Identifying Heavy-Flavor Jets Using Vectors of Locally Aggregated Descriptors

Georgy Ponimatkin[3,5,6]

Joint Work with **R. Kunnawalkam Elayavalli**[1,2], **J. H. Putschke**[2], **J. Bielčíková**[3,6] and **Josef Šivic**[4,5]

Based on **JINST 16 (2021) 03, P03017**

1 – Yale/BNL

2 – Wayne State University, Detroit, MI

3 – Nuclear Physics Institute, Czech Academy of Sciences

4 – Inria, Ecole Normale Superieure, CNRS, PSL Research University, Paris, France

5 – Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University

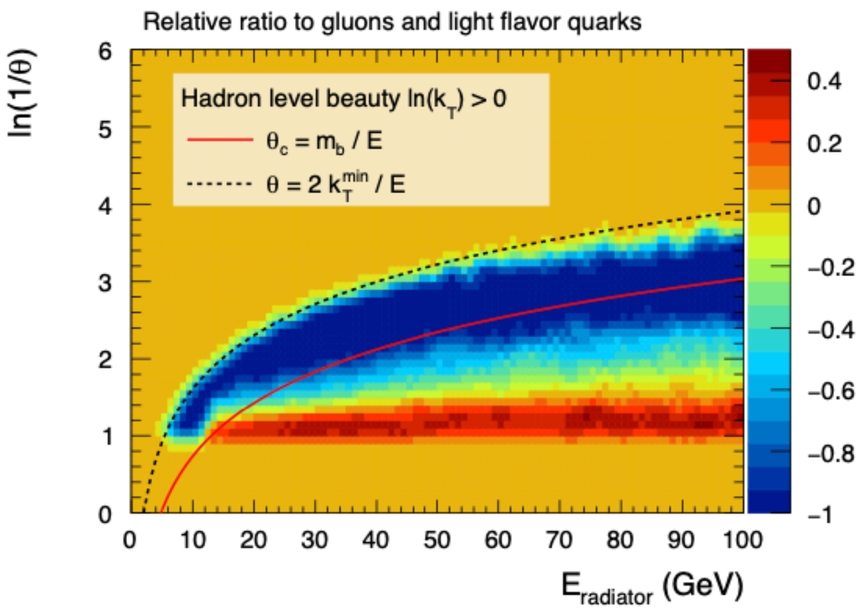
6 – Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University

Why Heavy-Flavor Jets?

Generally, heavy-flavor jets are an important class of observables in itself

Measuring jet substructure of low-p_T heavy-flavor jets is an exciting measurement in physics, since it can answer

- Change in radiation patterns of heavy-quarks (so-called dead-cone effect)
- Mass dependence of QCD splitting functions
- Mass dependence of jet energy loss in nuclear medium



L. Cunqueiro, M. Płoskoń, Phys.Rev.D 99 (2019) 7, 074027

Rethinking Heavy-Flavor 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 jet clustering algorithm
- Without assuming jet substructure, jet is a **set of tracks**
- And we wish to take a **set of tracks** as an input to the tagging algorithm
- In Computer Vision there is an approach that might help us – **NetVLAD**
  - **For each set it generates a fixed-size 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 Localization

Place of interest

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

Variable number of other objects

Rethinking Jet Tagging Paradigm

Why is NetVLAD good solution?

- It operates on so-called descriptors (high-level feature vectors coming from the last layer of CNN)
- It is resistant to noise (due to variable number of objects in place localization)

Thus we introduce particle descriptors

- In jet physics all measured variables are high level and we do not need feature extractor (i.e. CNN)
- Hence, we can assume that our state vectors are descriptors

