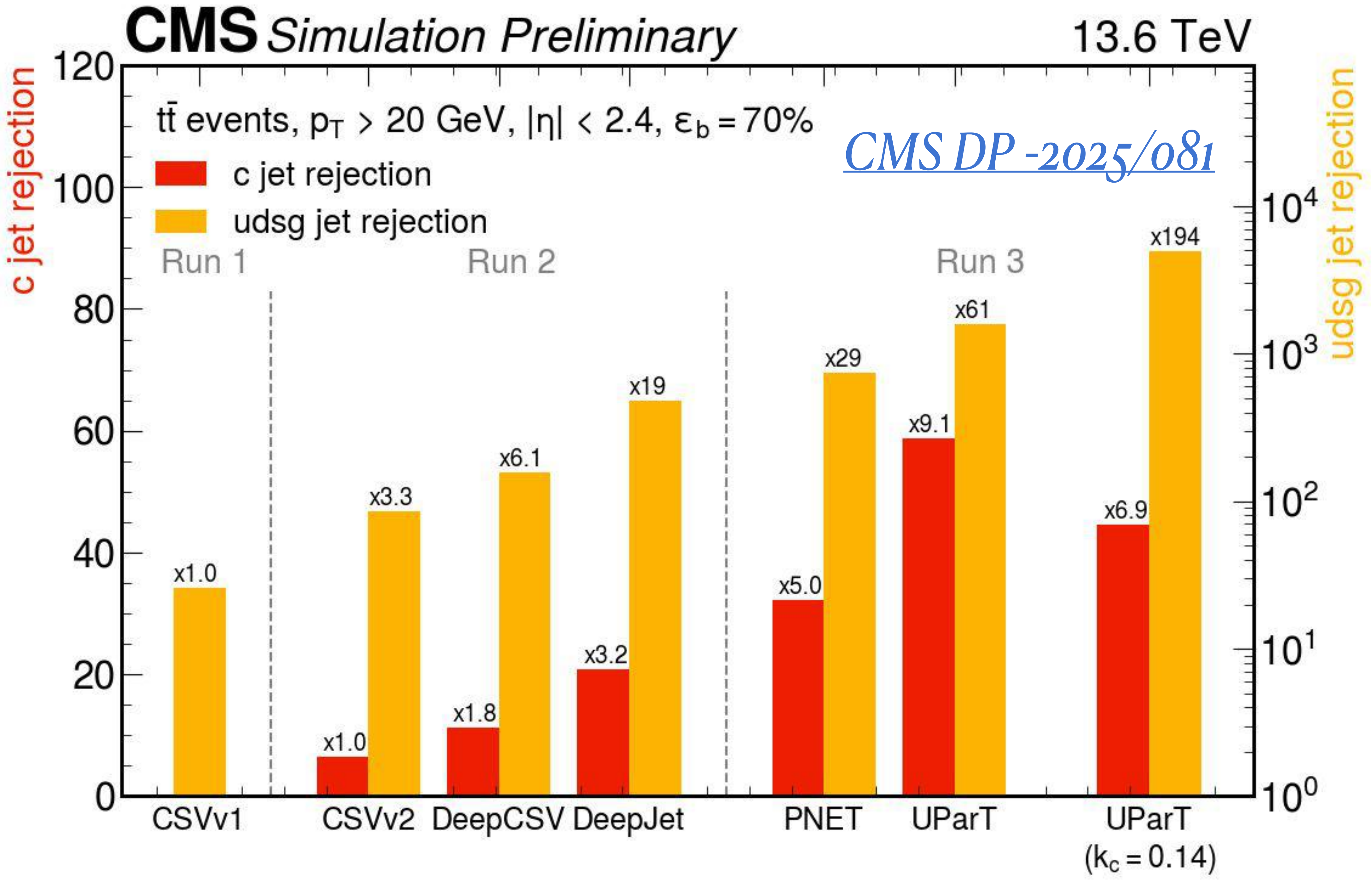


Summary

Deep learning has revolutionized jet algorithms in HEP:
- Evolved from handcrafted features to sophisticated DNN architectures.
- Growing field with still a lot to do

The **Unified Particle Transformer (UParT)** represents a significant milestone, achieving up to a factor **194 improvement** in background rejection compared to Run 1 taggers

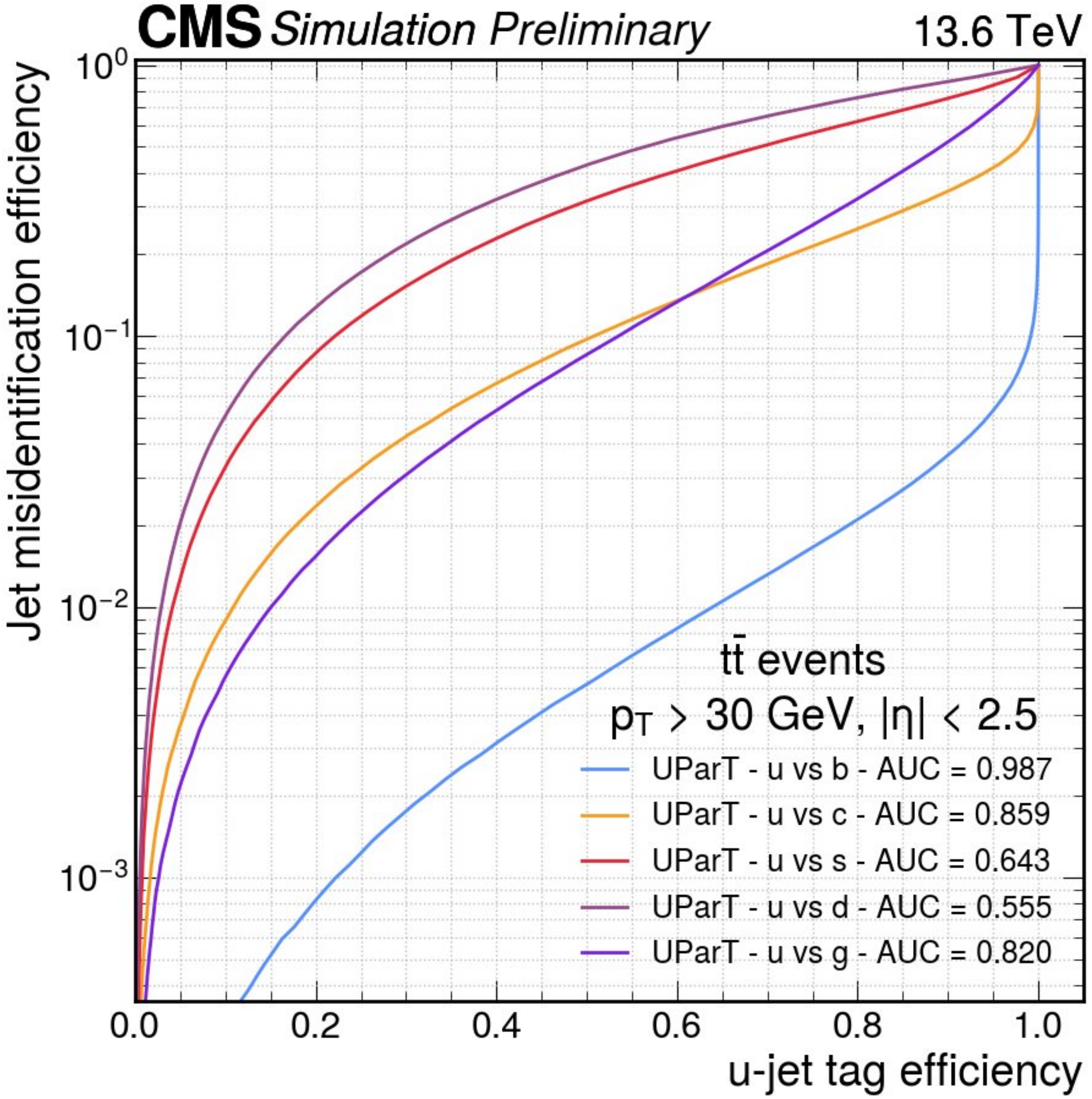
Key achievements:
- First s-tagging capability at the LHC
- Robust adversarial training (R-NGM) attempting to minimized data/MC disagreement
- State-of-the-art performance across most tagging tasks
- Demonstrated scaling laws with potential for further improvement



