CZECH INSTITUTE OF INFORMATICS ROBOTICS AND CYBERNETICS CTU IN PRAGUE

Identifying Heavy-Flavor Jets Using Vectors of Locally Aggregated Descriptors

Georgy Ponimatkin (CIIRC CTU/NPI CAS/FNSPE CTU)
In collaboration with: R. Kunnawalkam Elayavalli (WSU), J. Šivic (DI ENS/Inria/CIIRC CTU), J. Bielčíková (NPI CAS) and J. Putschke (WSU)

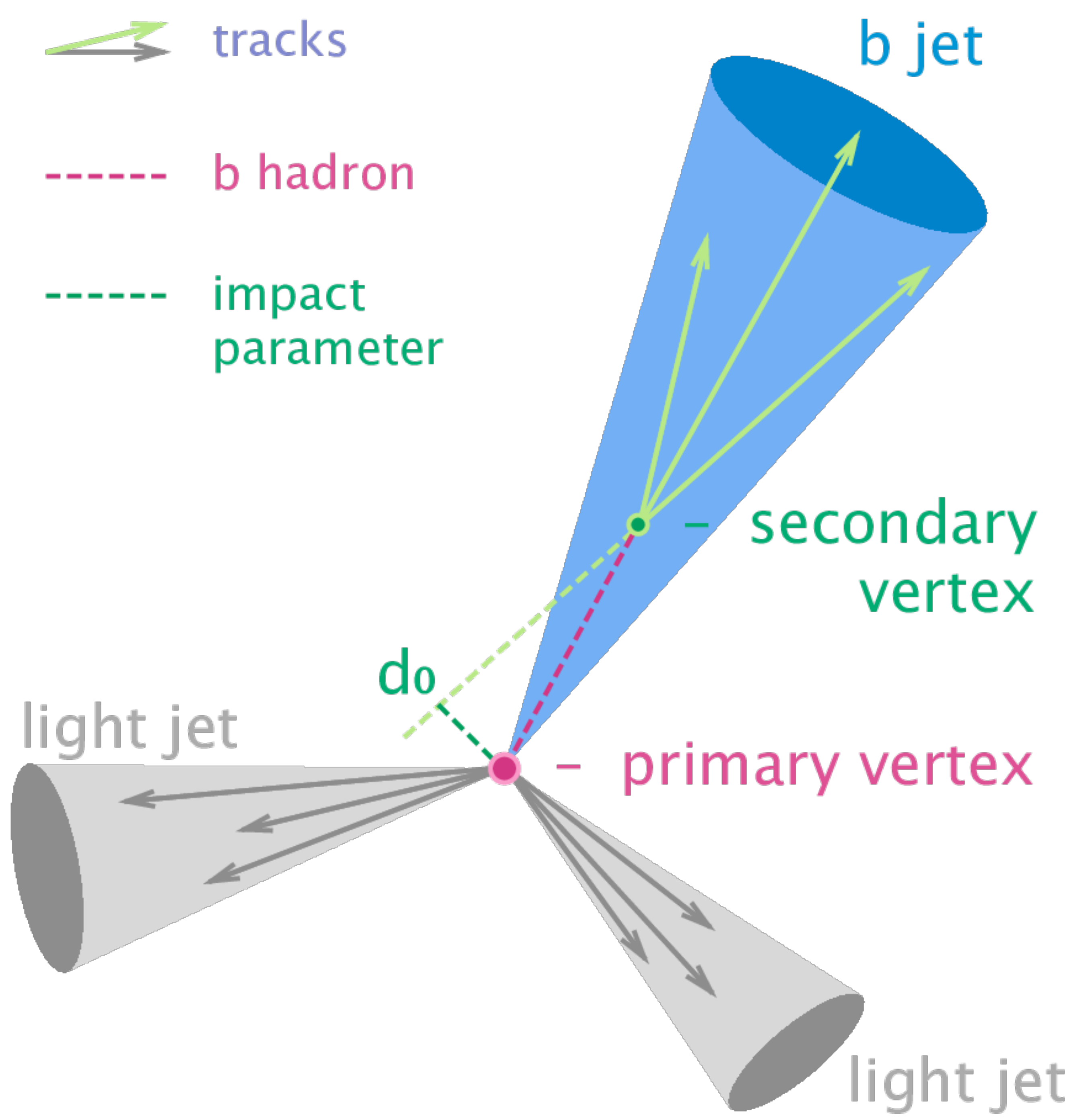
arXiv:2005.01842 [hep-ph]
Submitted to the JINST

Why heavy-flavour jets?

Heavy flavor jets are an important observable for many physics studies

Experimentally they are distinguished from light/gluon jets by:

- They are more collimated than their light counterparts
- Presence of the secondary vertex due to the decay of heavy flavor hadron

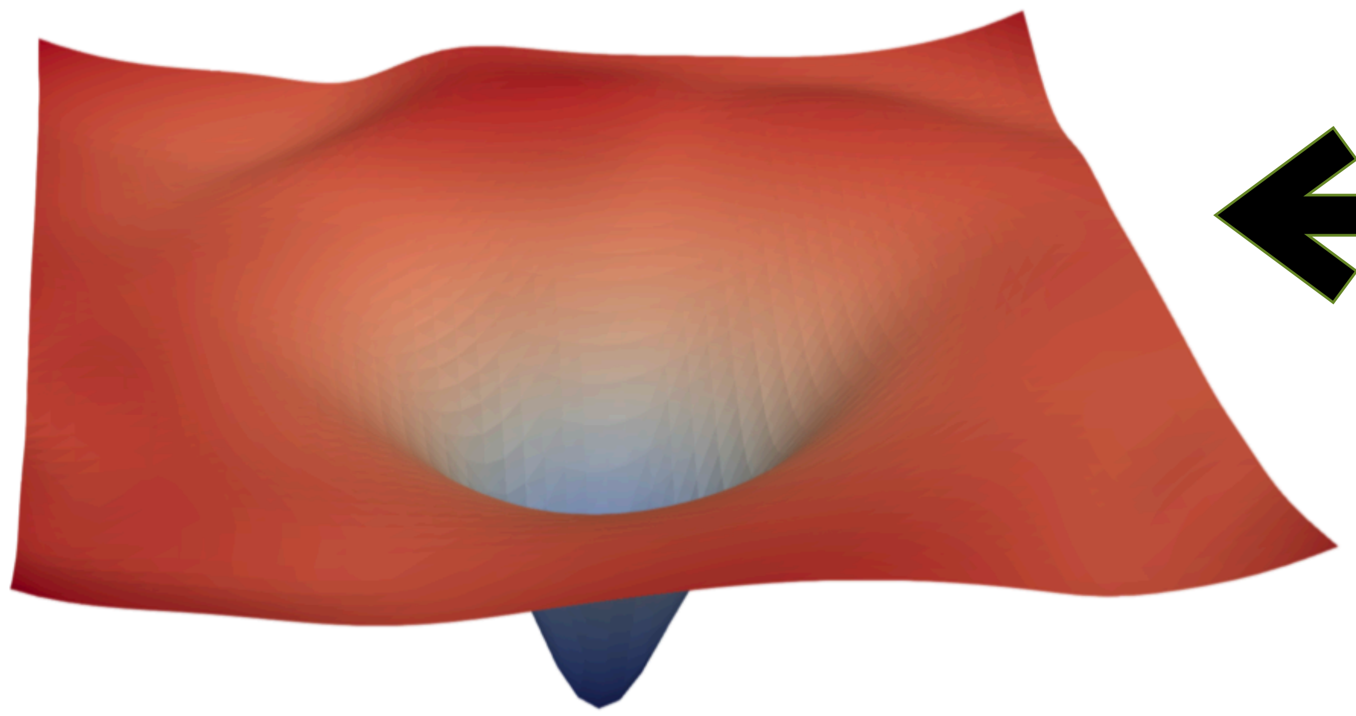


Why Machine Learning?

To solve jet classification task Machine Learning can be used

- It is an established way to solve mutli-dimensional problems
- Supervised machine learning
  - Learn functional mapping $f: \mathcal{X} \rightarrow \mathcal{Y}$ from given dataset
  - Select functional prior - Linear Model, SVM, Neural Network…
  - Look for best parametrization of chosen model
  - Train (i.e. minimize) some criterion - loss function

$$\theta^* = \underset{\theta \in \mathcal{P}}{\operatorname{argmin}} L(f(x;\theta), y)$$



An example of the low-dimensional parameter landscape.