Dataset Generation

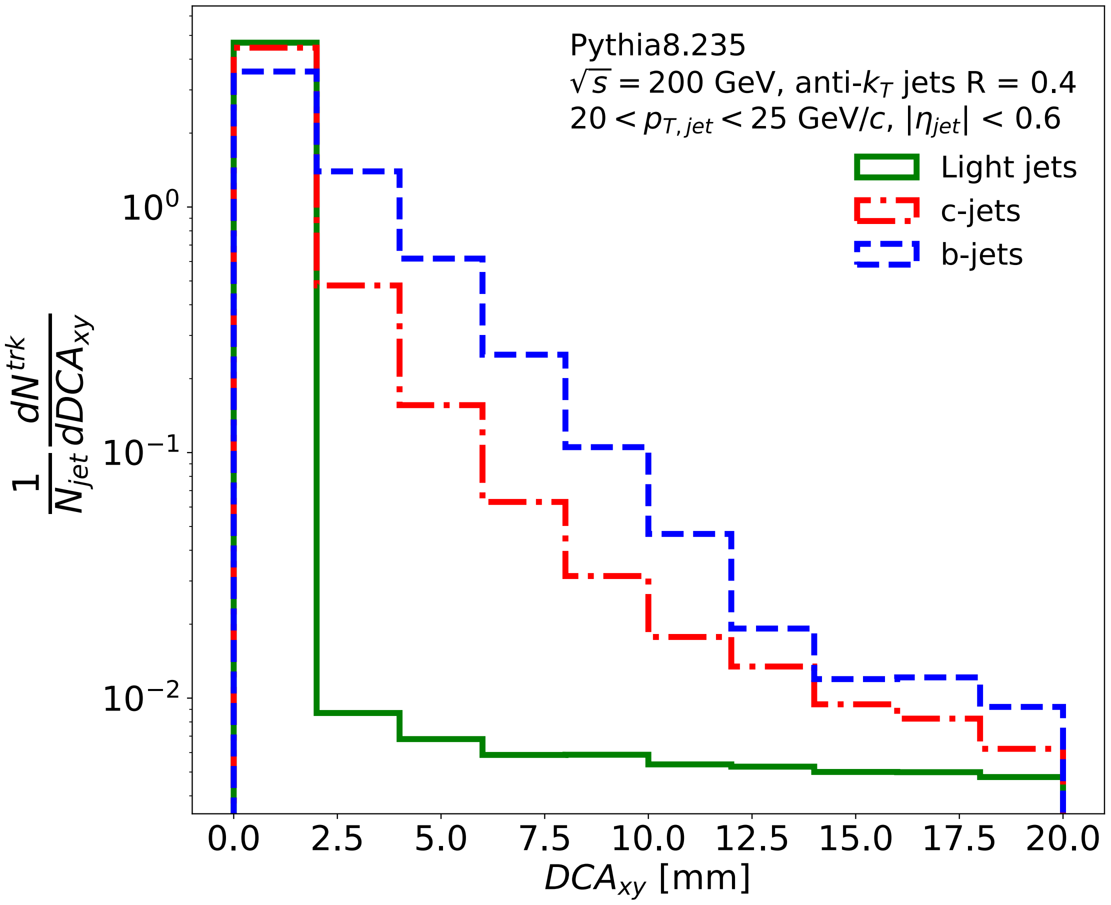
Pythia8.235 was used to generate data

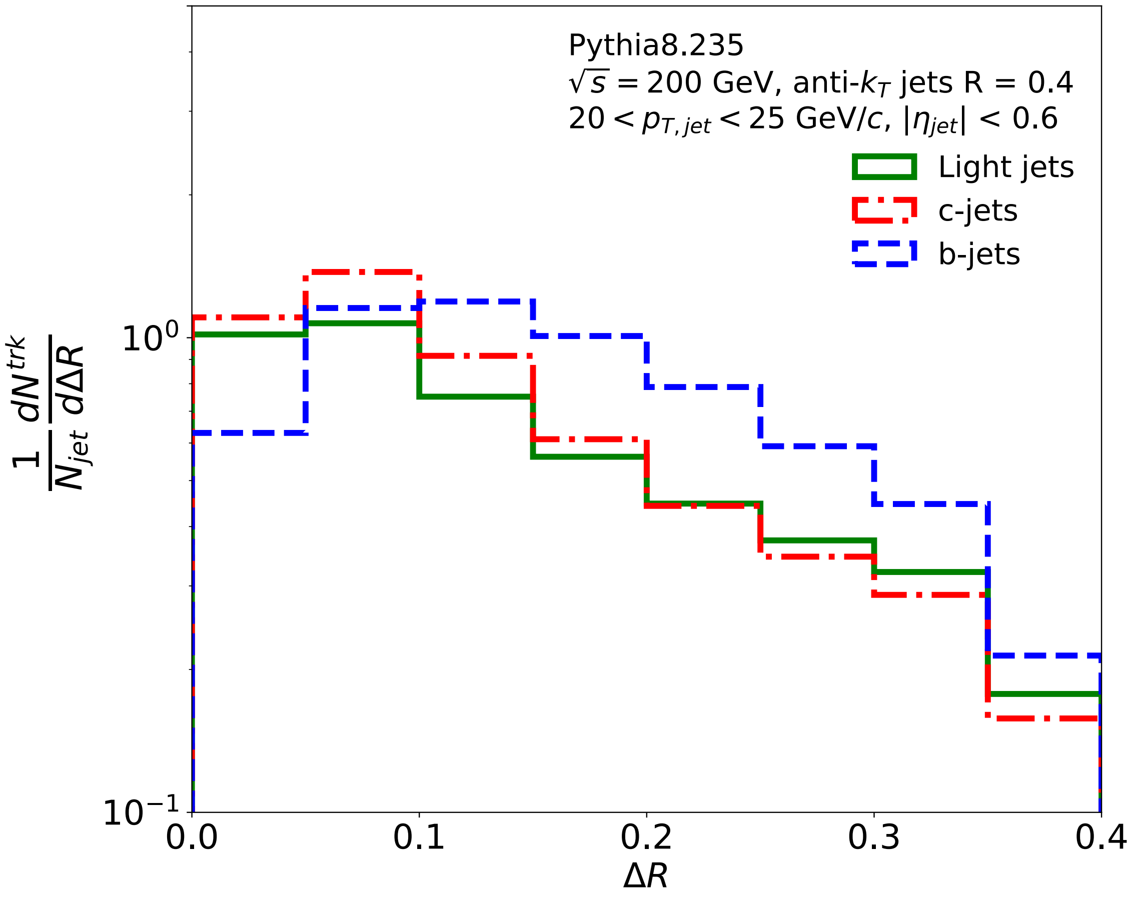
- 2 datasets are generated:
    - Weighted - that respects realistic jet flavor ratio
    - Balanced – 50% udsg-jet, 25% c-jet and 25% b-jet
- We separate datasets into two classes – udsg vs HF jets – which is better suited for RHIC physics
- The fast-sim approach is used to simulate finite resolutions:
    - 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

Input Features

Type	Inputs	Definition
Tracking	p_T	Transverse momentum in the $x - y$ plane
	η	pseudorapidity
	ϕ	azimuthal angle
Fragmentation	z	momentum fraction $\frac{p_T^{track}}{p_T^{jet}}$
	ΔR	distance between track and jet axis $\sqrt{\Delta\phi^2 + \Delta\eta^2}$
	$z\Delta R^2$	higher level feature
Secondary Vertex	DCA_{xy}	Distance of closest approach in $x - y$/
	DCA_z	Distance between primary and secondary vertex in the z axis

Input Feature Distributions for 20-25 GeV/c Jets





Model Architecture

Symbolically the whole model can be described as

$$\mathrm{JetVLAD} = \mathrm{NetVLAD}(N_C) \rightarrow D \times [\mathrm{ResidualBlock}] \rightarrow \mathrm{SoftMax}$$

- Whole network has only two free hyperparameters – number of NetVLAD clusters N_c and depth D
- We use fully-connected ResNet-style blocks, since they lead to faster optimization
- 50% drop-out is also applied to increase model generalization

We optimize model by SGD with cosine-annealed learning rate

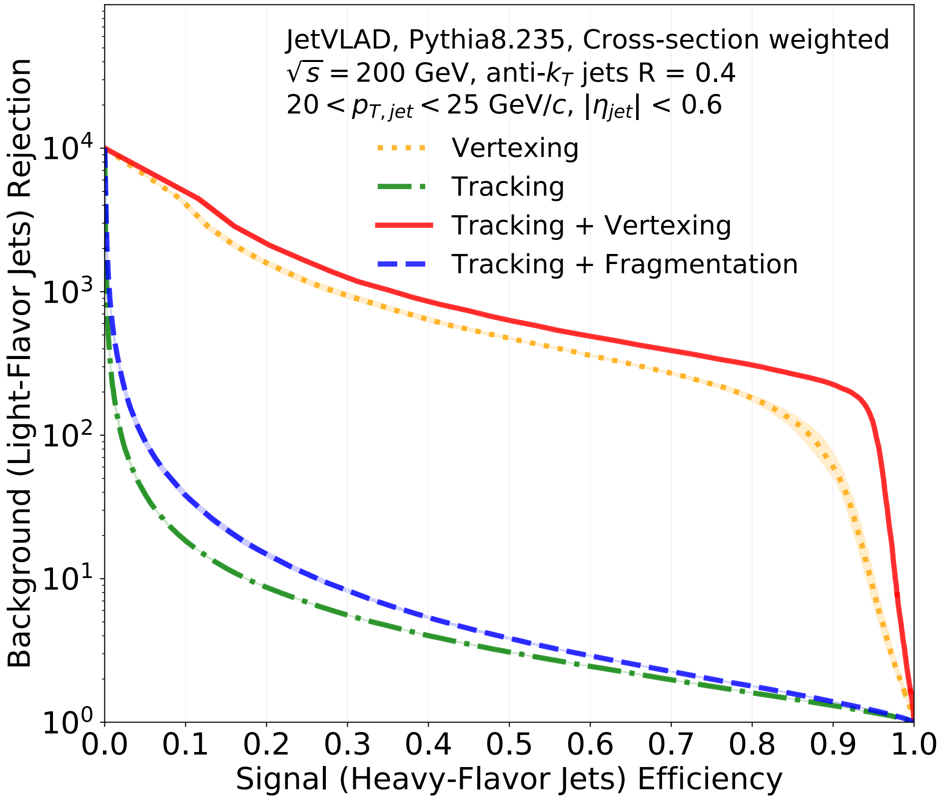
Performance Metrics

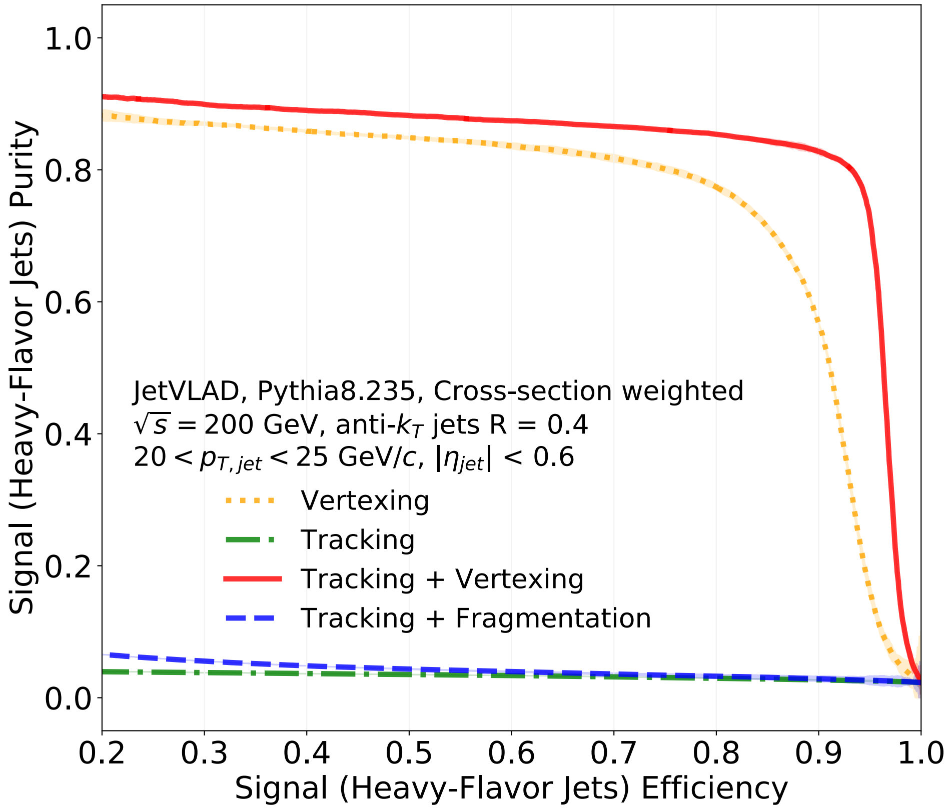
Physics	Machine Learning	Definition
Tagging Efficiency	True Positive Rate (TPR)/Recall	$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}}$
Misidentification Prob.	False Positive Rate (FPR)	$\mathrm{FPR} = \frac{FP}{N}$
Background Rejection		$\mathrm{REJ} = \frac{1}{\mathrm{FPR}}$
Signal Purity	Precision	$\mathrm{PREC} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$