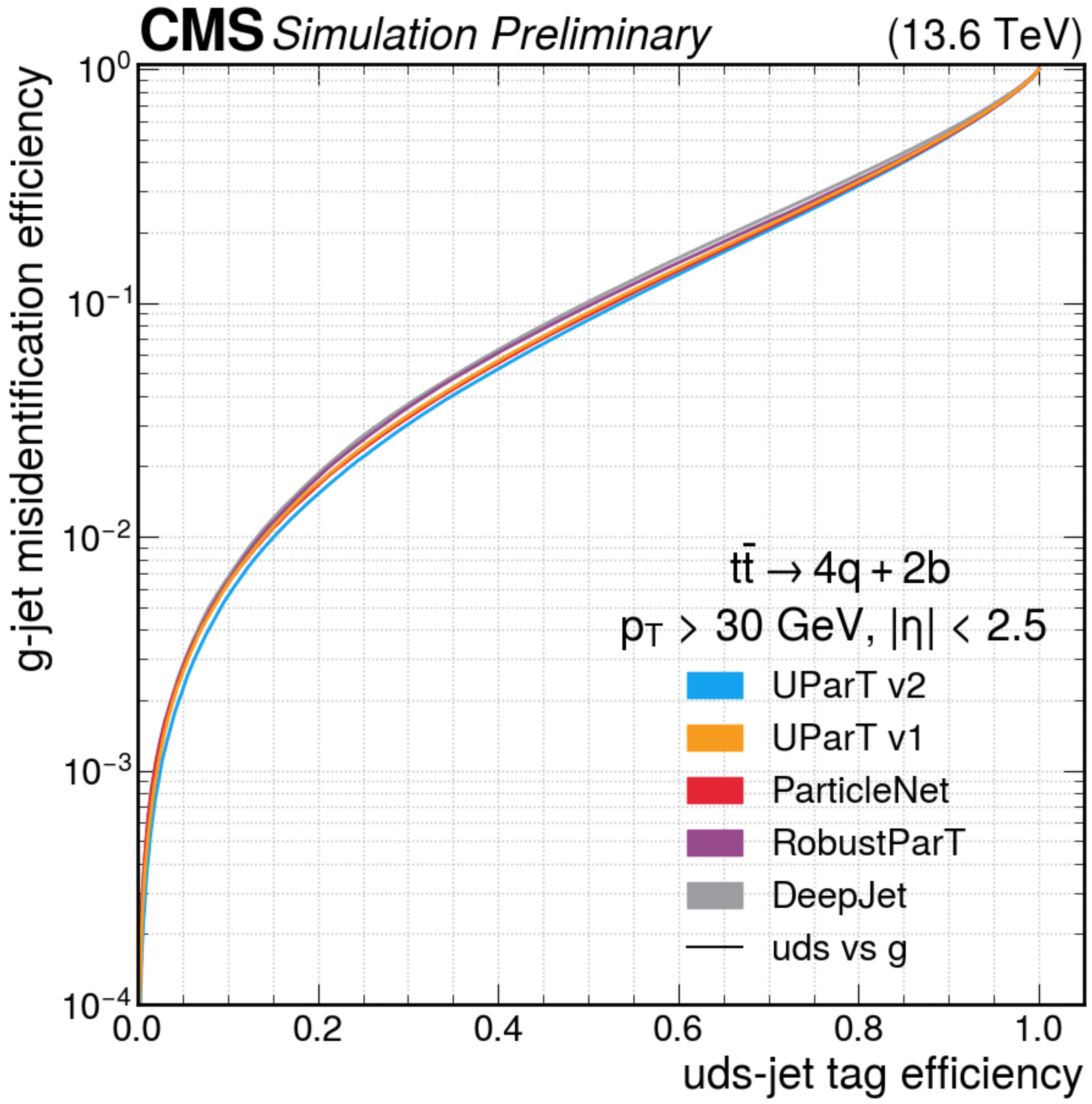
CMS DP -2025/081

Encyclo-[mascot]-dia

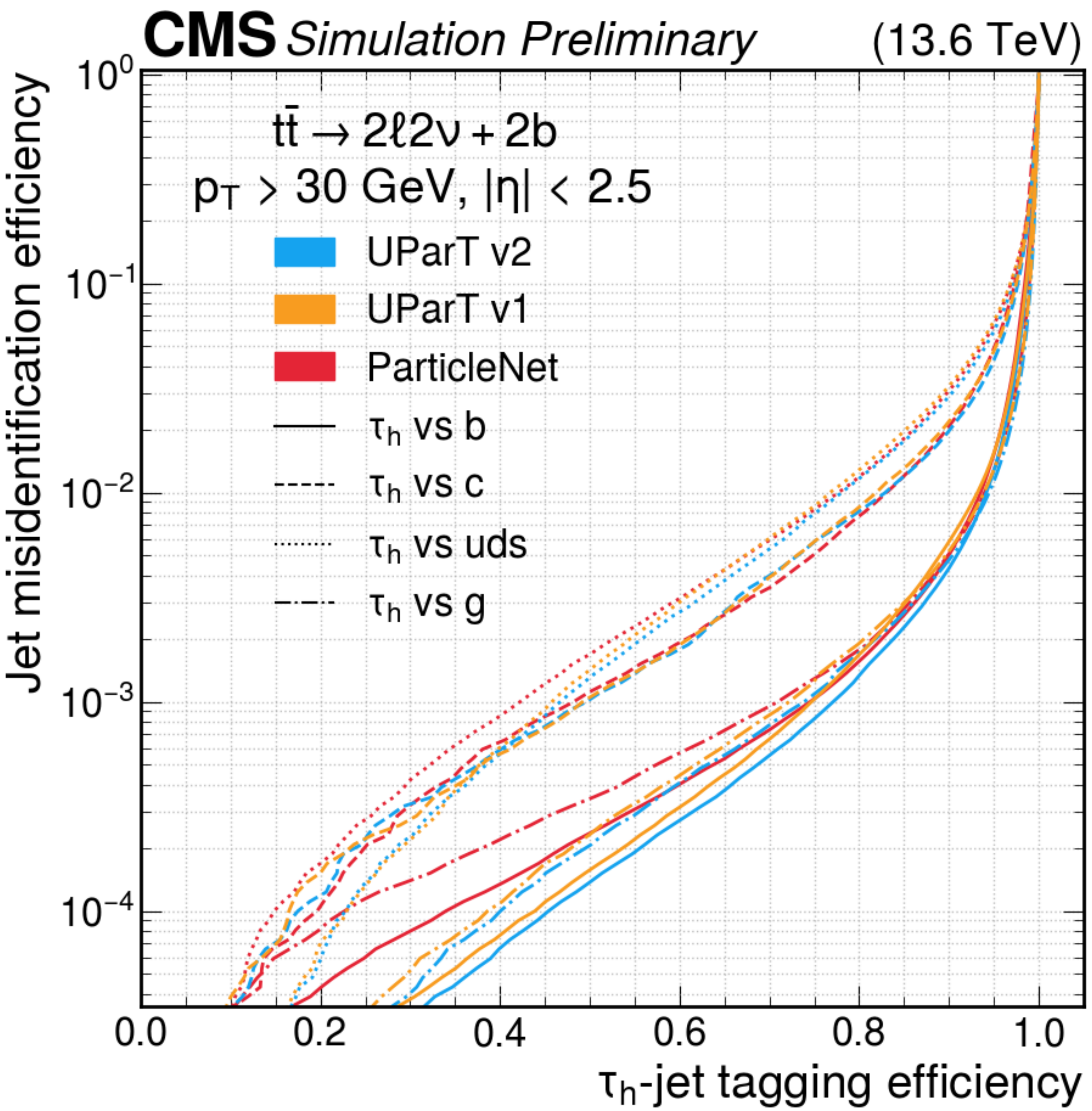
UvsD tagging

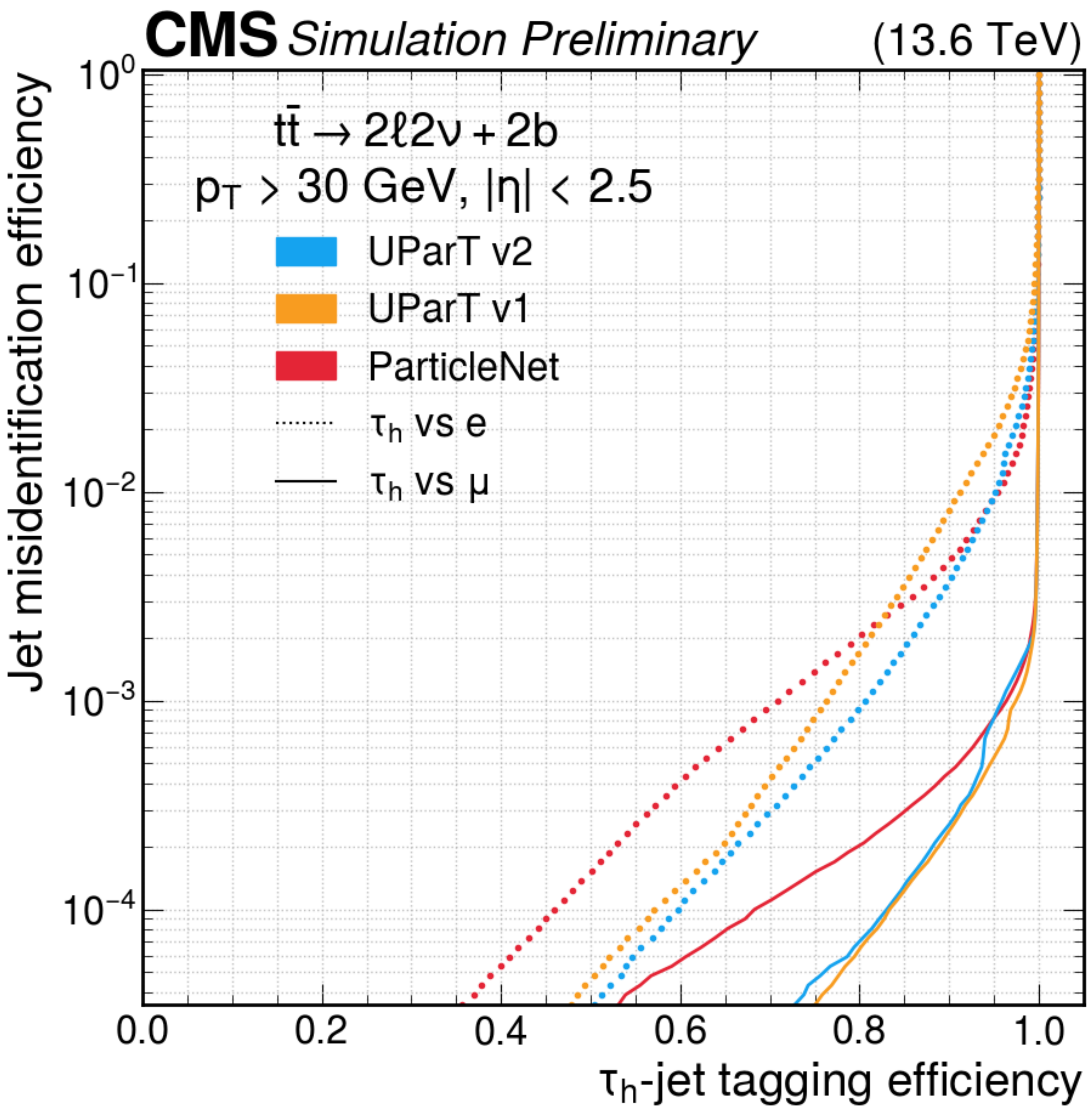


QG tagging

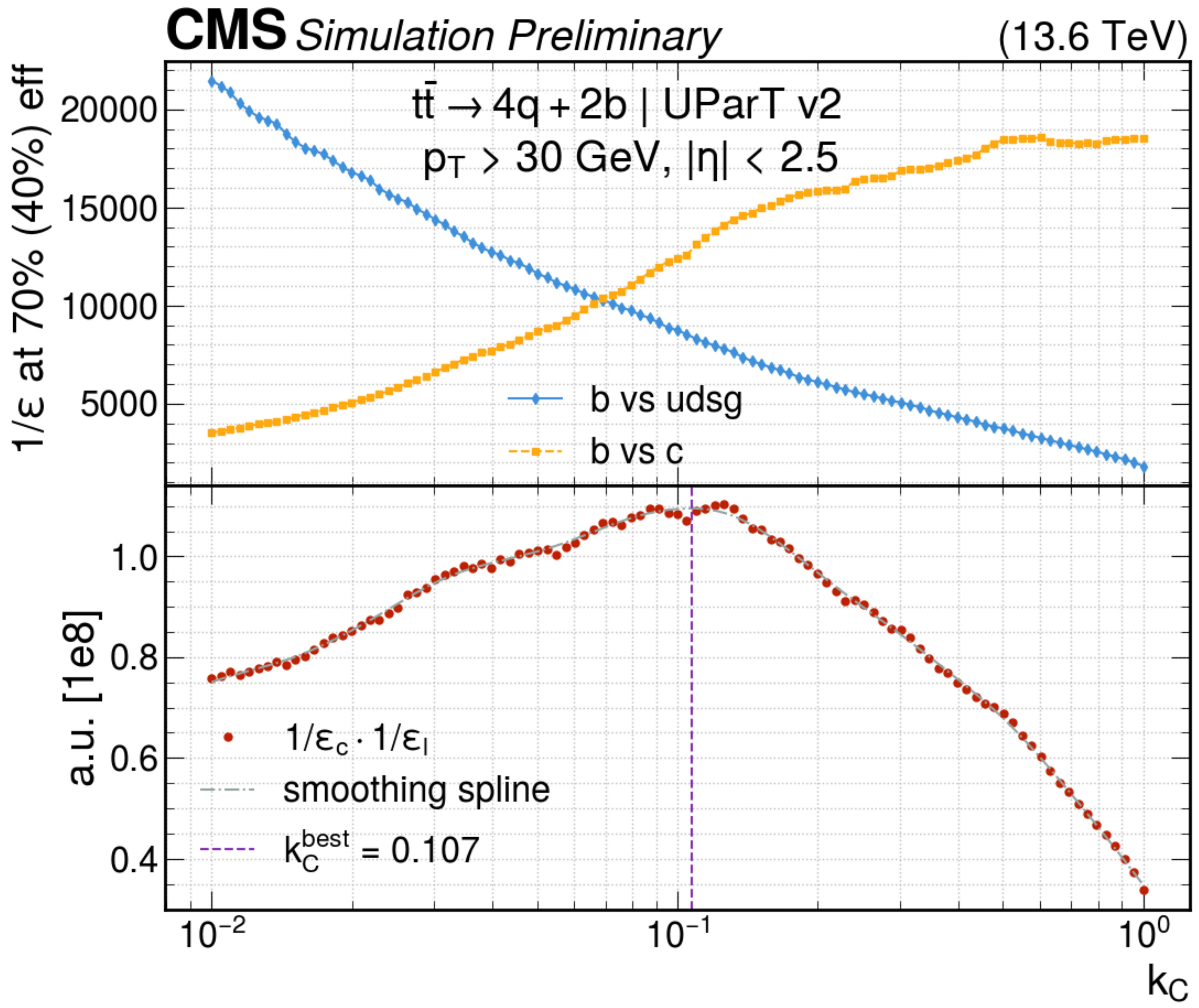

CMS Simulation Preliminary
(13.6 TeV)
g-jet misidentification efficiency
uds-jet tag efficiency
$t\bar{t} \rightarrow 4q + 2b$
$p_T > 30$ GeV, $|\eta| < 2.5$
UParT v2
UParT v1
ParticleNet
RobustParT
DeepJet
uds vs g