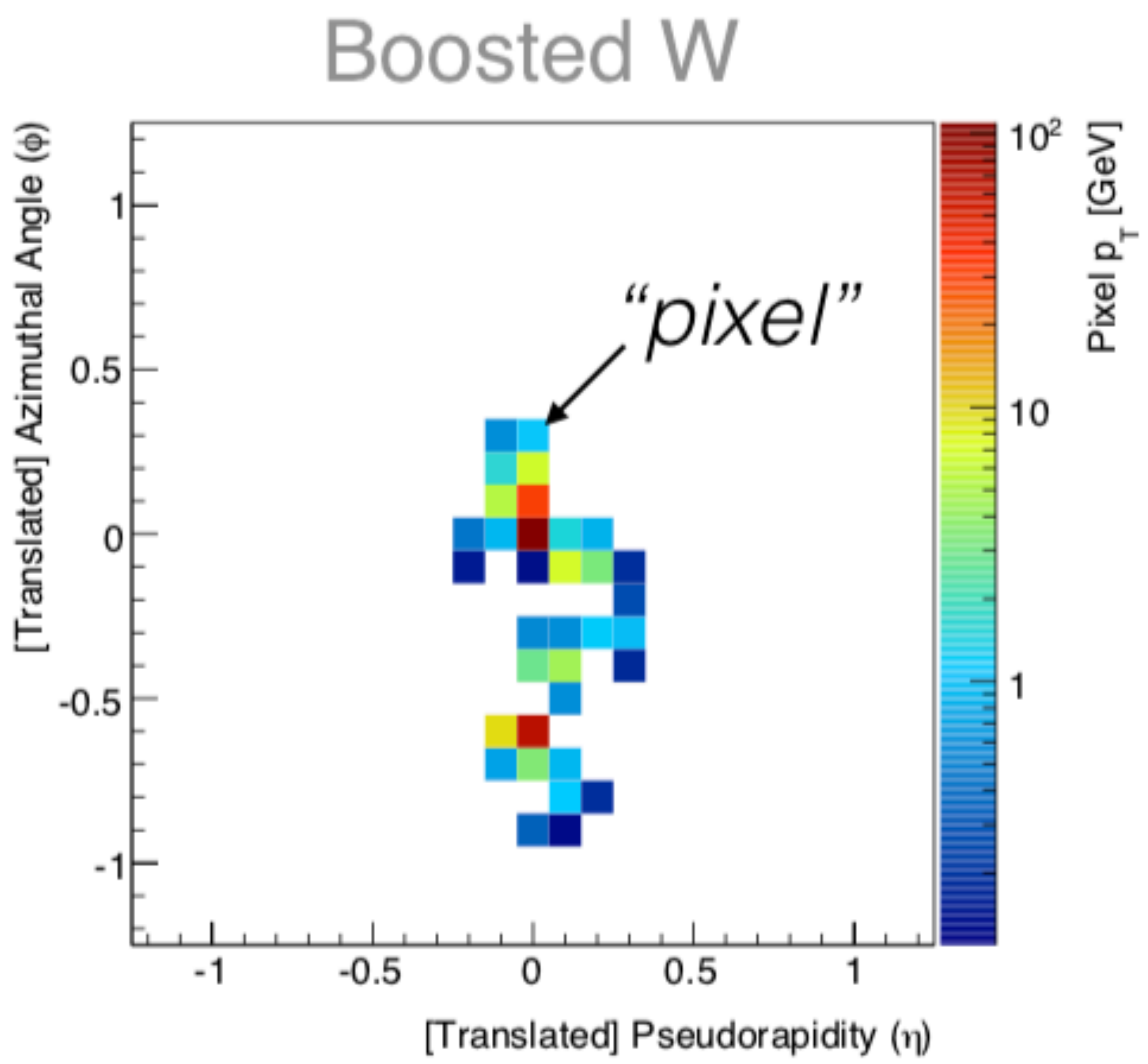
State of the Machine Learning based *c*/*b*-jet tagging

Previous research in ML-based jet tagging was mostly about jet images

- Tag jets initiated by *t* quark, *W* boson etc. using only (η, φ, p_T)
  - Heavy-flavor jet tagging requires more information
  - No simple way to unequivocally assign it in the image

Hence once should use a different approach

- Jet as **a sequence** of particles
  - Popular approach – sorting by p_T or vertex distance
- But there is another way - a **set of particles**



Credit: Benjamin Nachman

Rethinking jet tagging

What is a jet?

- Event – a set of particle state vectors

$$\mathcal{E} = \{\mathbf{r}_i | i \in \{1, \dots, n\}, \mathbf{r}_i = (p_i^{\mu}, v_x, v_y, v_z, \dots)\}$$

- Jet – a subset of event identified by the clustering algorithm
- Take a **set** of tracks as an input to the tagging algorithm
- Approach that can help us with that - **NetVLAD:**
  - **For each set it generates a fixed-sized vector** that characterizes it

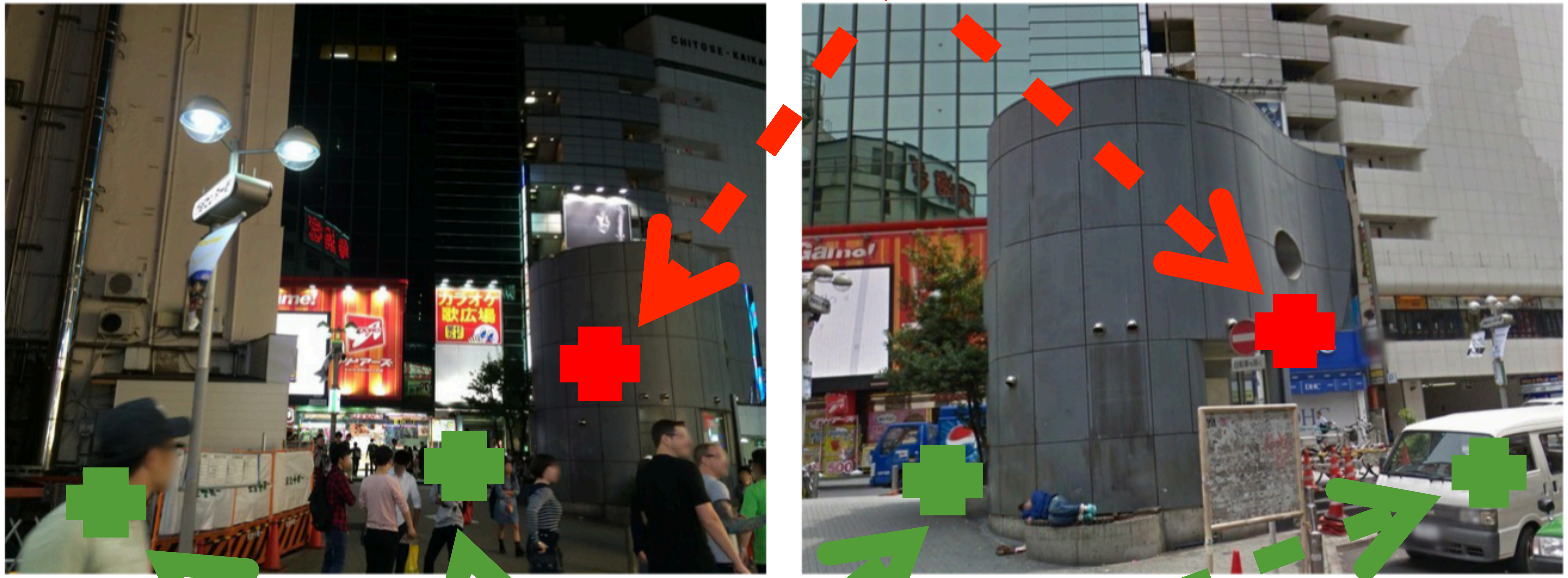
NetVLAD: CNN architecture for weakly supervised place recognition

Relja Arandjelović, INRIA *
Petr Gronat, INRIA*
Akihiko Torii, Tokyo Tech †
Tomas Pajdla, CTU in Prague ‡
Josef Sivic, INRIA*

IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 6, pp. 1437-1451, 1 June 2018.

Place Localisation

Place of interest



(a) Mobile phone query (b) Retrieved image of same place

Variable number of other objects

Rethinking jet tagging

Particle descriptors?

- In computer vision input is low level - we need a feature extractor before NetVLAD
- In jet physics – all measured variables are already high level
- Thus our state vectors can be treated as descriptors