Finding Optimal Input Variables

The following tagger versions are constructed:

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

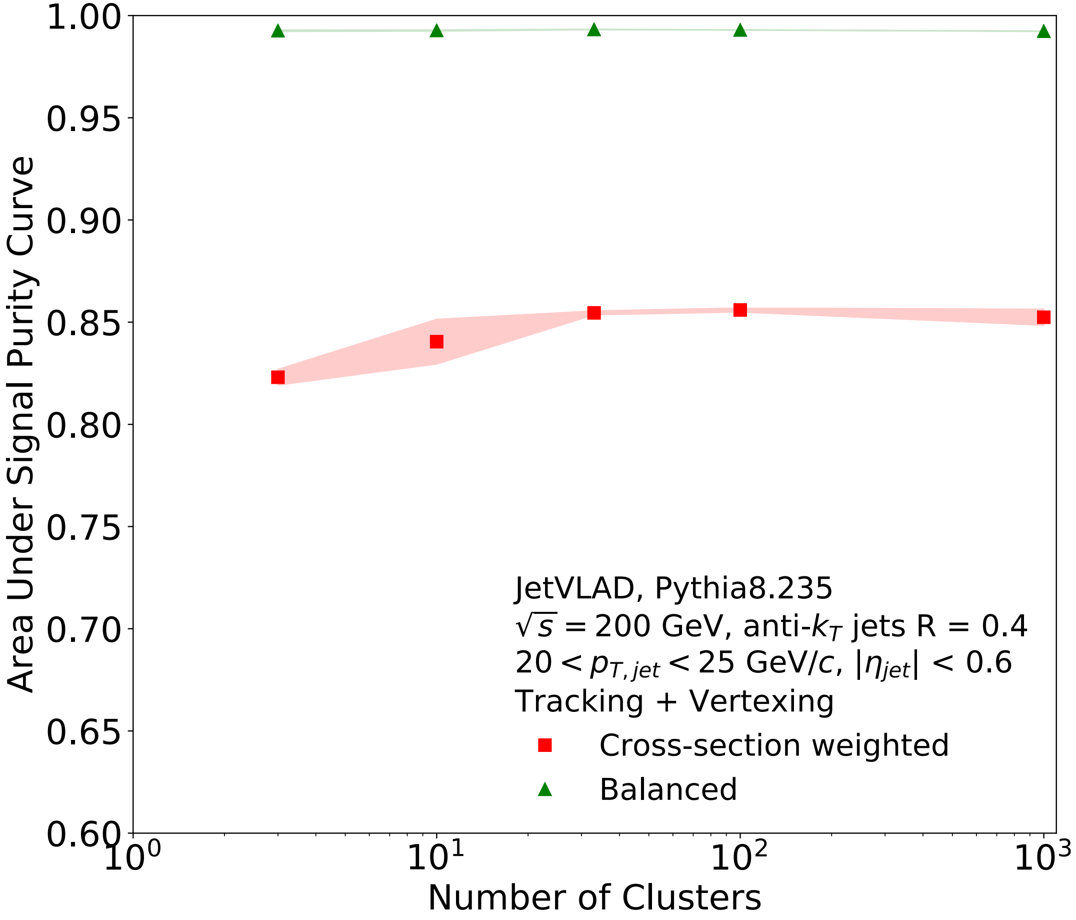


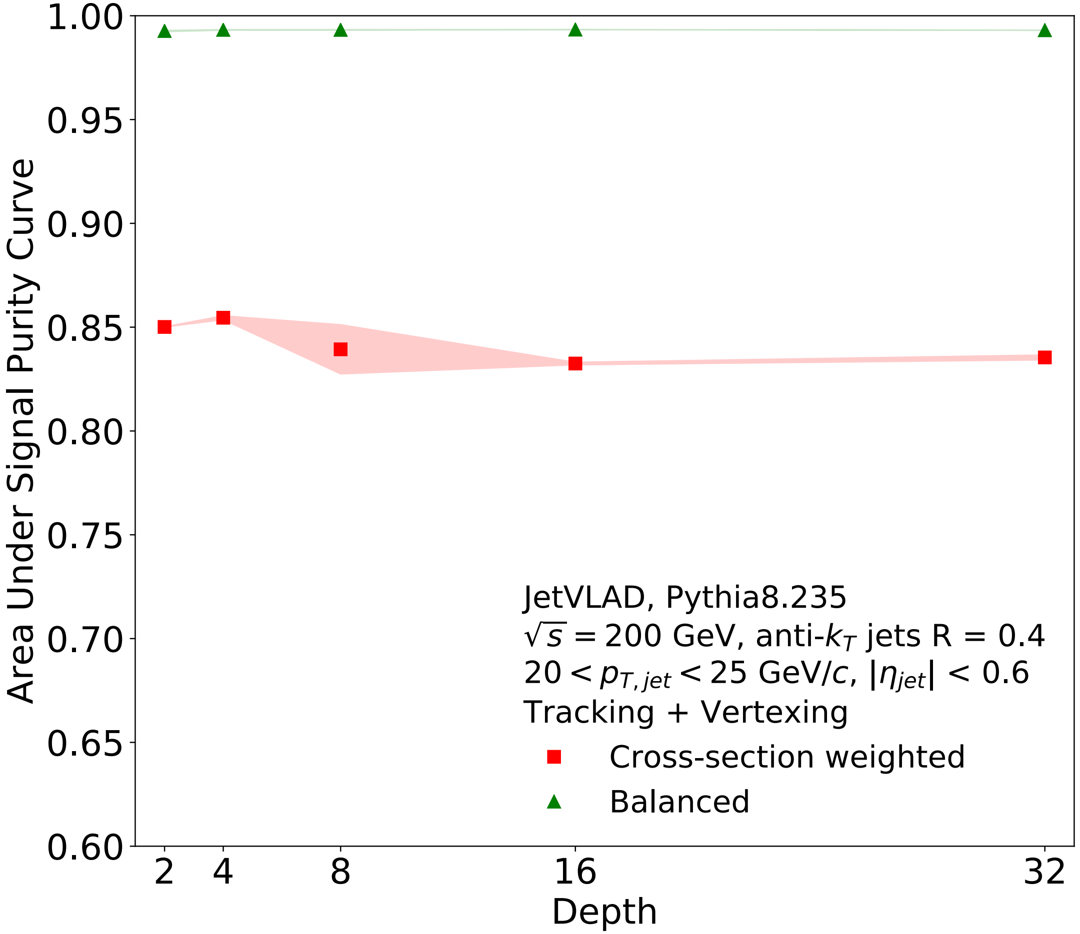


Finding Optimal Hyperparameters

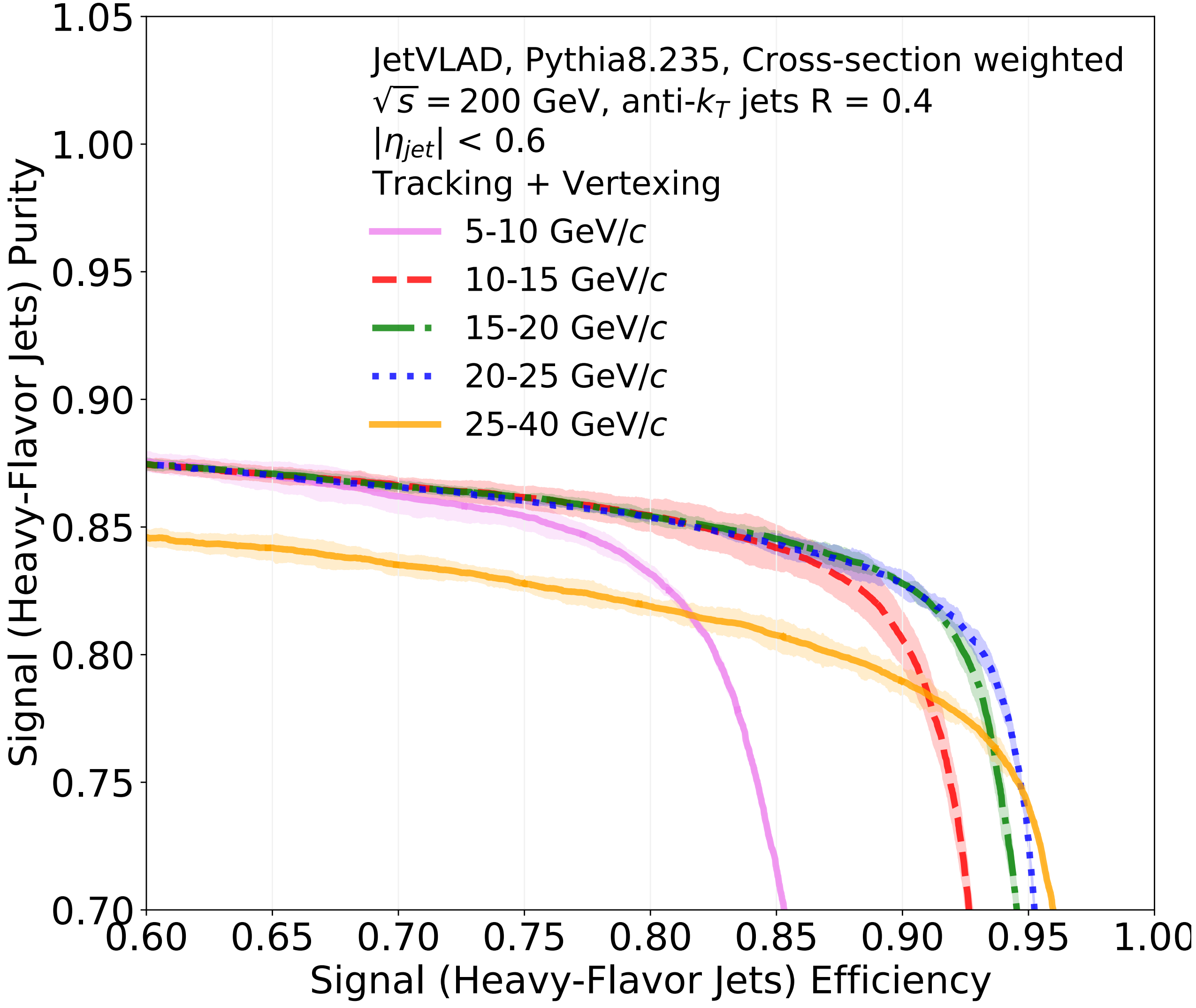