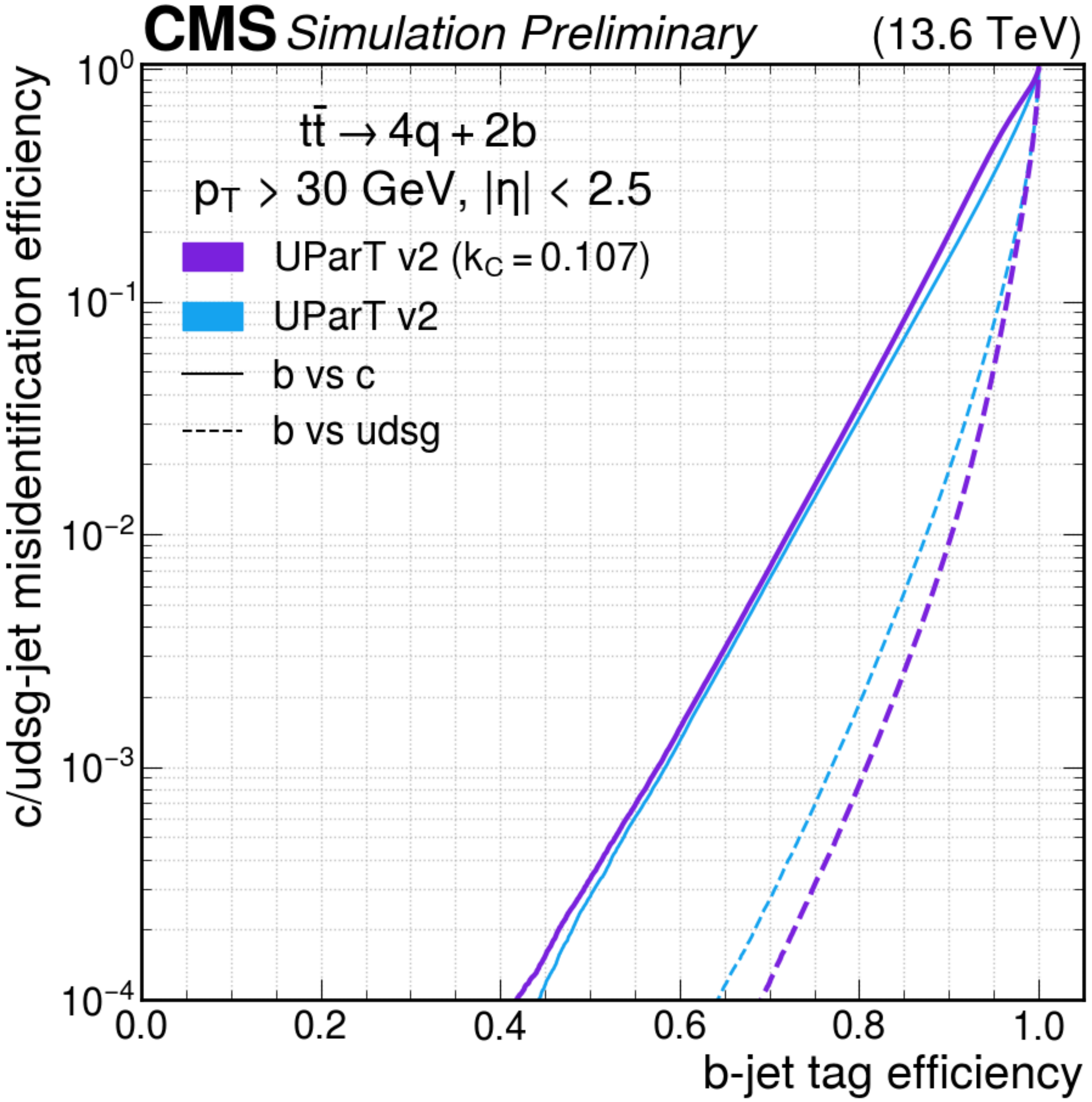
τ tagging


CMS Simulation Preliminary
(13.6 TeV)
$t\bar{t} \rightarrow 2\ell 2\nu + 2b$
$p_T > 30$ GeV, $|\eta| < 2.5$
UParT v2
UParT v1
ParticleNet
τ_h vs b
τ_h vs c
τ_h vs uds
τ_h vs g
Jet misidentification efficiency
τ_h-jet tagging efficiency


CMS Simulation Preliminary
(13.6 TeV)
$t\bar{t} \rightarrow 2\ell 2\nu + 2b$
$p_T > 30$ GeV, $|\eta| < 2.5$
UParT v2
UParT v1
ParticleNet
τ_h vs e
τ_h vs μ
Jet misidentification efficiency
τ_h-jet tagging efficiency

Weighted b-tagging



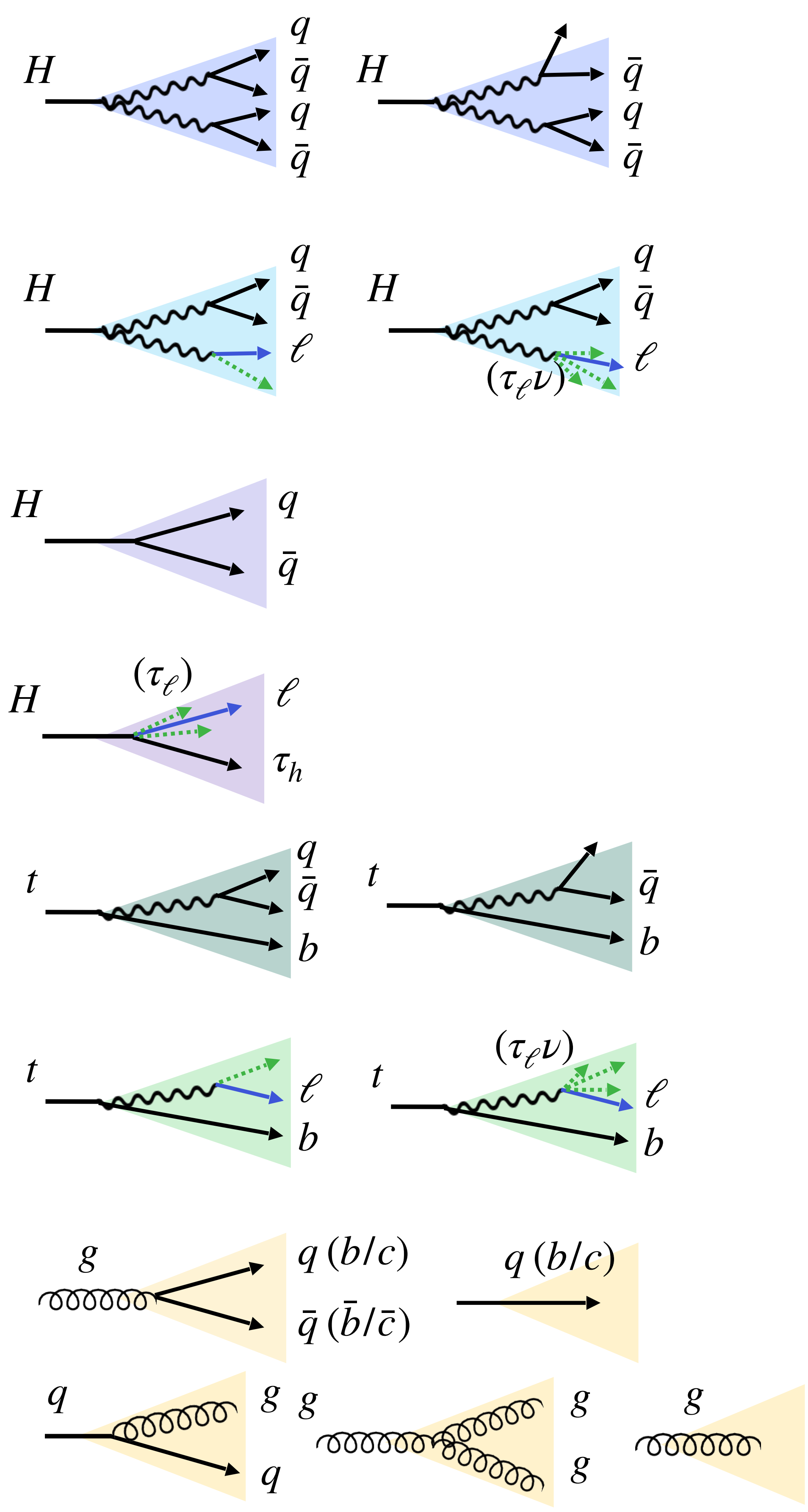


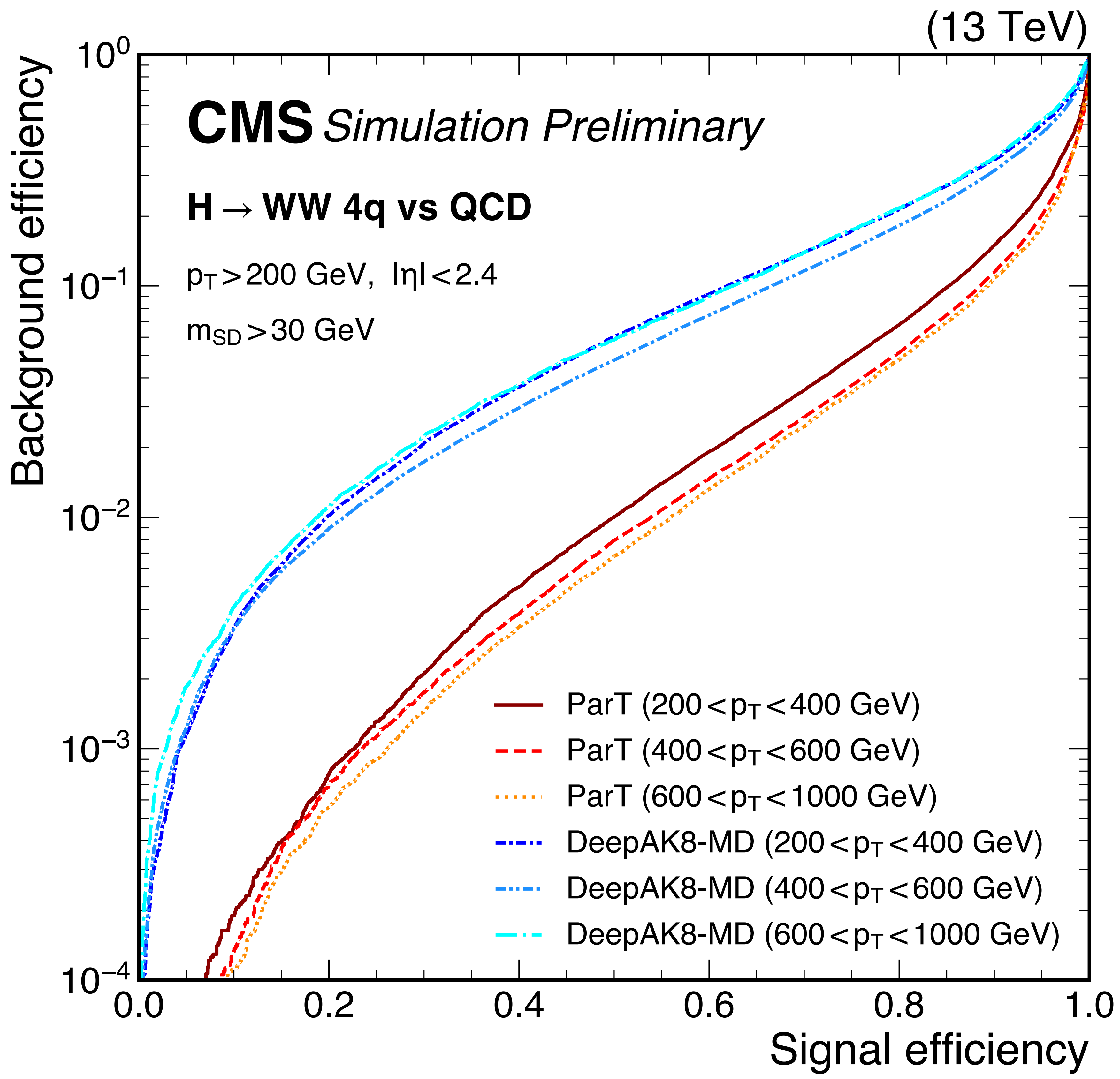
$$\text{BvsAll weighted} = \frac{\text{prob(b)}}{k_c \cdot \text{prob(c)} + (1 - k_c) \cdot \text{prob(udsg)}}$$