Dataset generation

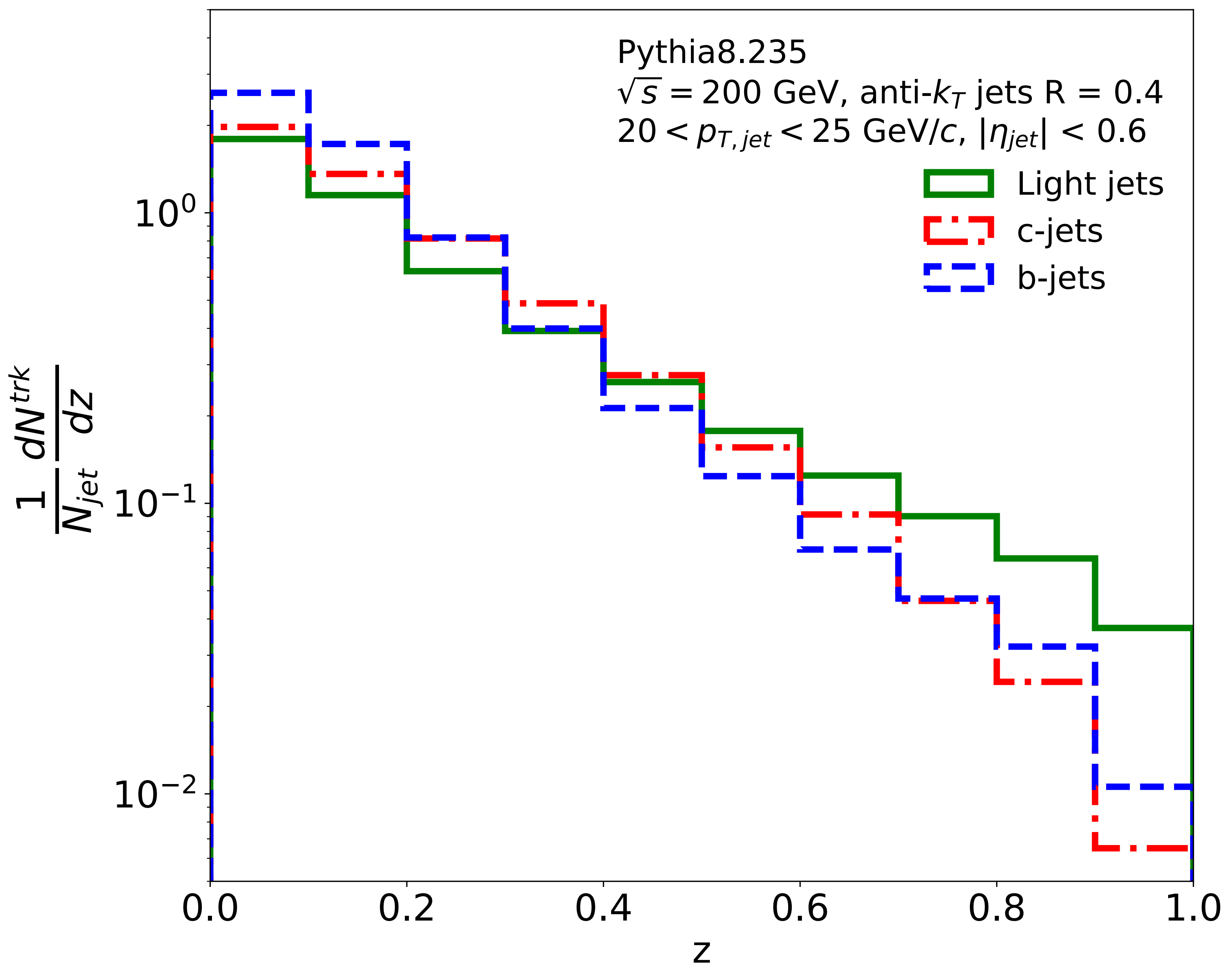
Pythia 8.235 is used to generate data

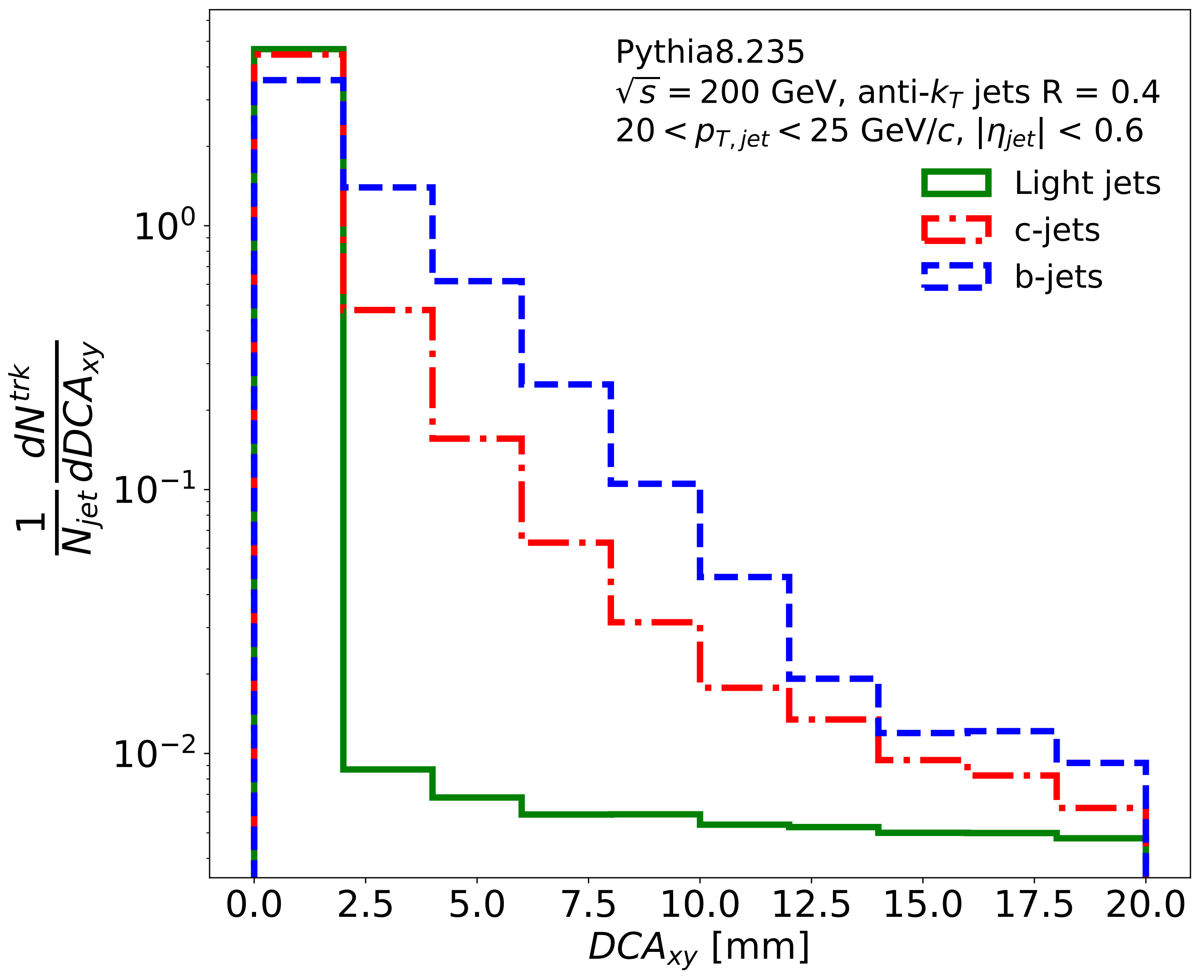
- 2 datasets are generated:
  - Weighted – ”HardQCD” that respects realistic jet flavor ratio
  - Balanced/Uniform - 50% light, 25% c-jet and 25% b-jet
- Separate dataset into 2 classes - light vs HF jets - better suited for RHIC physics
- The fast-sim approach is used to simulate finite resolutions:
  - Gaussian smearing of p_T is used in order to account for finite TPC resolution
  - Resolution of the STAR HFT is used to smear vertex information

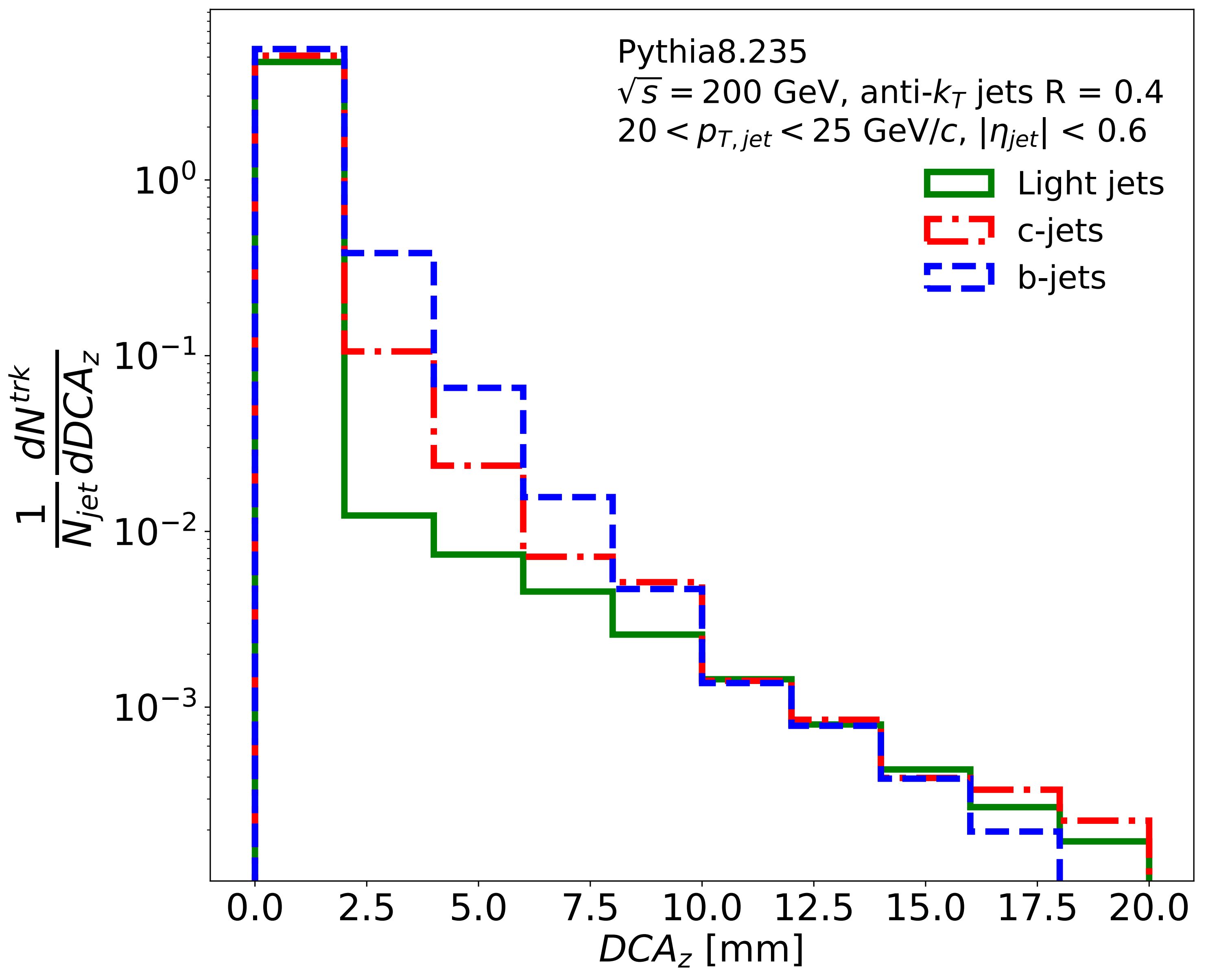
The following input variables are used:

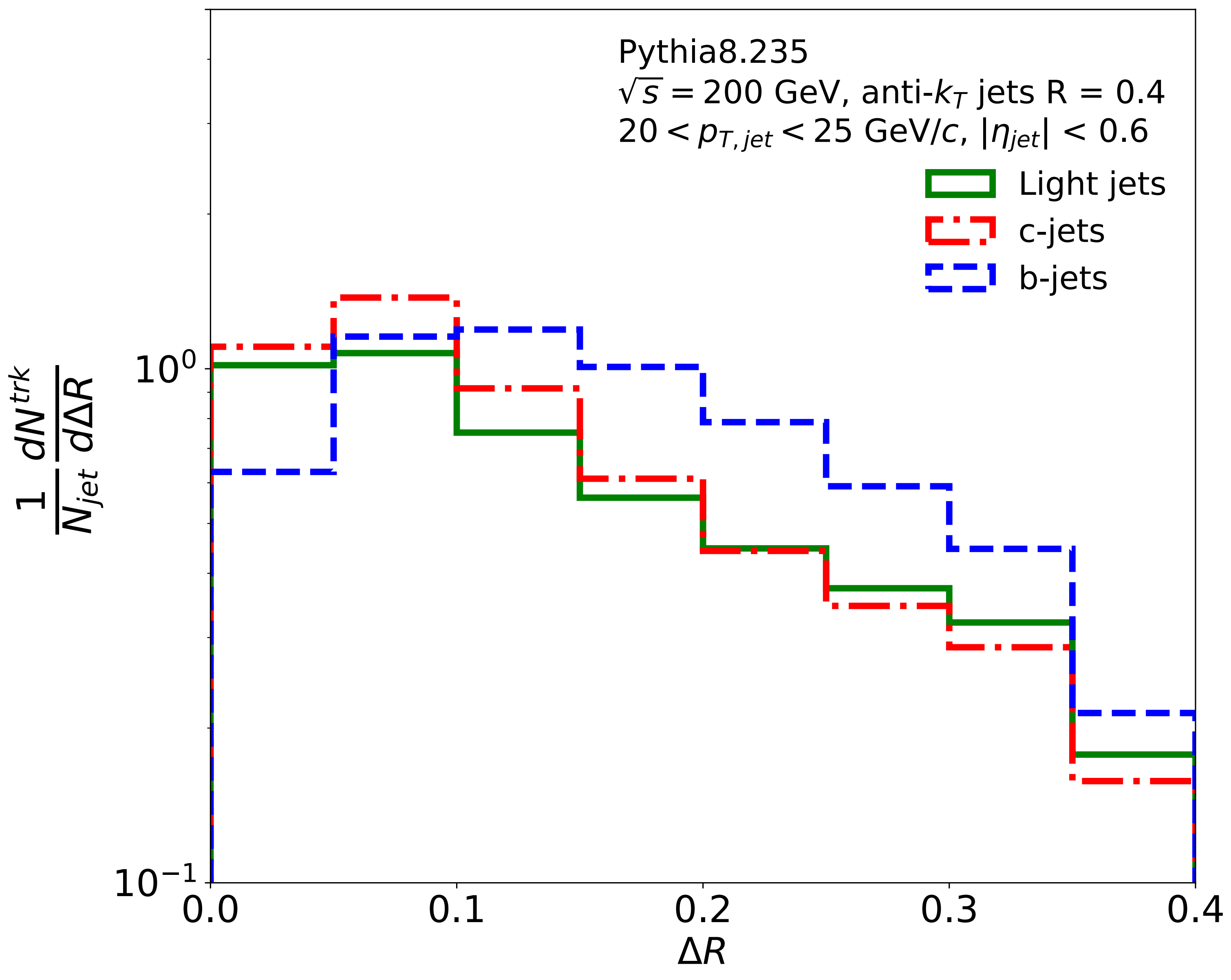
- Track p_T, η, φ
- $\mathrm{DCA_{xy}}$ and $\mathrm{DCA_z}$ of the track (distance of the closest approach to primary vertex)
- $z = \frac{p_{T,track}}{p_{T,jet}}$, $\Delta R(\mathrm{track, jet})$ and $z(\Delta R)^2$ - being track momentum fraction, distance to jet axis and jet mass fraction