The optimal hyperparameters were found by running random grid search on depth D and number of clusters N_c and we obtain $N_c = 33$ and $D = 4$

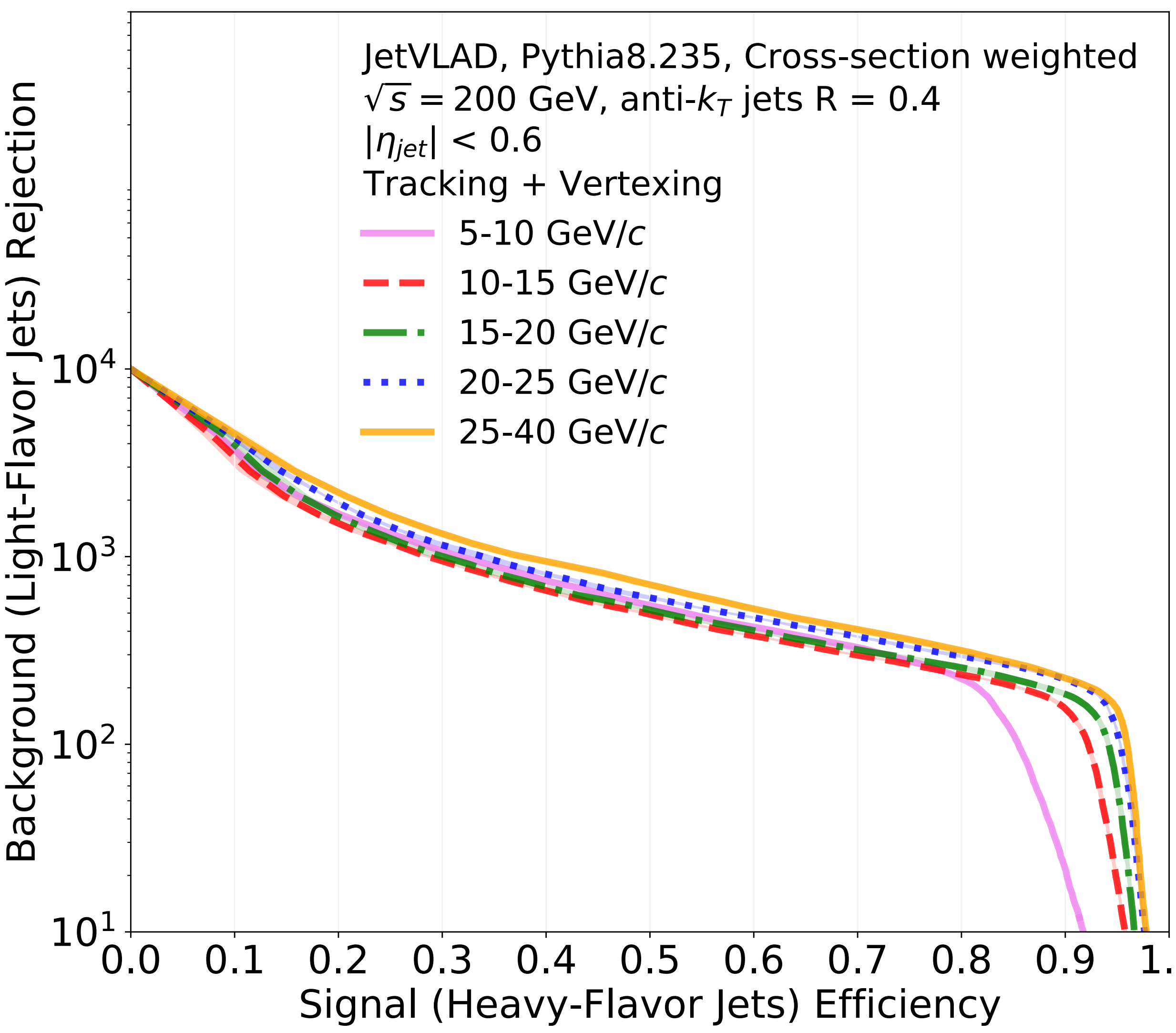
- It is important then to estimate sensitivity of hyperparameters with respect to model performance
- We do so by running hyperparameter sensitivity scan – fixing one and varying another