Jet regression

GloParT v1

Process	Final state		Flavor	# of classes
H→WW (full-hadronic)	qqqq	⊗	0c / 1c / 2c	3
	qqq			3
H→WW (semi-leptonic)	eνqq	⊗	0c / 1c	2
	μνqq			2
	τ_eνqq			2
	τ_μνqq			2
	τ_hνqq			2
H→qq		⊗	bb	1
			cc	1
			ss	1
			qq (q=u/d)	1
H→ττ	$\tau_e\tau_h$			1
	$\tau_\mu\tau_h$			1
	$\tau_h\tau_h$			1
t→bW (hadronic)	bqq	⊗	1b + 0c / 1c	2
	bq			2
t→bW (leptonic)	beν	⊗	1b	1
	bμν			1
	bτ_eν			1
	bτ_μν			1
	bτ_hν			1
QCD			b	1
			bb	1
			c	1
			cc	1
			others (light)	1





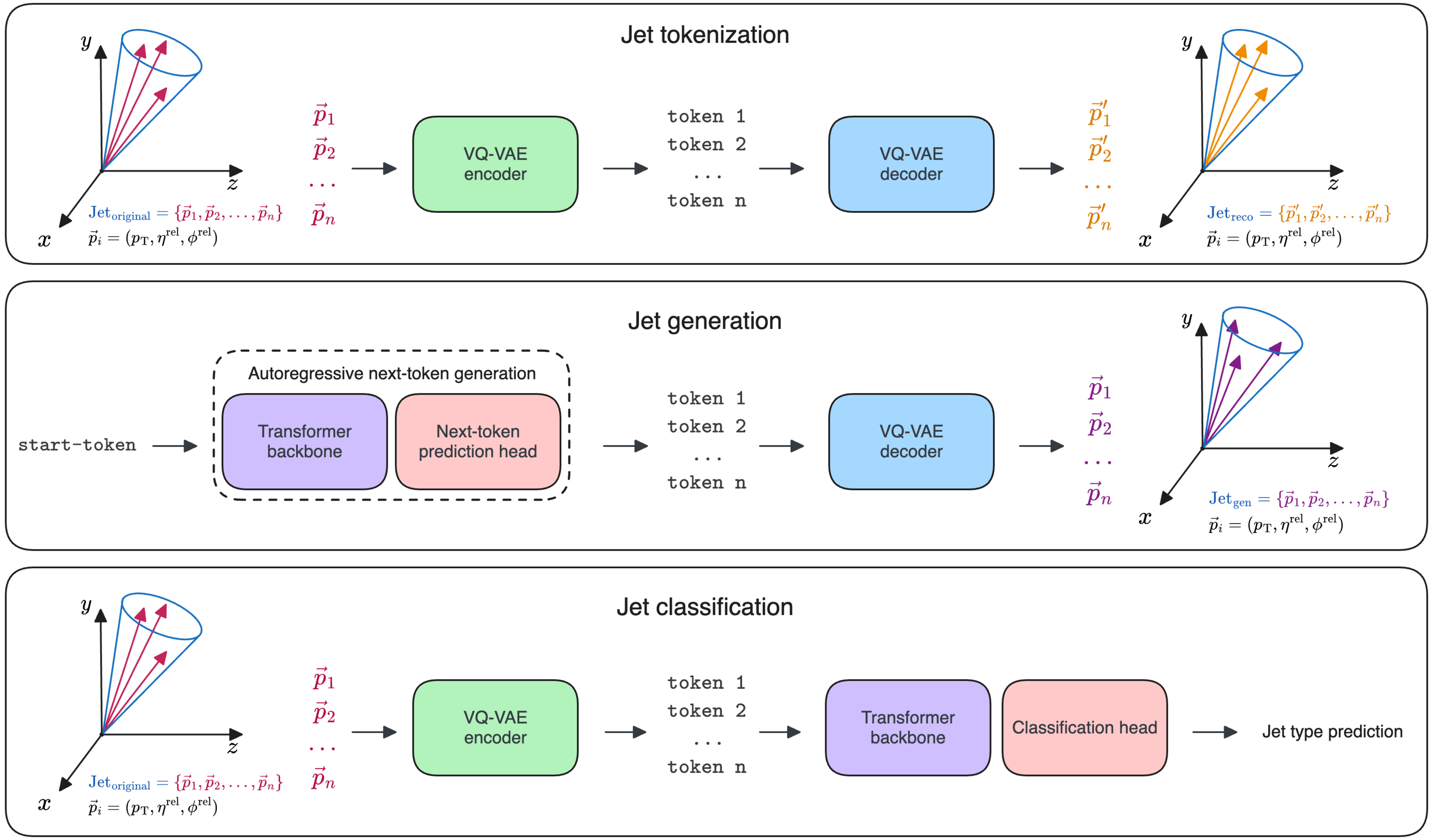
CMS-PAS-JME-25-001

Unsupervised world model: autoregression

Consider the jet as a sequence:

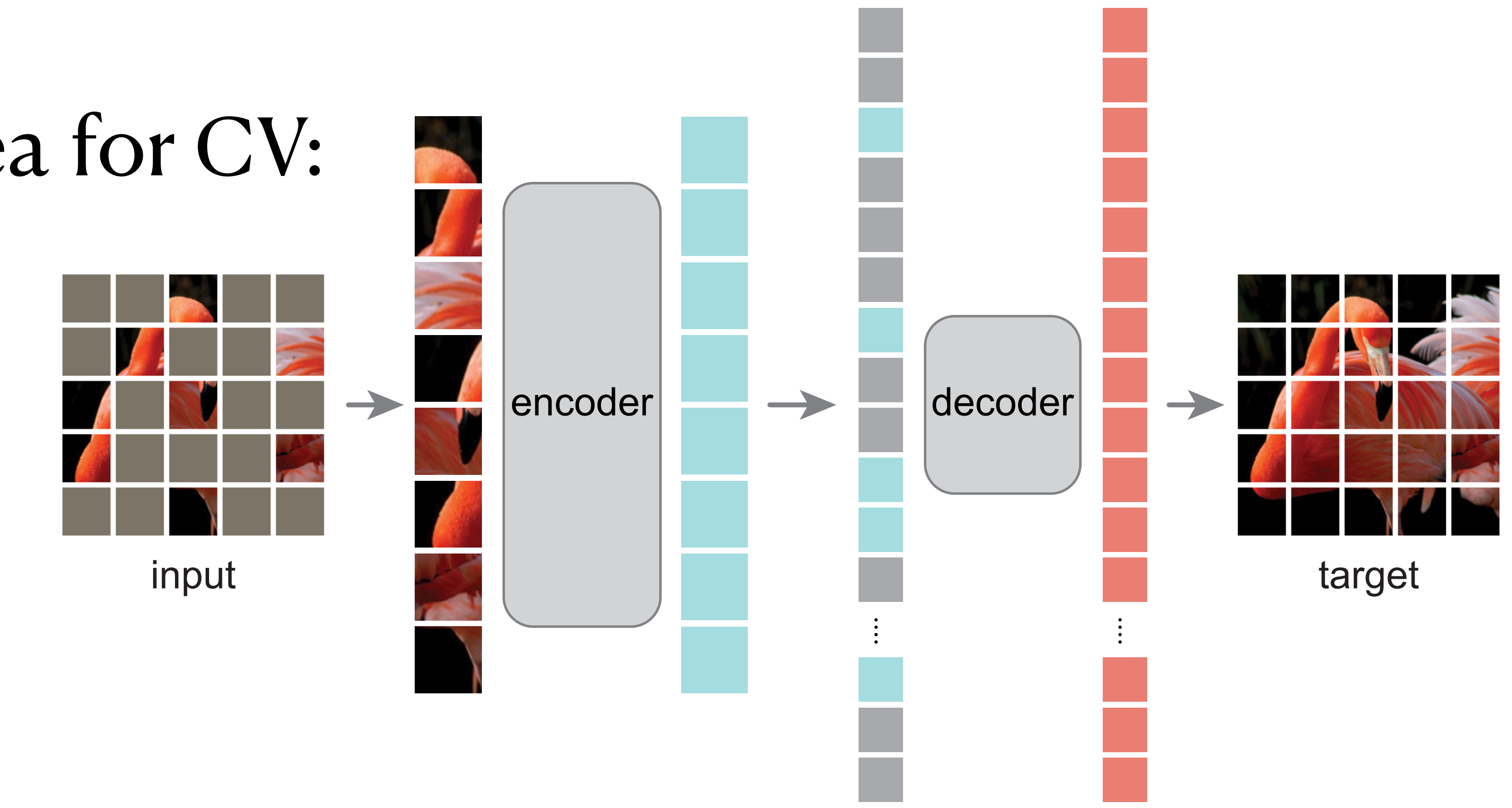
- Create a tokenizer (constituent to latent space compression)
- Try to predict the next token (constituent)