Input Feature Distributions for 20-25 GeV/*c* Jets





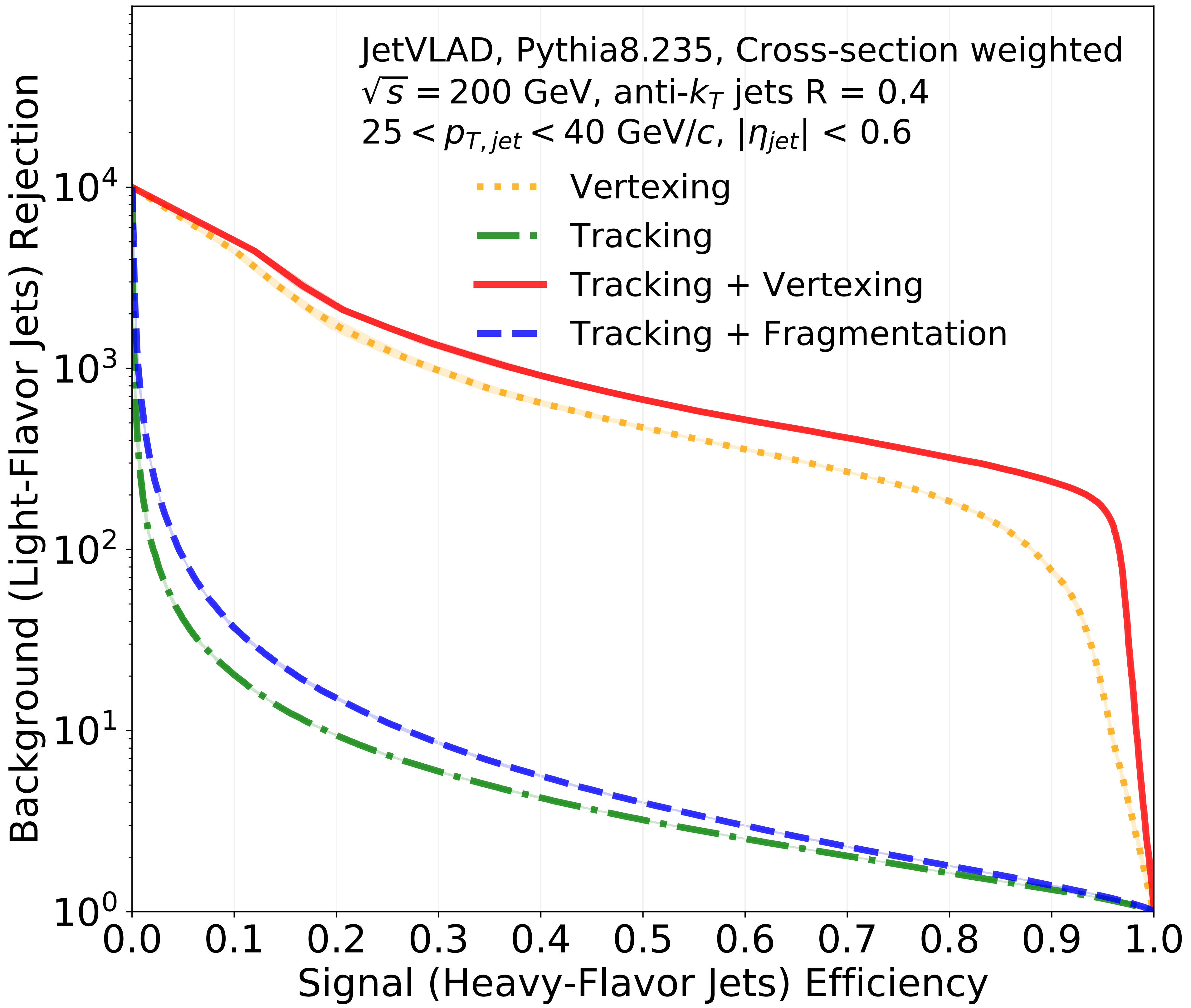


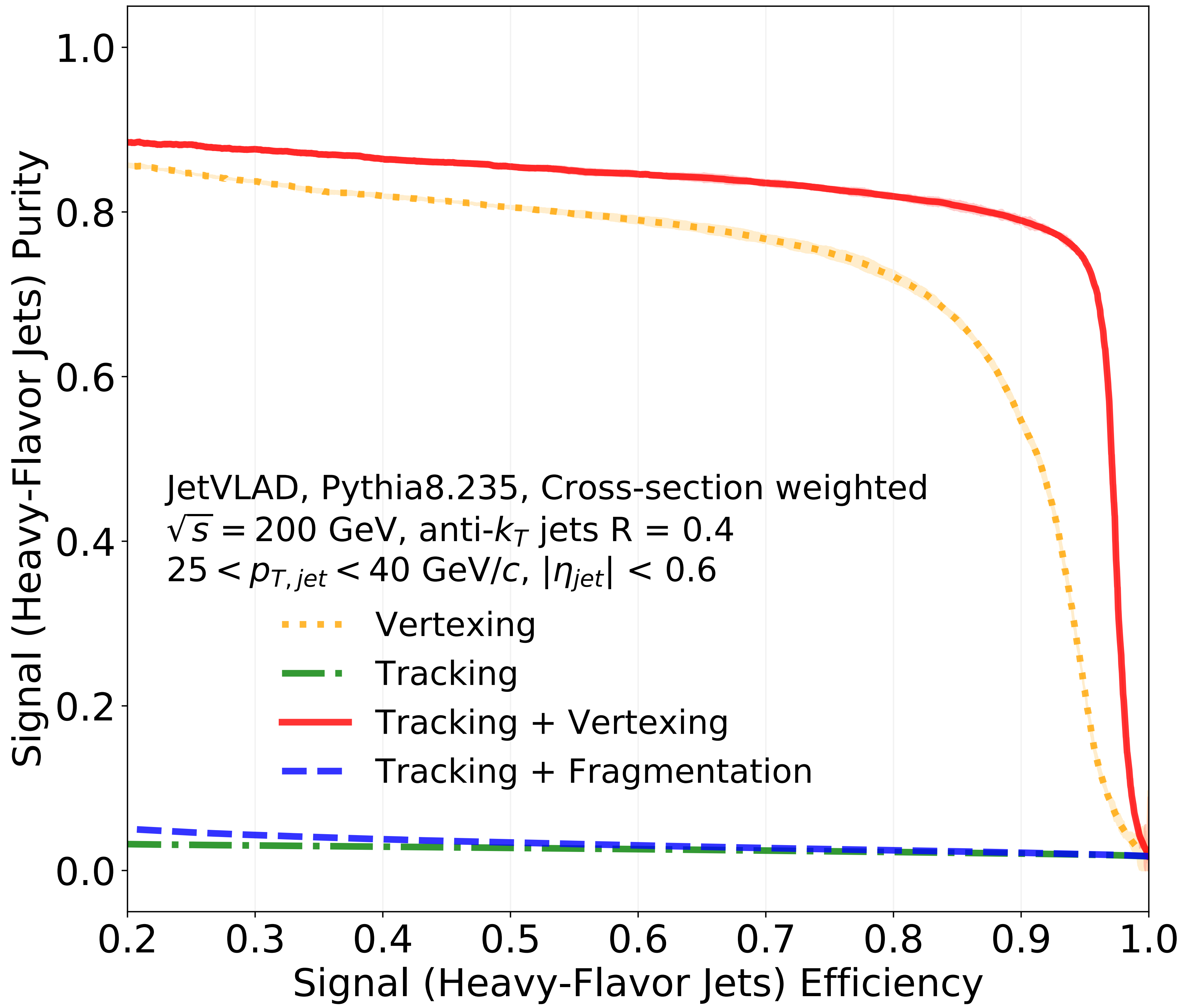


Tagger input variables

The following tagger versions are constructed:

- Vertexing - (DCA_{xy}, DCA_z)
- Tracking - (p_T, η, φ)
- Tracking + Fragmentation - $(p_T, \eta, \varphi, z, \Delta R, z(\Delta R)^2)$
- **Tracking + Vertexing - $(\boldsymbol{p_T, \eta, \varphi, DCA_{xy}, DCA_z})$ – the optimal choice**





Metrics

Name in Physics	Name in ML	Definition
Efficiency	True Positive Rate/Recall	$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$
Misid. Probability	False Positive Rate	$FPR = \frac{FP}{N} = \frac{FP}{FP + TN}$
Rejection	----------------------------	$Rej = \frac{1}{FPR}$
Purity	Precision	$PREC = \frac{TP}{TP + FP}$

Jet p_T dependent rejection and purity op

Efficiency	Purity	Rejection
80%	99%	268
50%	99%	579
Efficiency	**Purity**	**Rejection**
80%	99%	366
50%	99%	740

Unweighted/Balanced

jets in 5-10 GeV/c

jets in 25-40 GeV/c

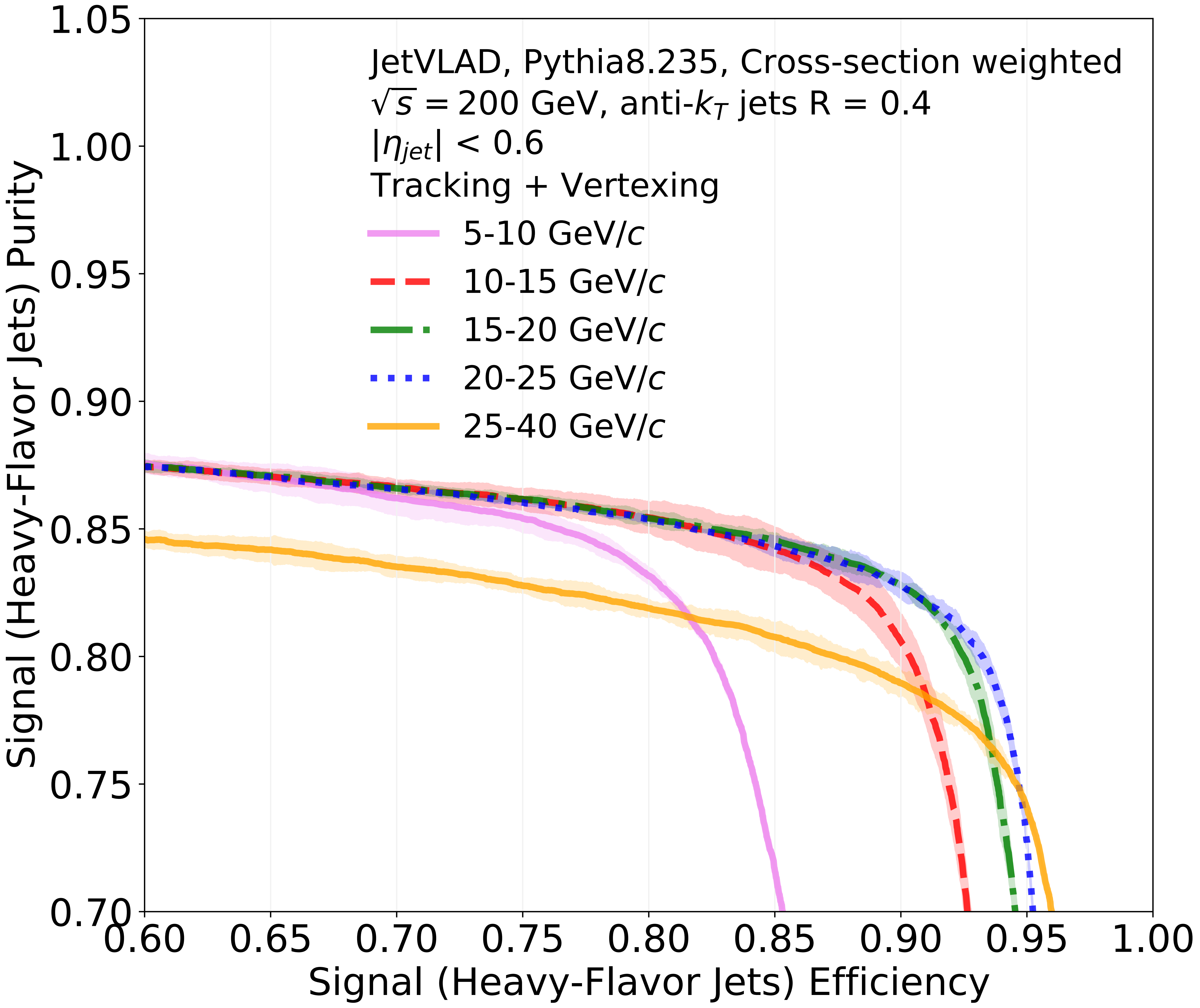
Efficiency	Purity	Rejection
80%	83%	223
50%	88%	540
Efficiency	**Purity**	**Rejection**
80%	81%	322
50%	85%	677