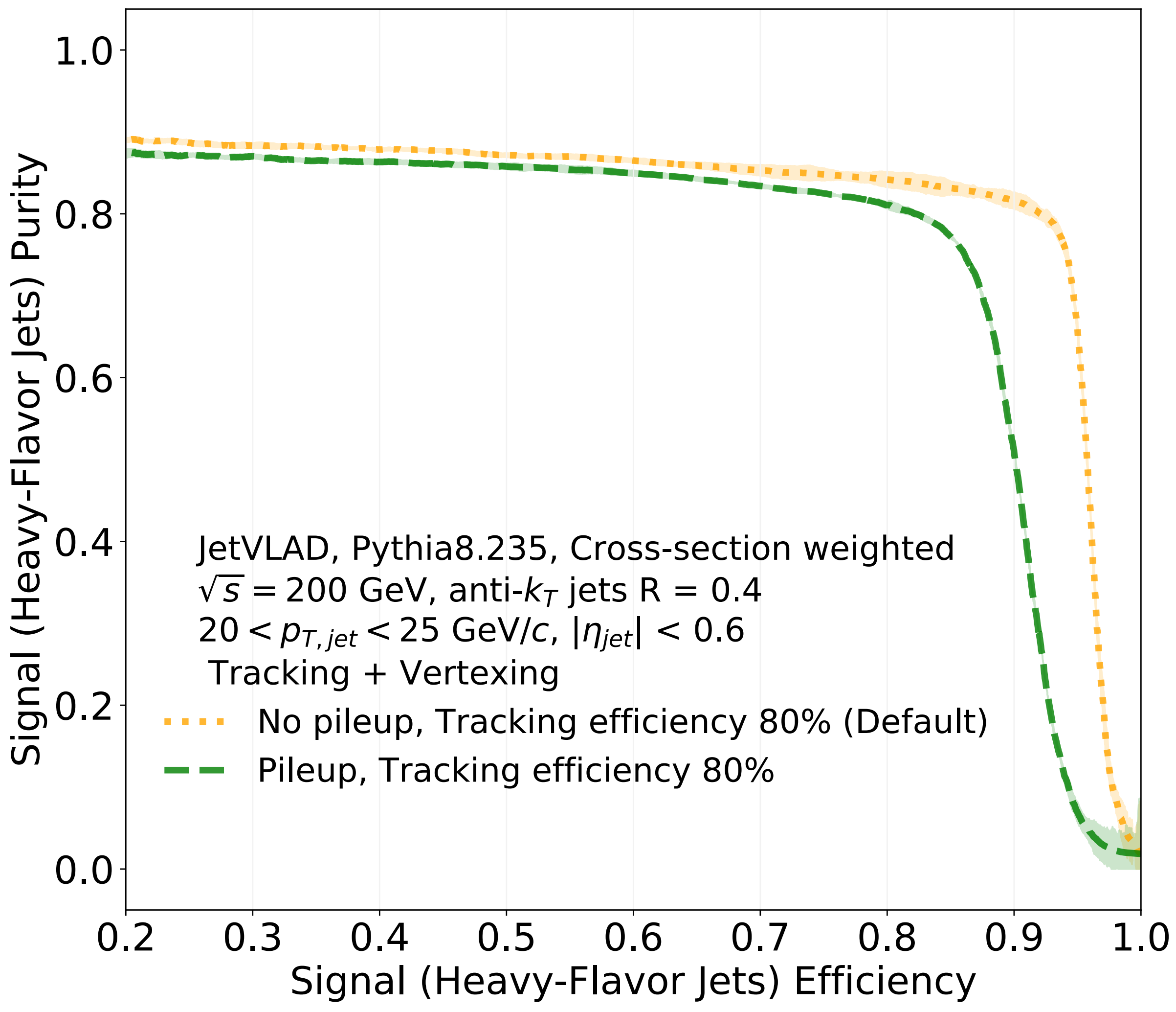
Jet p_T Dependent Rejection and Purity Graphs