Example: [*OmniJet-α*]



Unsupervised world model: masked modeling

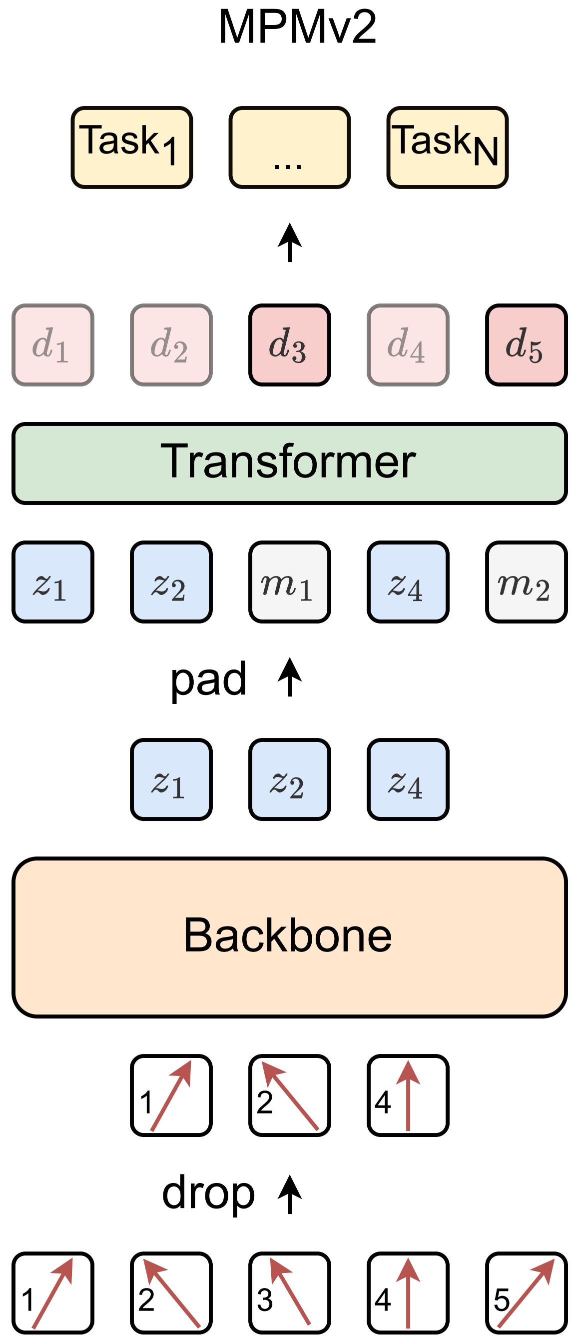
Original idea for CV:

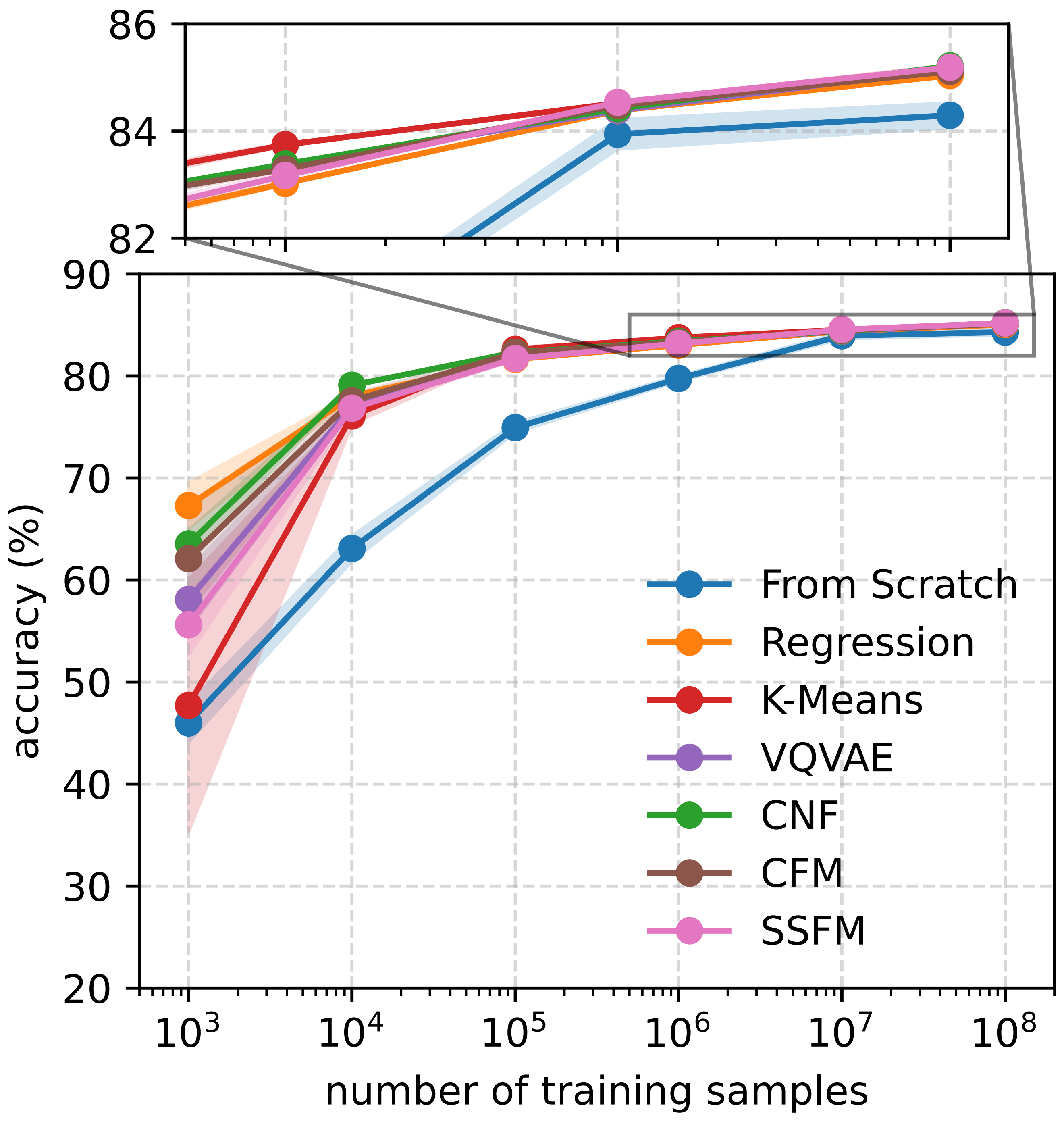


Masked Particle Modeling is similar to Masked Autoencoder:

- No discrete tokenization
- Assumed Masking and Reconstructing the missing patches is possible for a jet