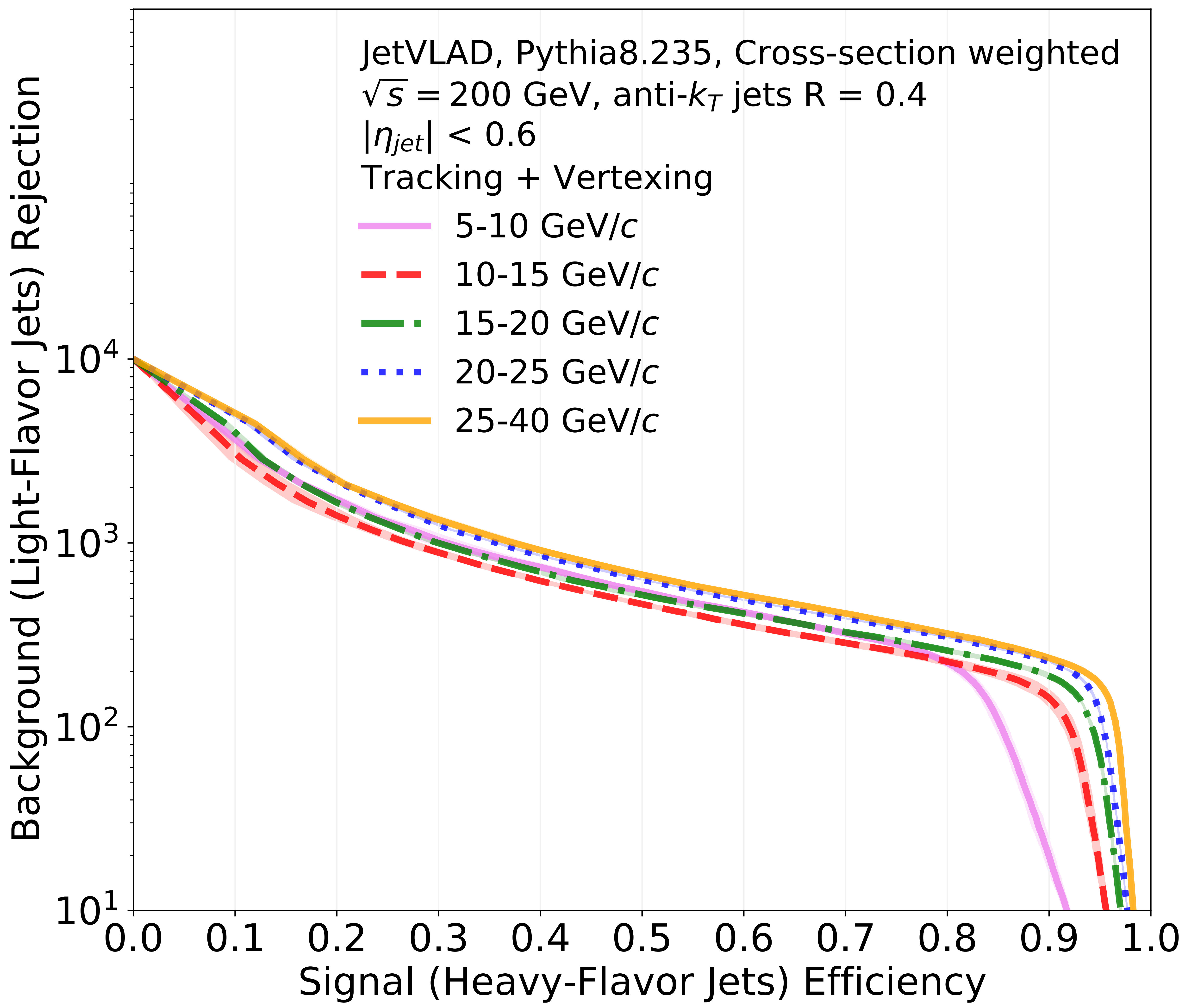
Weighted/HardQCD

The algorithm achieves good performance across different p_T ranges

- Excellent performance for low-p_T as well as high-p_T jets

Jet p_T dependent rejection and purity graphs





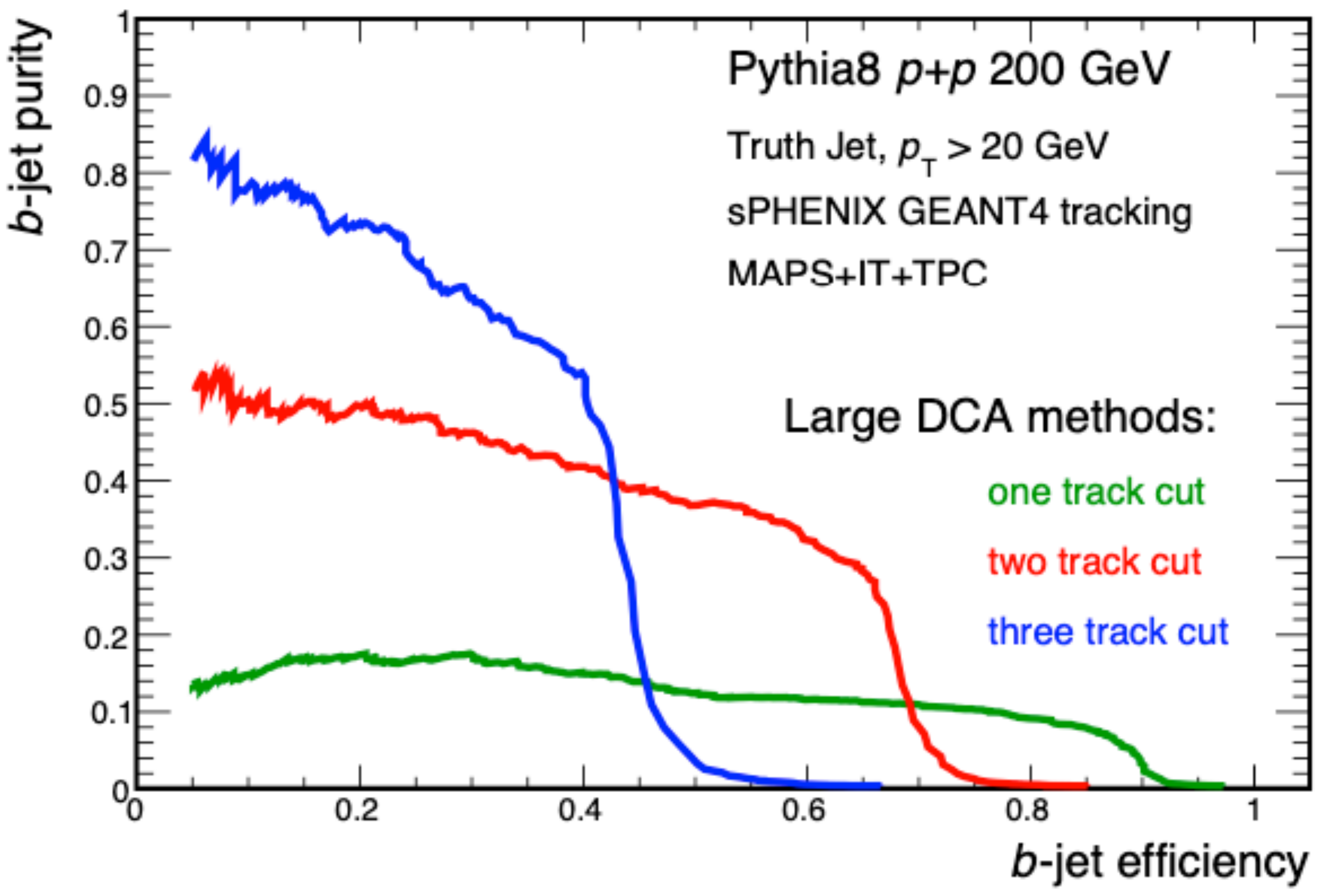
Conclusions

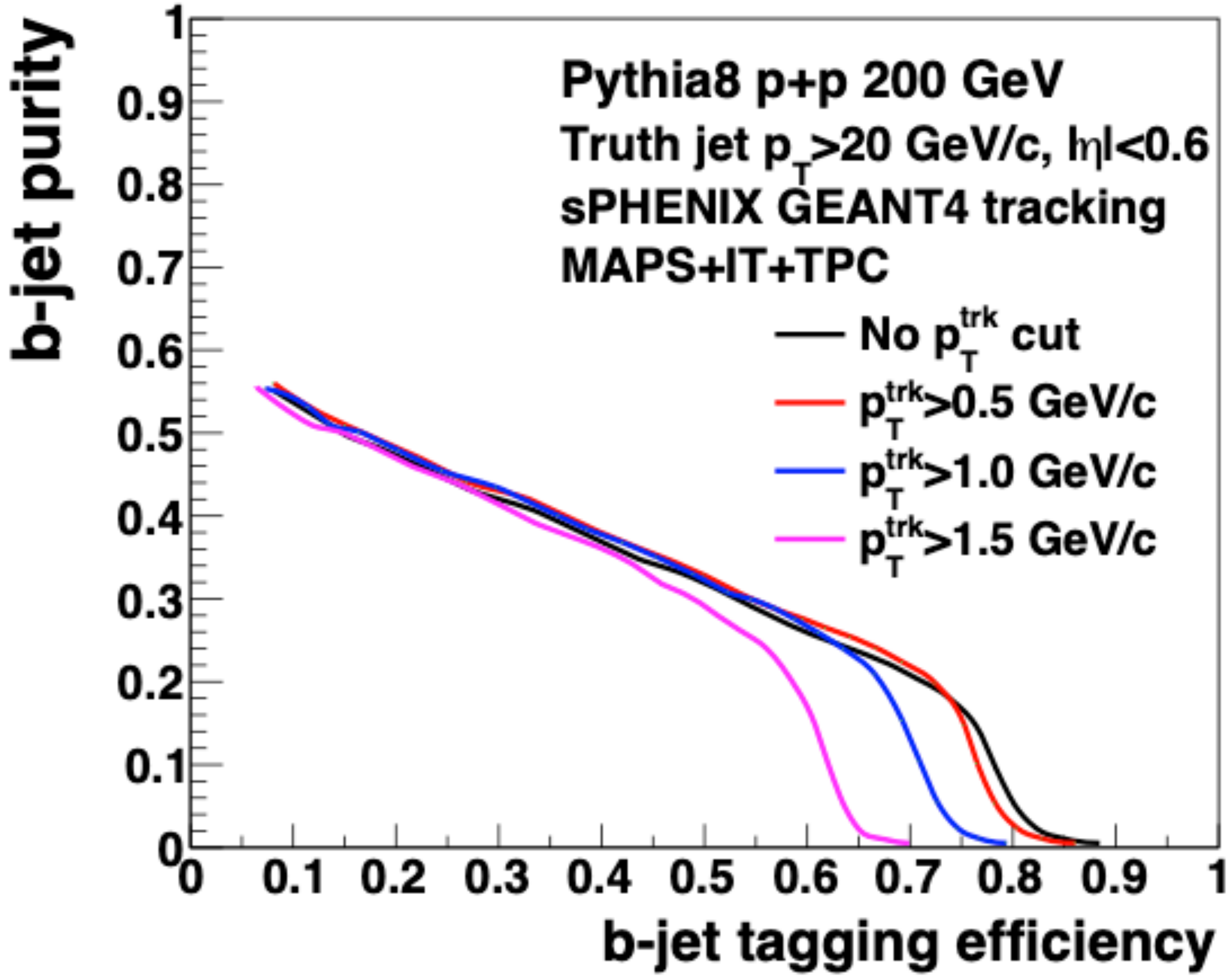
- We propose a novel set-based tagging methods based on the NetVLAD layer
- The model allows to identify heavy-flavor jets up to the low-p_T regime
  - Purity of 83%, Efficiency of 80% and rejection factor of ~220 is achievable
  - **Posibility to look for signatures of heavy-flavor jet radiation patterns at low p_T**
- Performance is dependent on the resolution of the hardware
  - **Next generation trackers (sPHENIX mVTX) should provide even better performance**

Acknowledgments

We would like to acknowledge Dennis Perepelitsa, Leticia Cunqeiro Mendez and Ming Liu for their helpful comments and suggestions and CIIRC IT department for providing us with HPC computational resources.