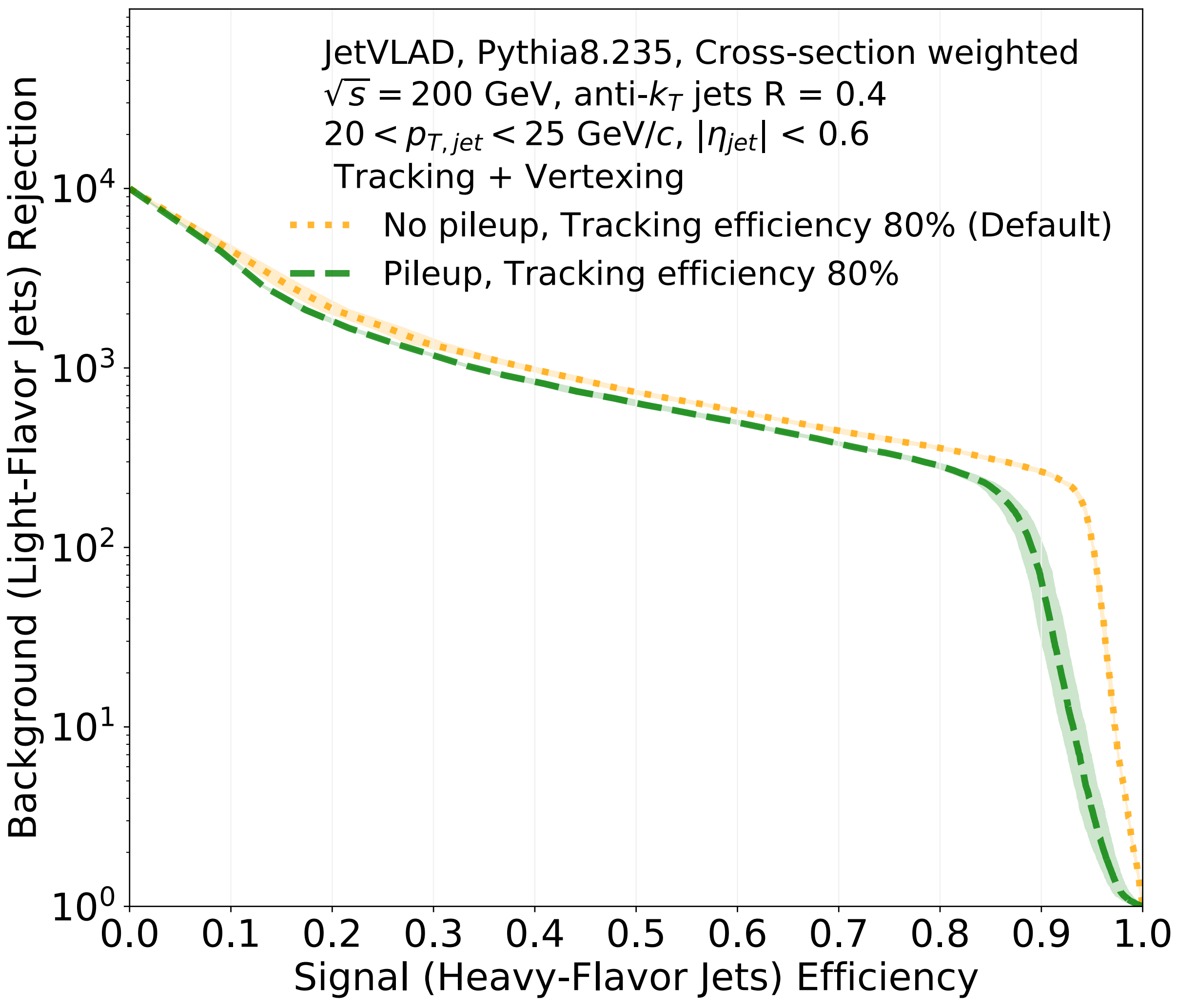




Our architecture achieves good performance across different p_T ranges

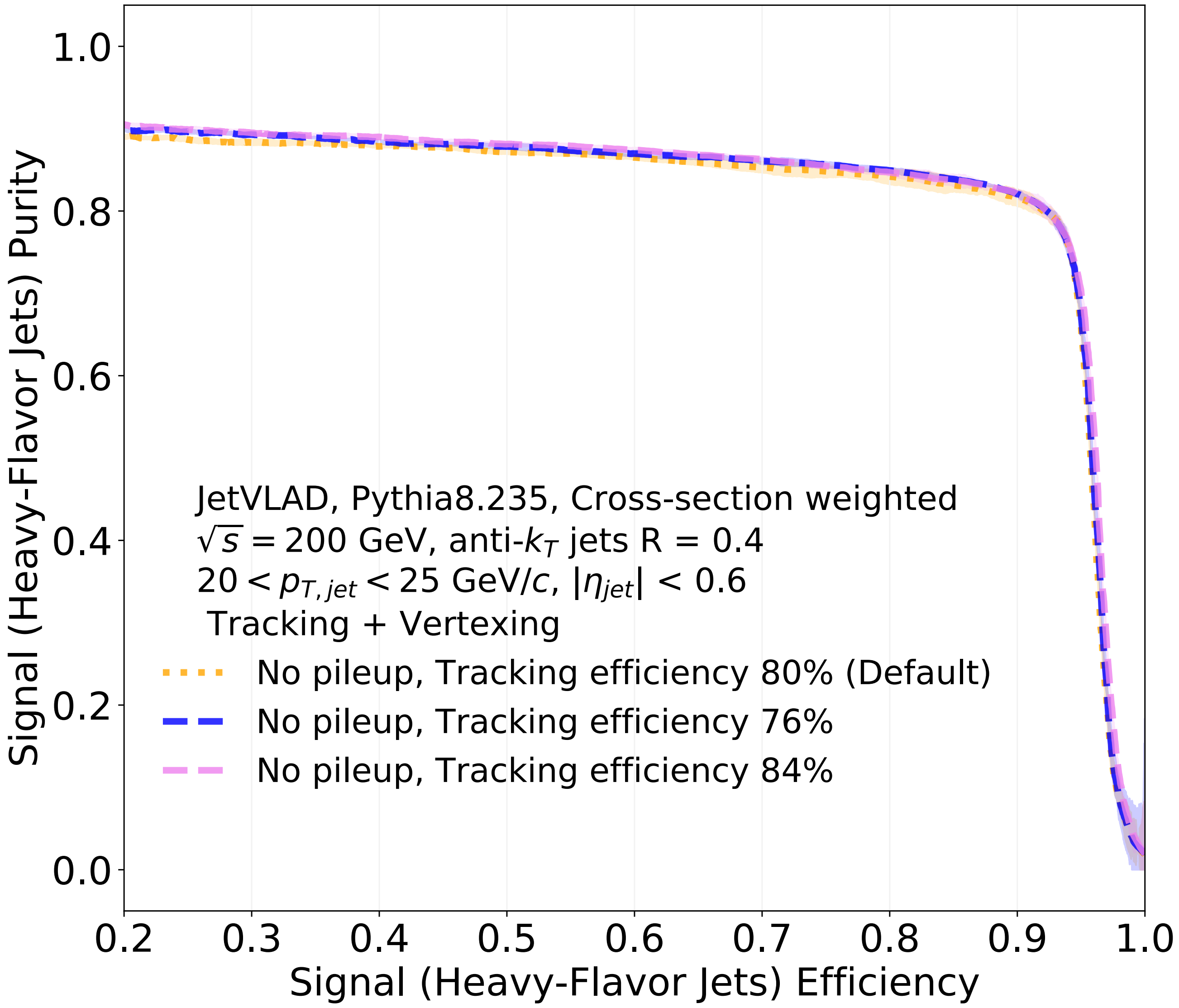
How is Pileup Influencing the Model?