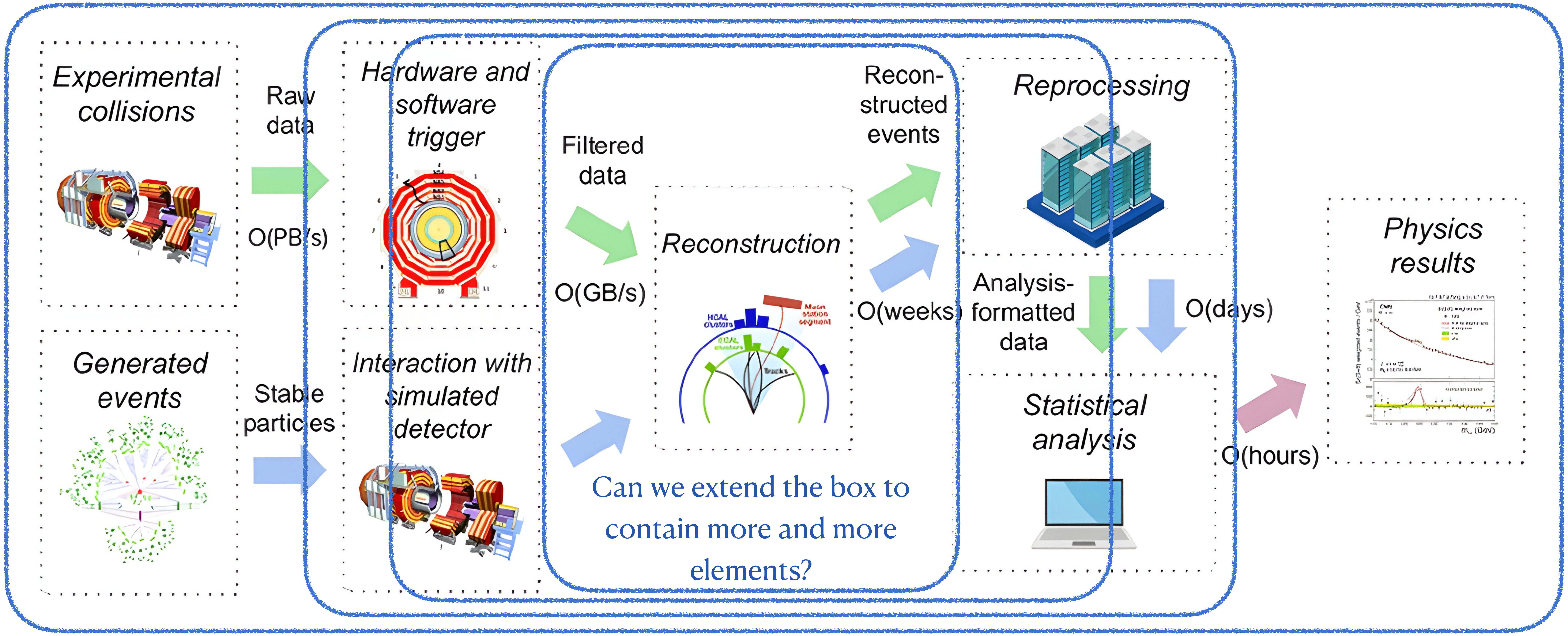
Example: *MPMv1* and *MPMv2*





Beyond the jet itself

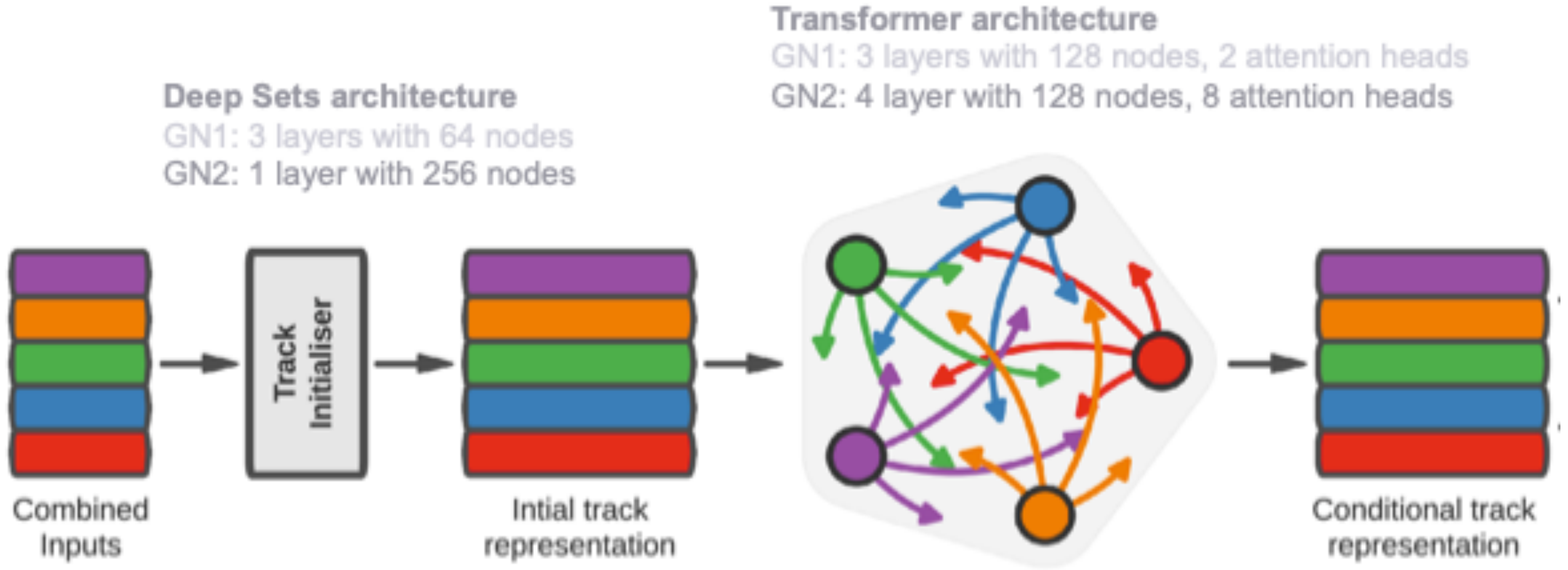
Front.Big Data 4 (2021) 661501



Beyond this: could we create a world model from hits to the statistical analysis (including all the tracking, calorimetry, PF, vertexing, etc) ? On data and/or simulations ?

Training time

GN2 algorithm structure



GN2 is a multi-modal and multi-task algorithm based on transformer architecture ("Pre-LN Transformer") w/ about 2.6M parameters trained on mixture of ttbar and Z' jets .

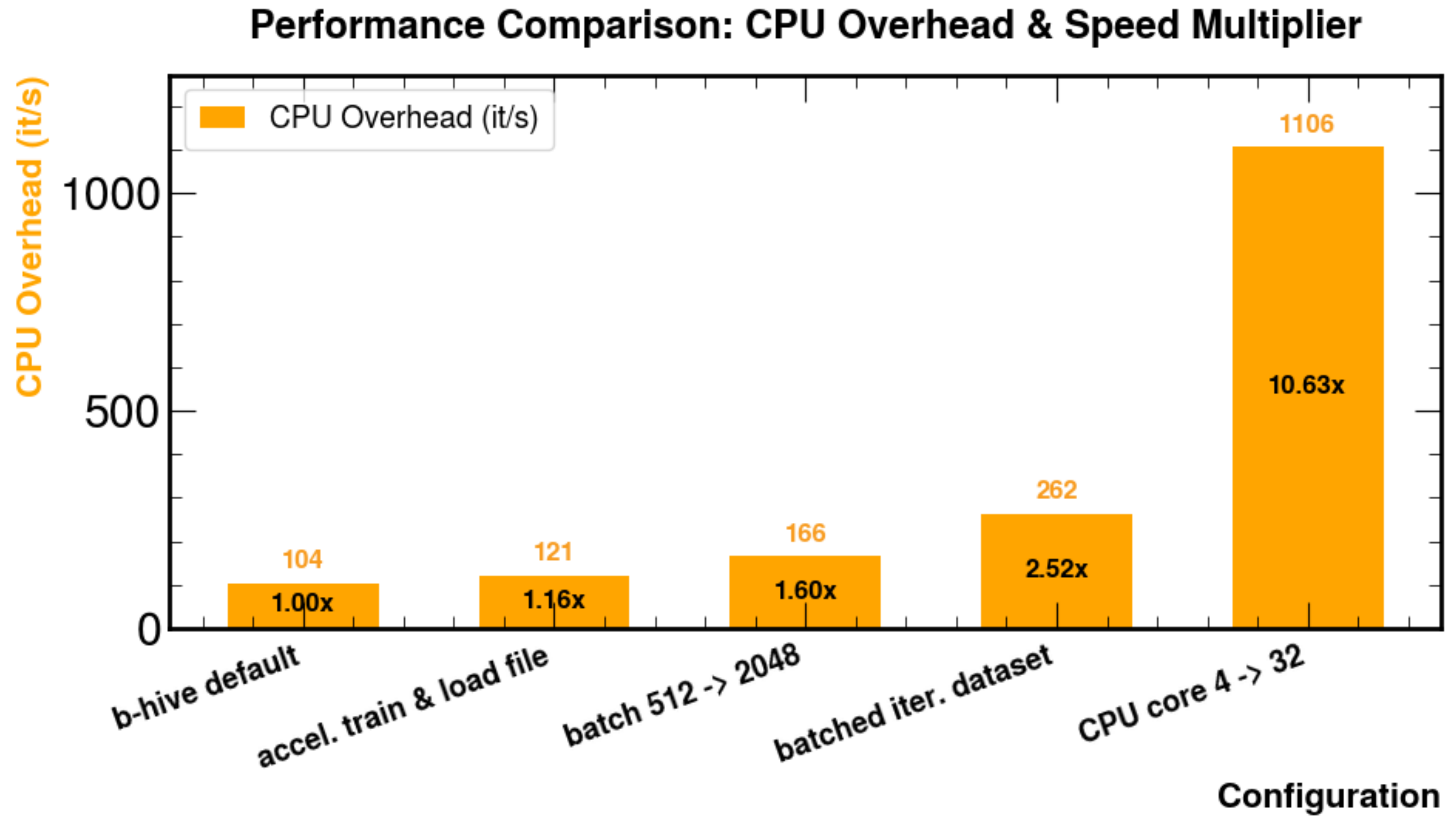
Auxiliary tasks improve training convergence and enhance overall performance. Training with OneCycleLR.