Backup: Classical methods



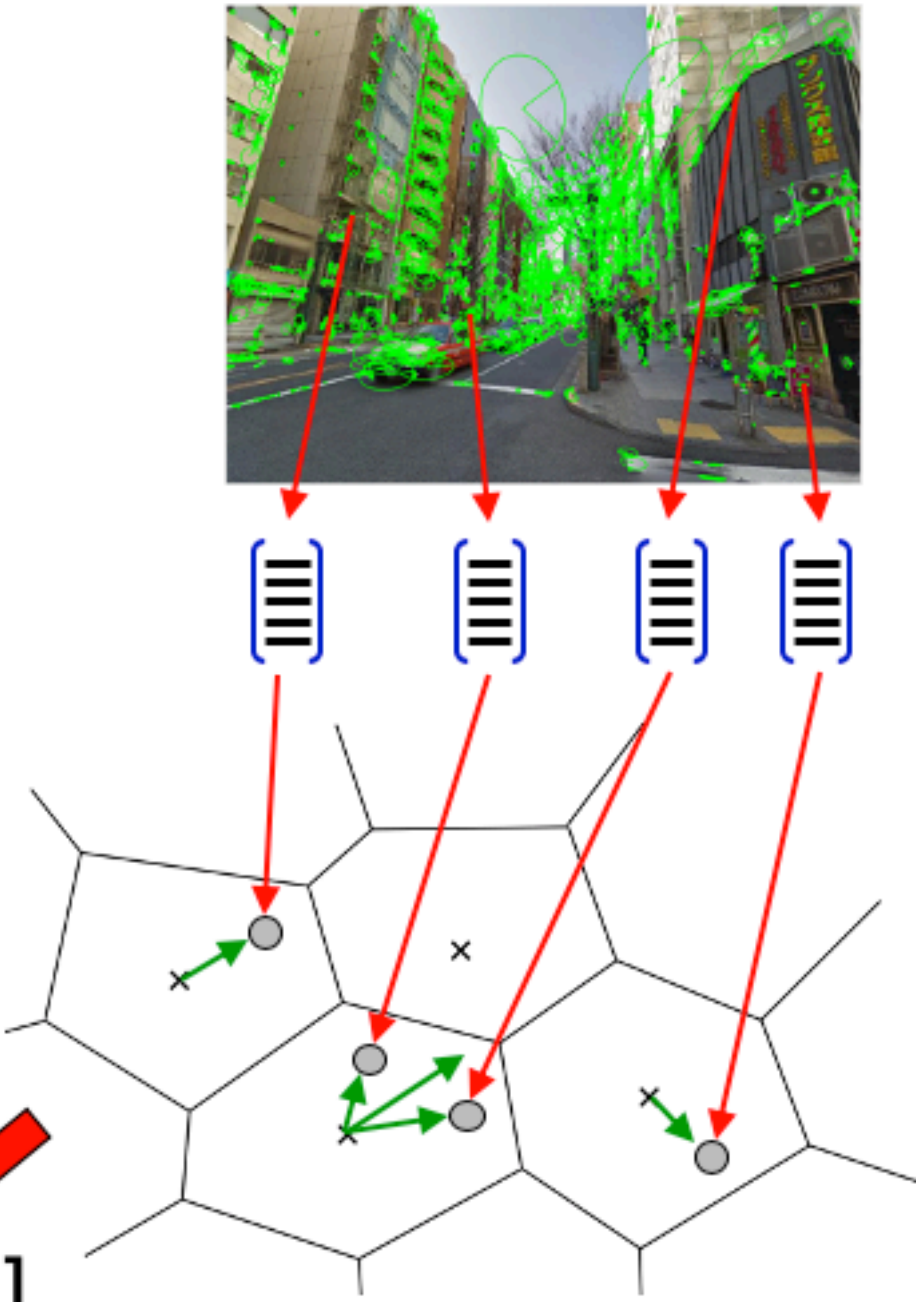


[“A monolithic active pixel sensor detector for the sPHENIX experiment.”]

Backup: NetVLAD principle

Review: Vector of Locally Aggregated Descriptors (VLAD)

$$V(:,k) = \underbrace{\sum_{i=1}^{N}}_{\text{Sum over all N descriptors in the image}} \overbrace{a_k(x_i)}^{\text{0/1 assignment of desc. } i \text{ to cluster } k} \underbrace{(x_i - c_k)}_{\text{Residual vector}}$$



V = [↗, . . , ↗, ↘, …]

[Jégou et al. 10]

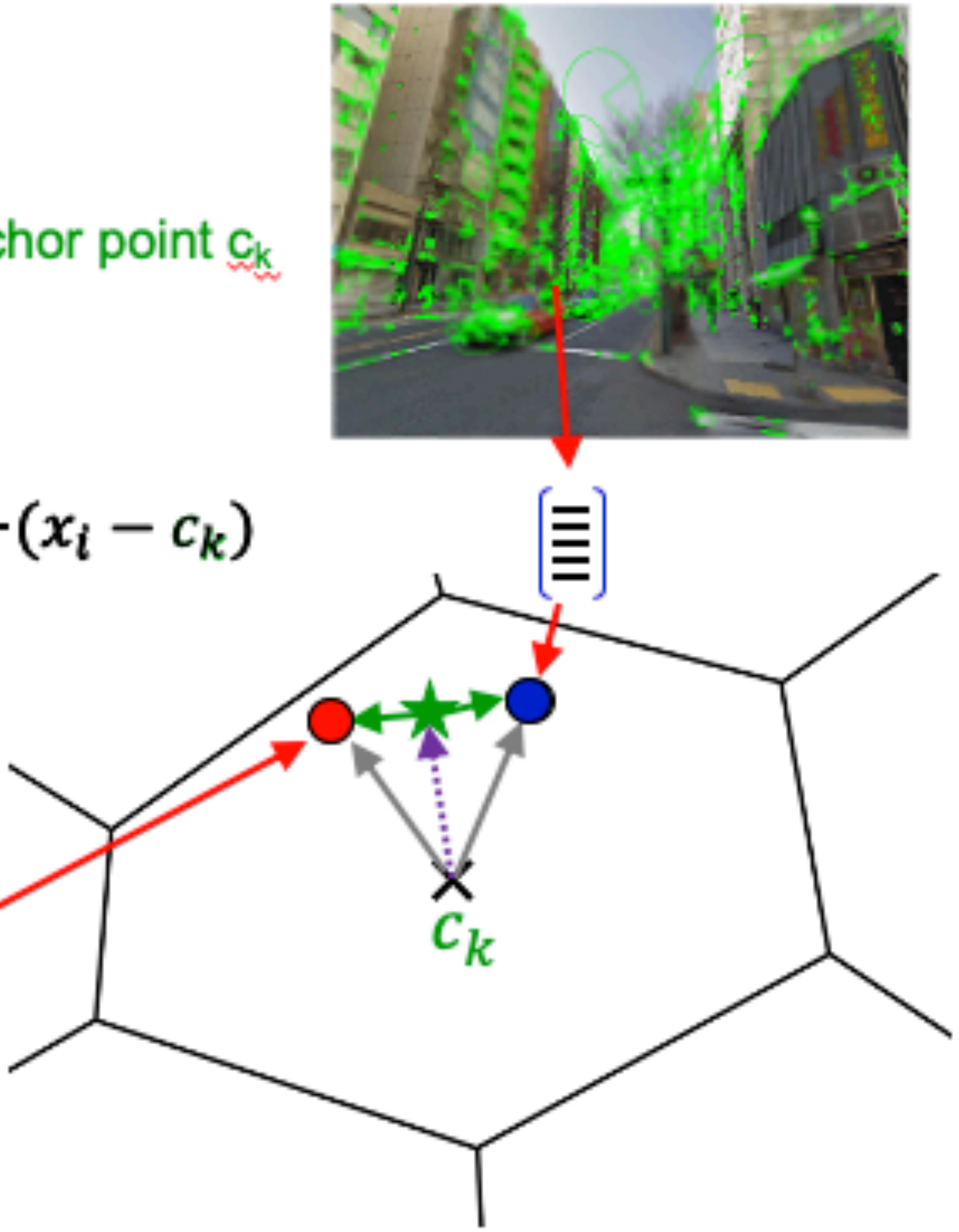
[Arandjelović et al. 16]

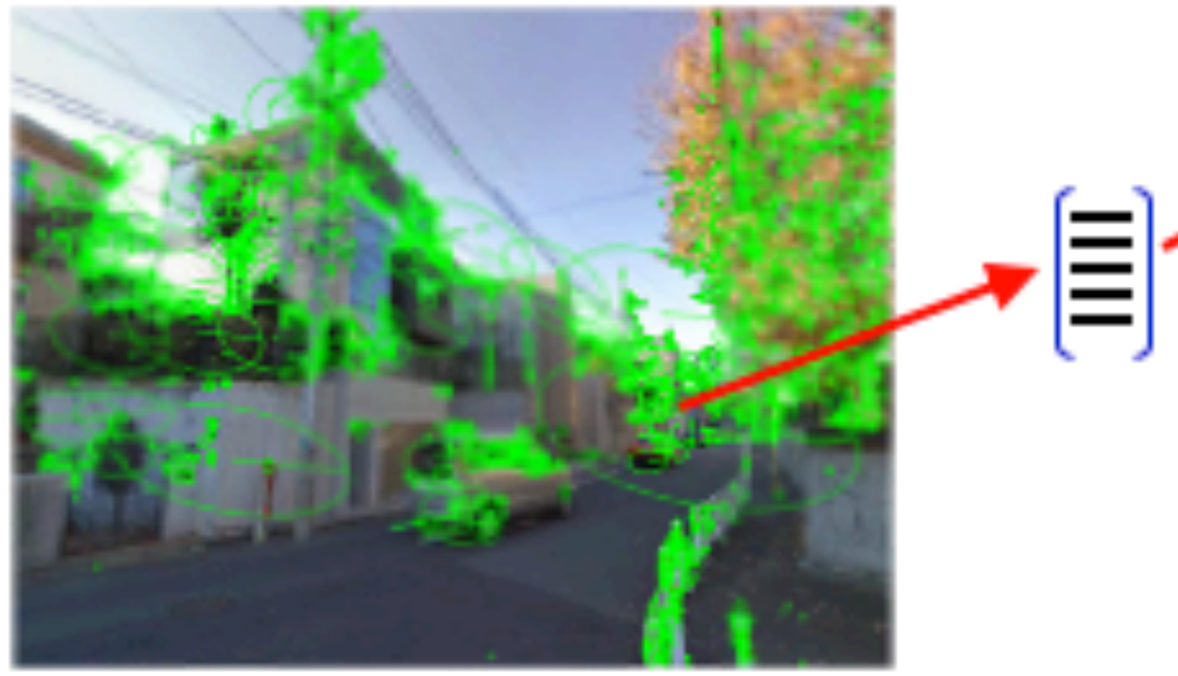
Backup: NetVLAD principle

NetVLAD: Trainable pooling layer

Decouple assignment (w_k, b_k) from anchor point c_k

$$V(:,k) = \sum_{i=1}^{N} \frac{e^{w_k^T x_i + b_k}}{\sum'_k e^{w_{k'}^T x_i + b_{k'}}} (x_i - c_k)$$