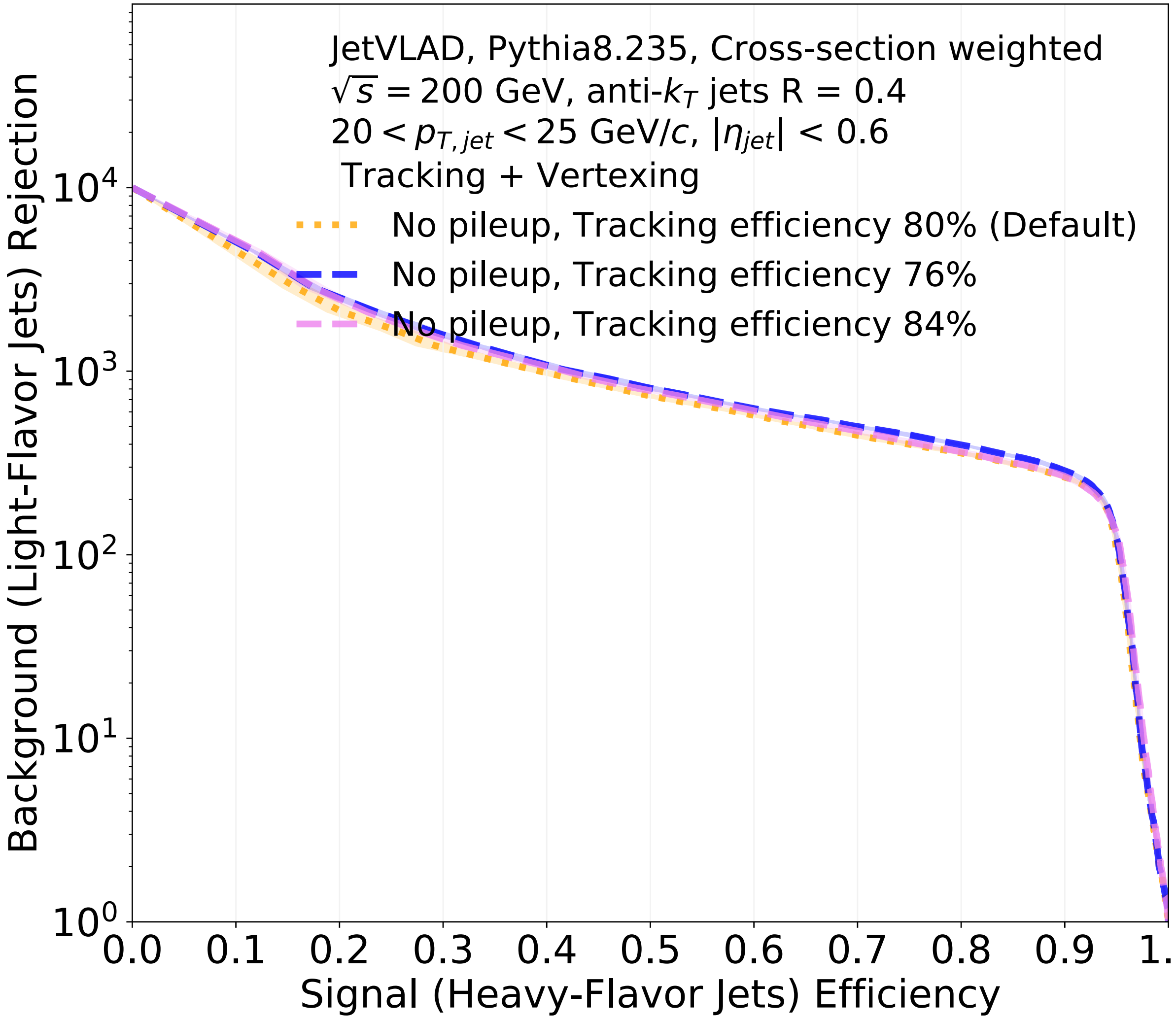




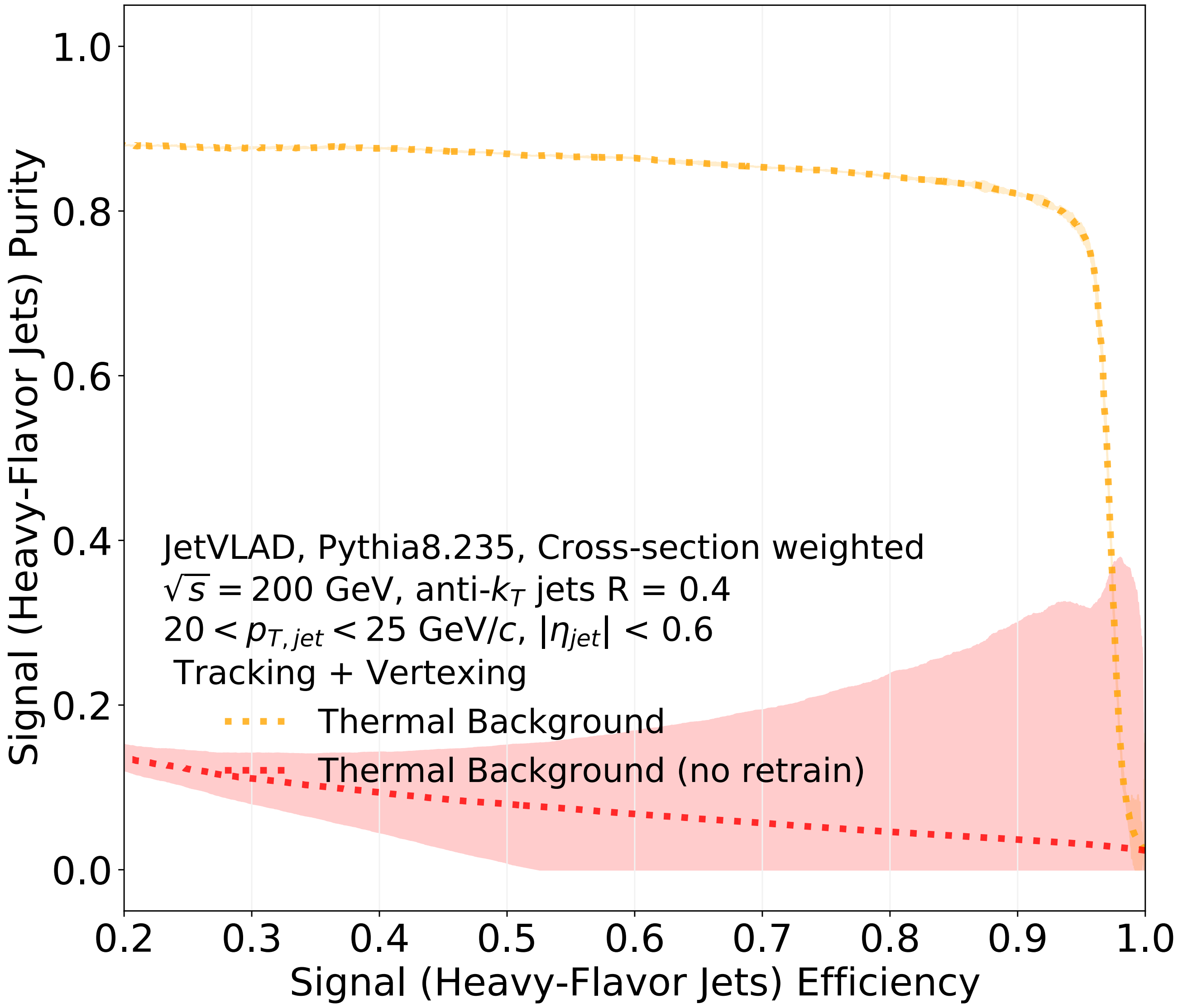
Add random min-bias event at random z position

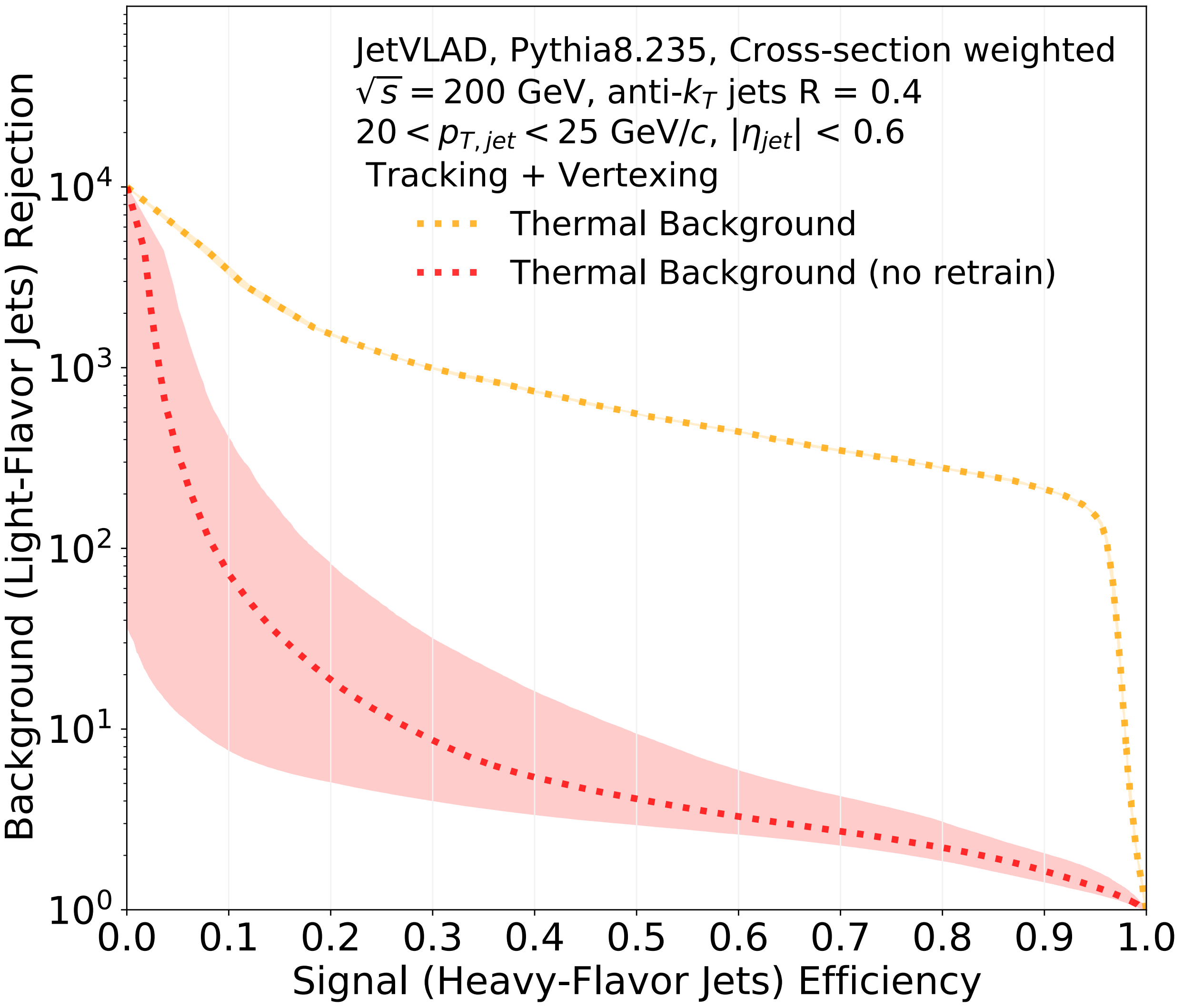
How is Tracking Efficiency Influencing the Model?





How is Thermal Backgrdoun Influencing the Model?