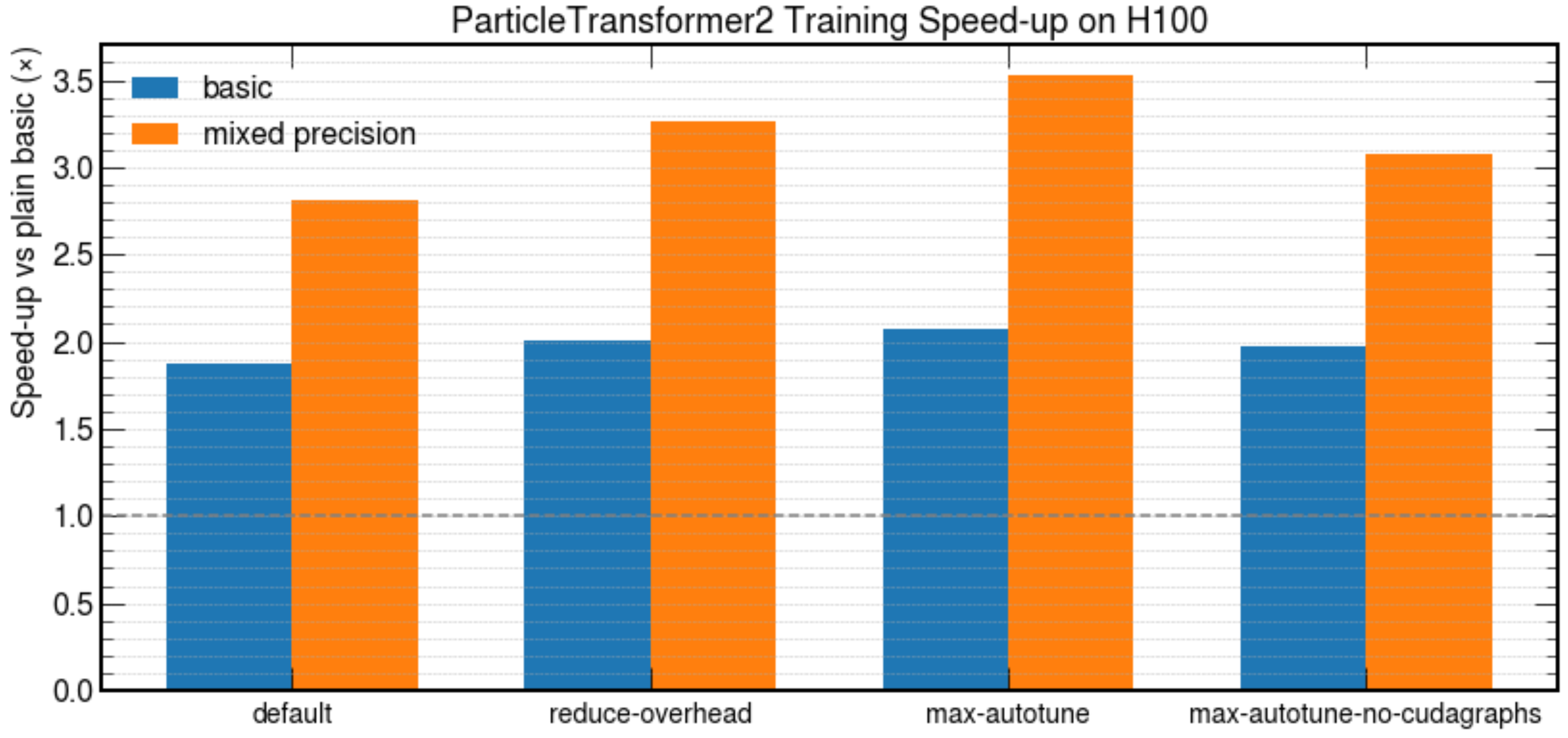
Trainings run for 300M jets with 1 NVIDIA A100 GPU → 4.8 hours / epoch. Software for training is "salt".

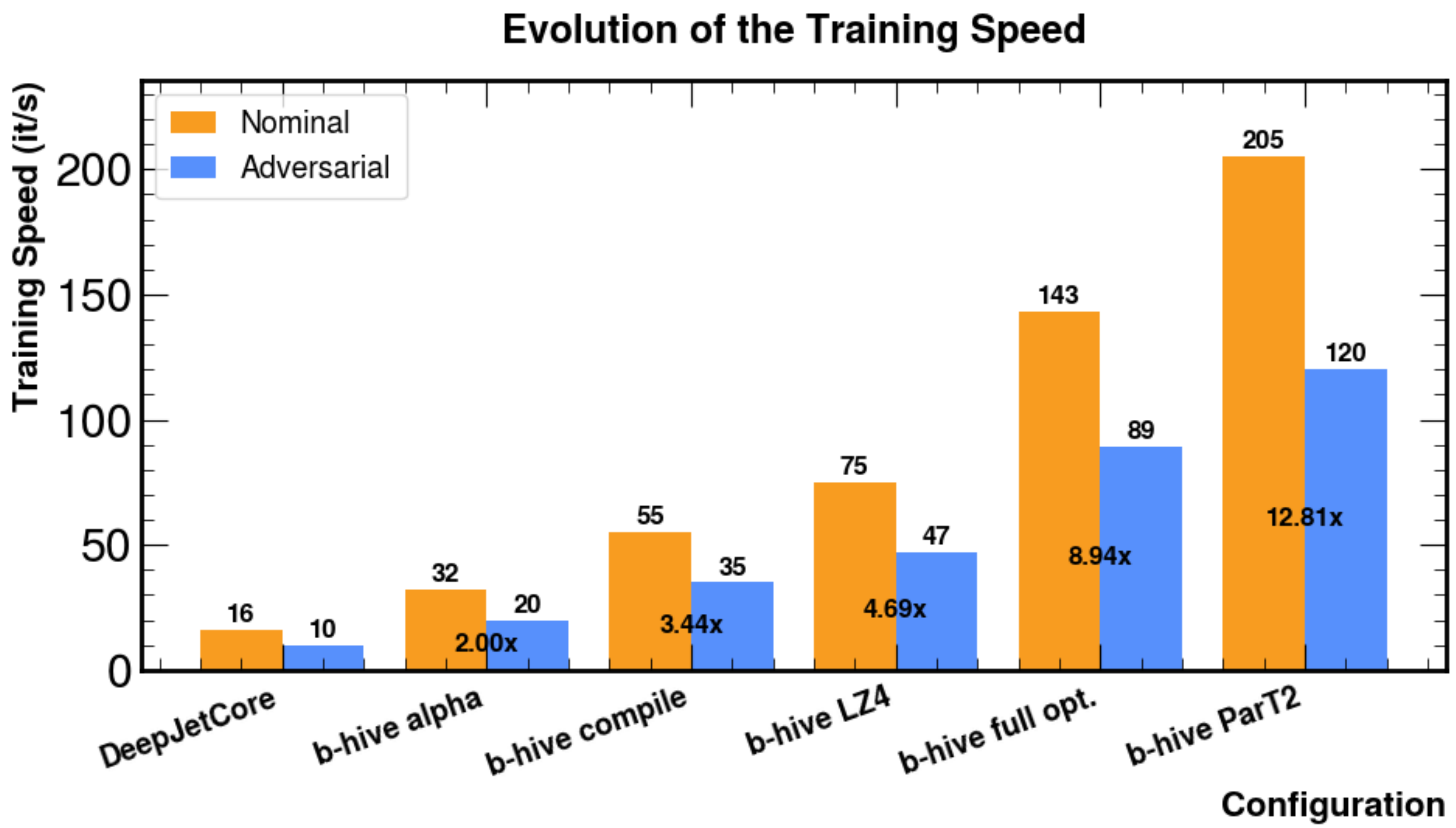
	Size of model	Training time	Dataset size	# of GPUs
ATLAS GN2	2.6 M	~ week	300 M	4
CMS UParT	2 M	1.5 days	30 M	1
ATLAS GN3	12 M	~ 2 weeks	410 M	4
CMS UParTv2	5.7 M	3 days	70 M	1

FTAG algorithms in ATLAS

Training time: b-hive benchmark







Training speed of a default ParT model (3+1 attention layers, dim=128 and nhead=8)