[Arandjelović et al. 16]

Backup: Training procedure and architecture

Training procedure

- SGD with $\eta = 0.013$ and cosine modulation with warm restart with $T_0 = 1, T_m = 3$
- Training is done for 2000 epochs at maximum
- Early stopping criterion is set for 20 epochs, looking for changes in validation loss

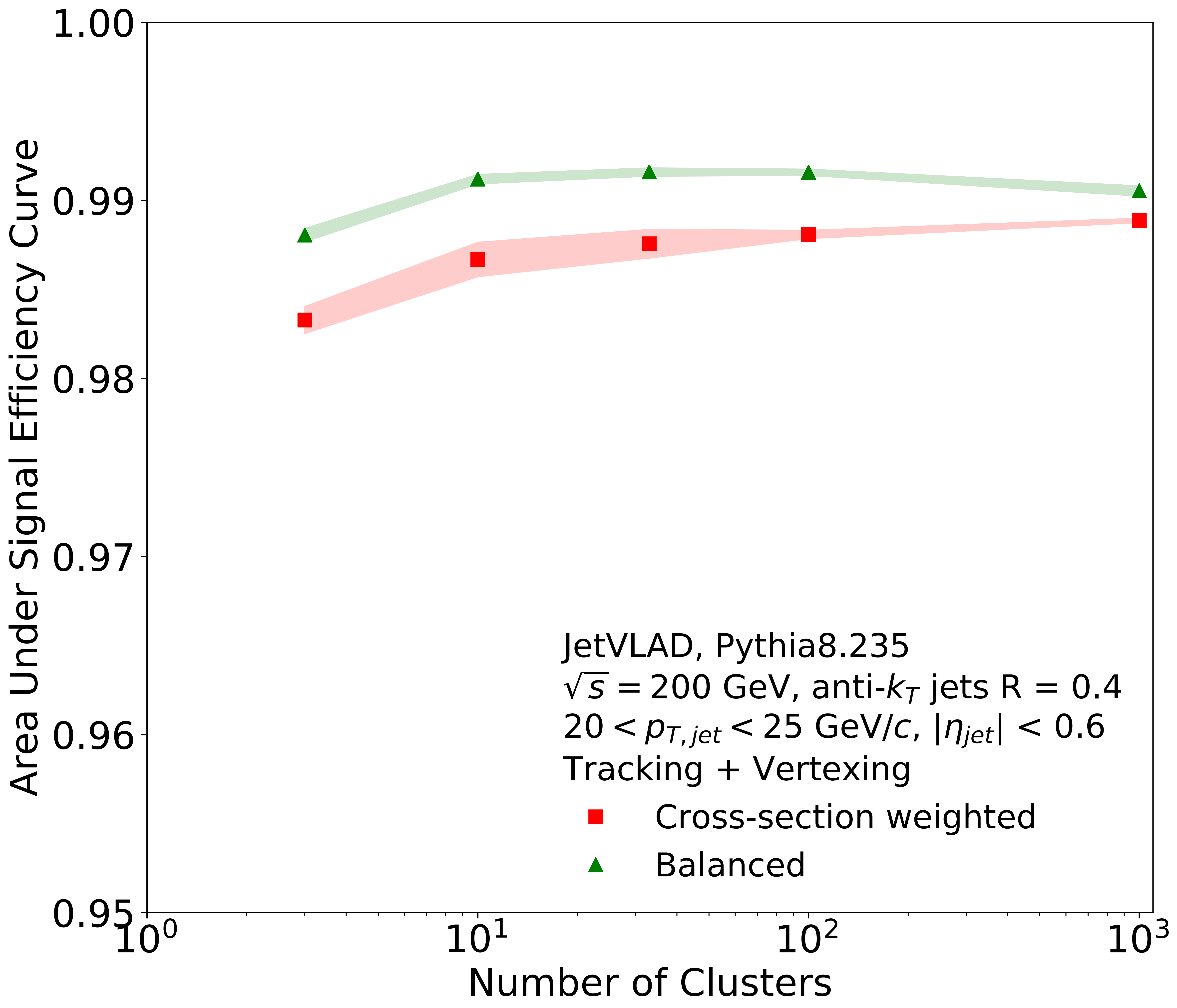
Model architecture

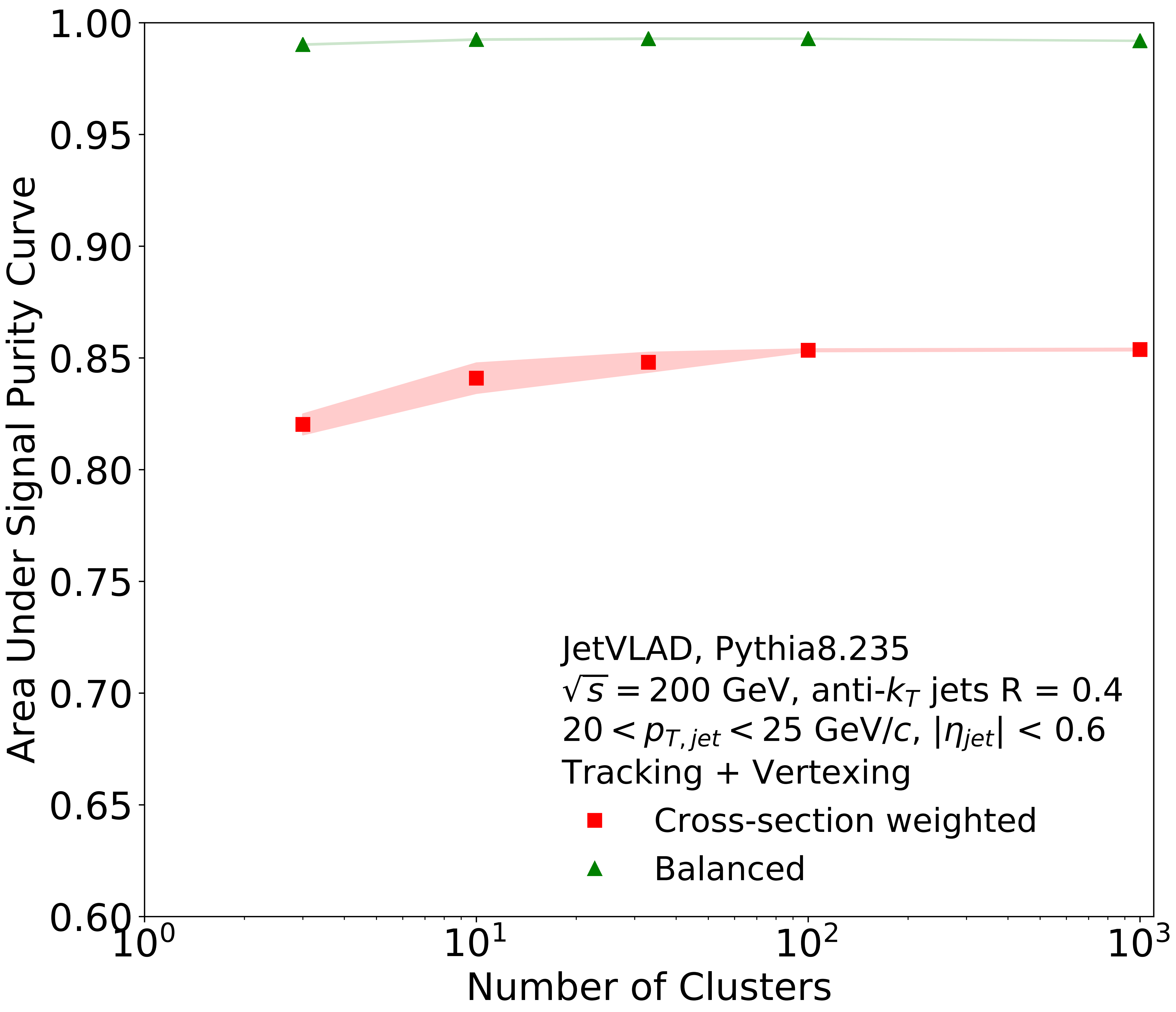
- Input is taken with NetVLAD layer
- Further we use Residual Blocks - Linear -> ReLU -> BN -> Linear -> Identity + ReLU
- Dropout for p = 0.5 is used to regularize model
- Random grid search was used for optimal hyperparameters

Backup: Hyperparameter sensitivity test

- We need to understand what are effects of DOF on performance
- This is done by varying depth and number of clusters (fixing one, varying another)
- We choose jets in 20-25 GeV/*c* bin because they are the middle ground

Backup: Cluster sensitivity test





Backup: Depth sensitivity test