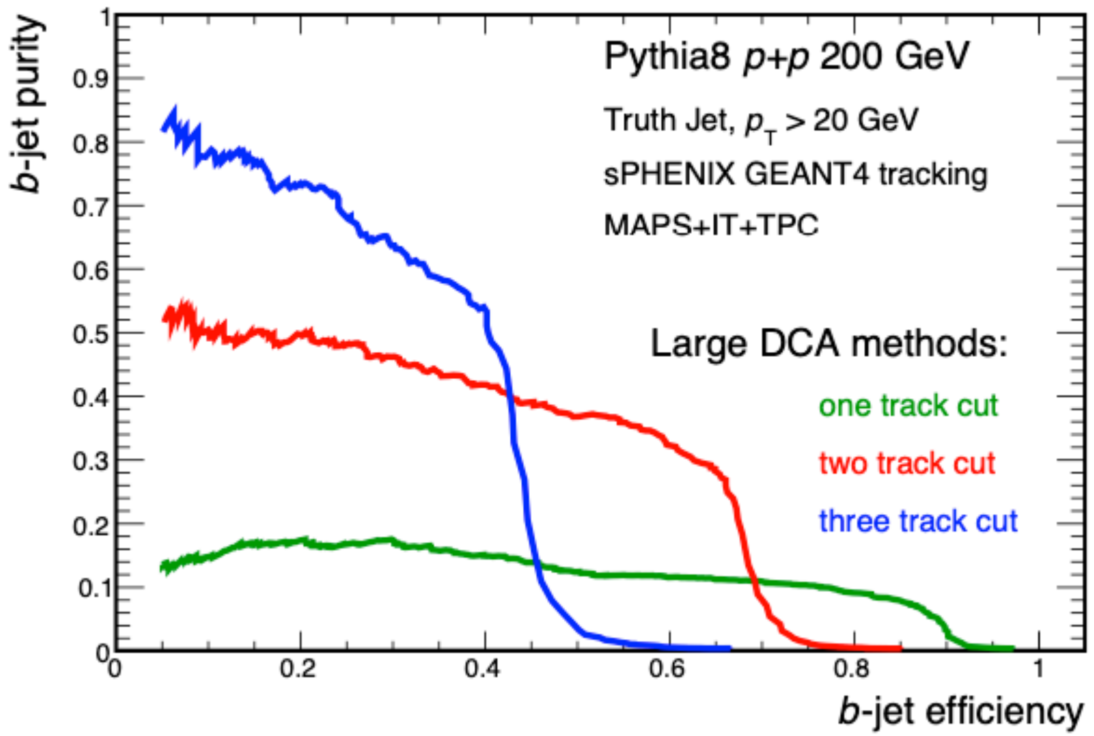
Conclusions

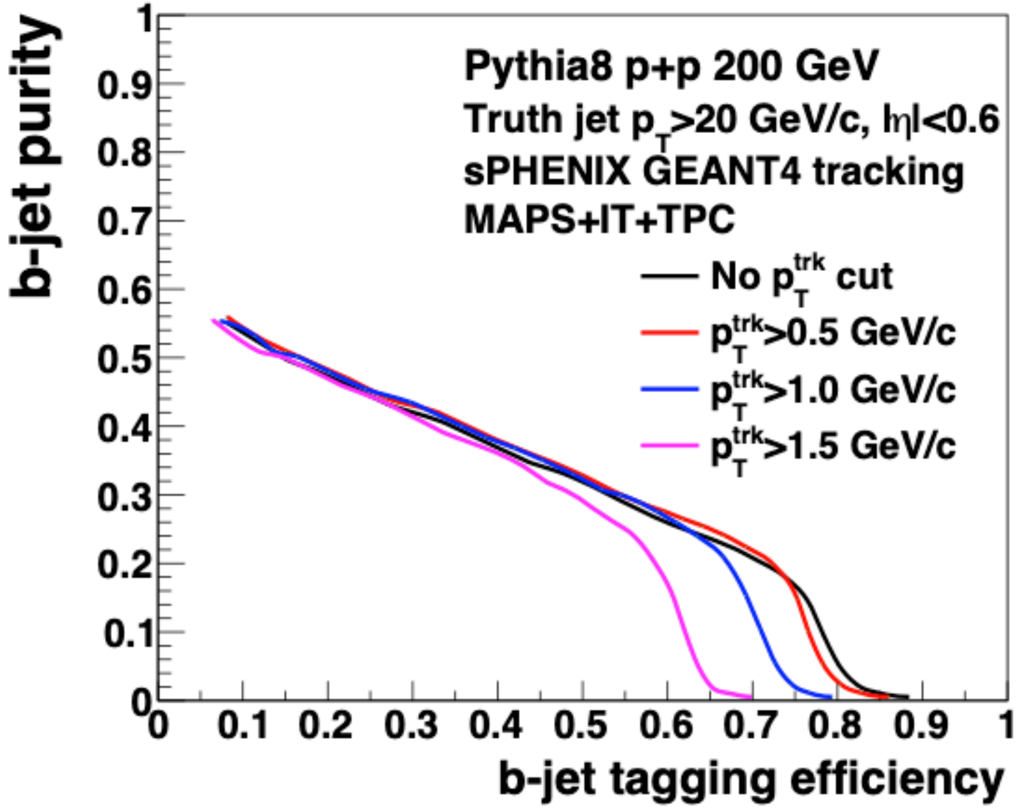
- We propose a novel set-based tagging method based on the NetVLAD layer
- The model allows to identify heavy-flavor jets down to the low-p_T at RHIC energies
  - Purity of $\sim 80\%$, Efficiency of $\sim 80\%$ and rejection factor of ~ 200 is achievable for low-p_T jet
  - This enables searches for signatures of heavy-flavor jet radiation patterns
- The NetVLAD layer is a general aggregation layer with many possible applications such as:
  - Jet similarity search
  - γ/π^0 discrimination
- We are investigating self-supervised and unsupervised learning applications of JetVLAD
  - Consistency cycles
  - Contrastive learning
  - Goal – to build multimodal jet-event feature space for fast similarity search of events/jets

https://github.com/ponimatkin/NetVLAD-tagger-pytorch

Thank you for your attention!

Backup: Performance of Classical Methods



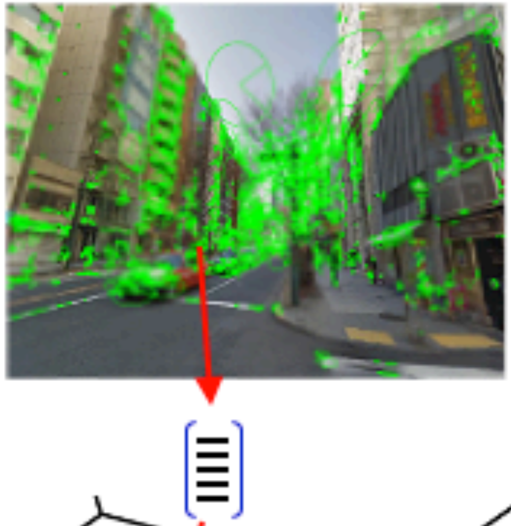


Taken from: “A monolithic active pixel sensor detector for the sPHENIX experiment”

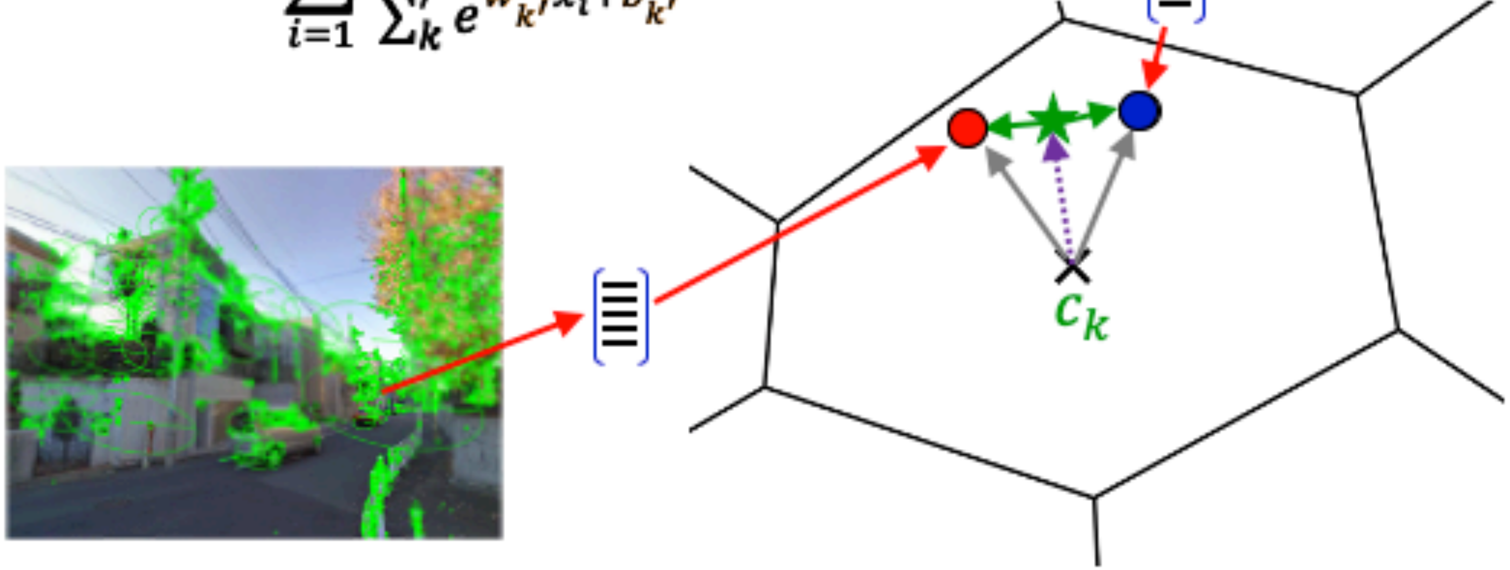
Backup: NetVLAD Principle

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.