Comparing and improving hybrid deep learning algorithms for identifying and locating primary vertices

S. Akar [1], M. Peters [1], H. Schreiner [1,2], M. D. Sokoloff [1], W. Tepe [1]

University of Cincinnati [1], Princeton University[2]

Abstract

- LHCb's **High Level Trigger** will process 5 TB/s of data. Machine learning algorithms have the potential to improve fidelity and execute very quickly
- We are developing a **hybrid deep learning algorithm** to identify primary and secondary vertices in *pp* collisions
- Previous DNN models architecture and performances presented at
  - ACAT 19 J.Phys.Conf.Ser. 1525 (2020) 1, 012079
  - CDT 20 arXiv:2007.01023
  - CHEP 21 EPJ Web Conf. 251 (2021) 04012

Supported by NSF awards OAC-1740102 & OAC-1739772 and via sub-awards under Cooperative Agreement OAC-1836650

The Run 3 LHCb Detector & Baseline Trigger

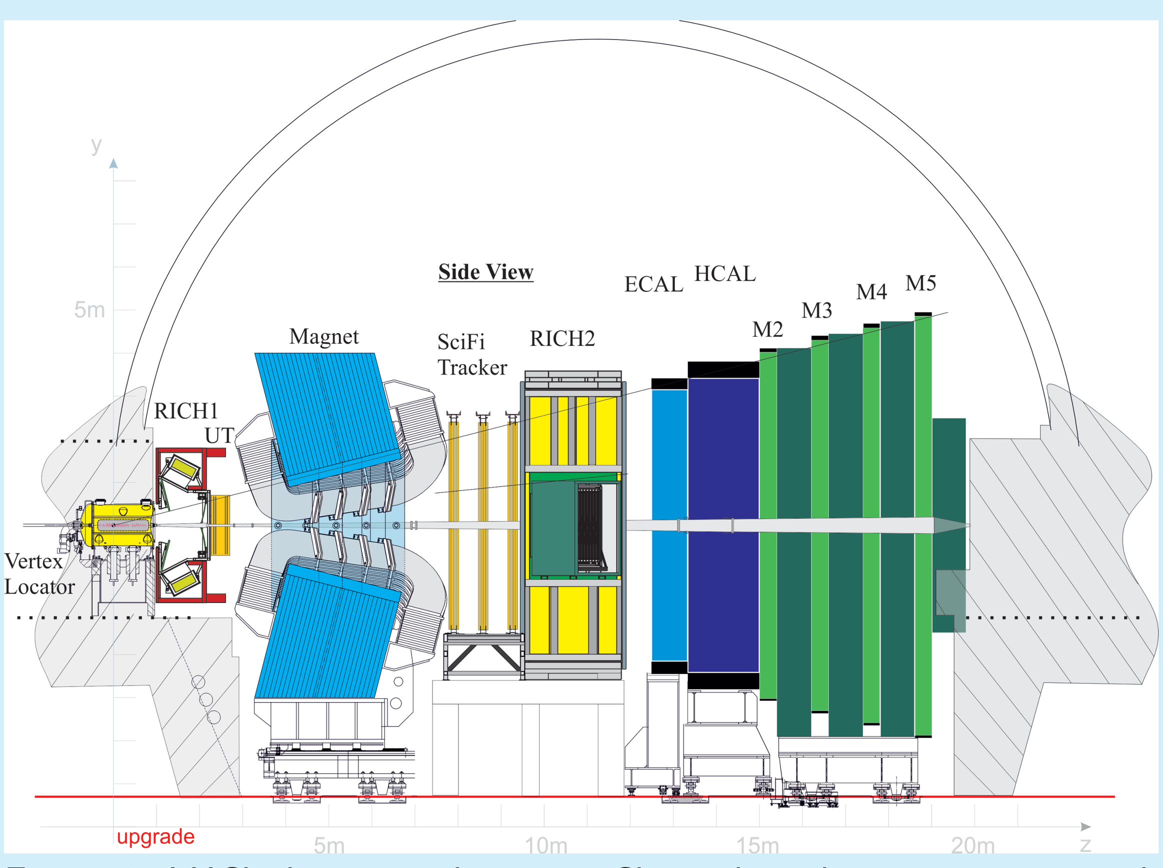


Figure 1: LHCb detector schematic. Charged tracks are reconstructed using data collected in the Vertex Locator (VELO) and 4 additional tracking stations (UT, T1–T3). LHCb is ~ 20 m long, 10 m high.

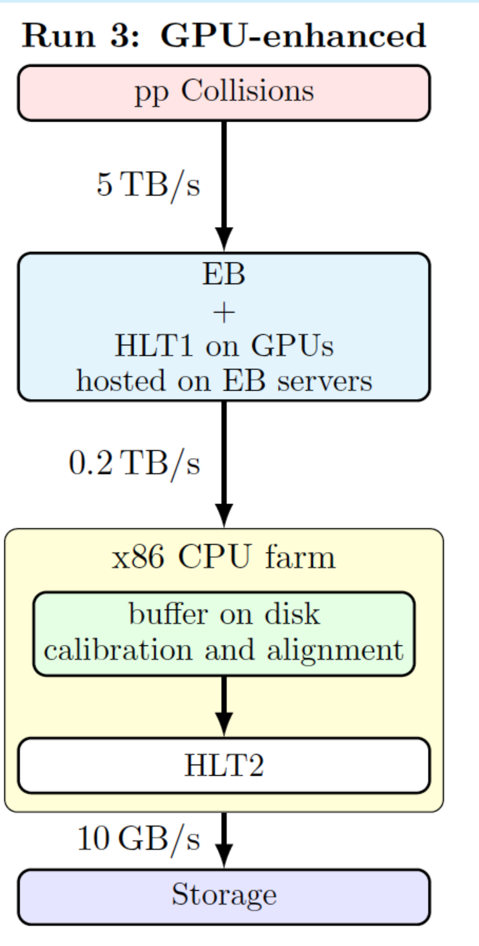
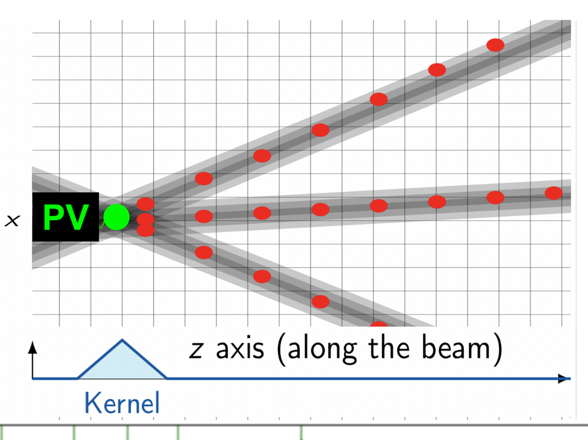


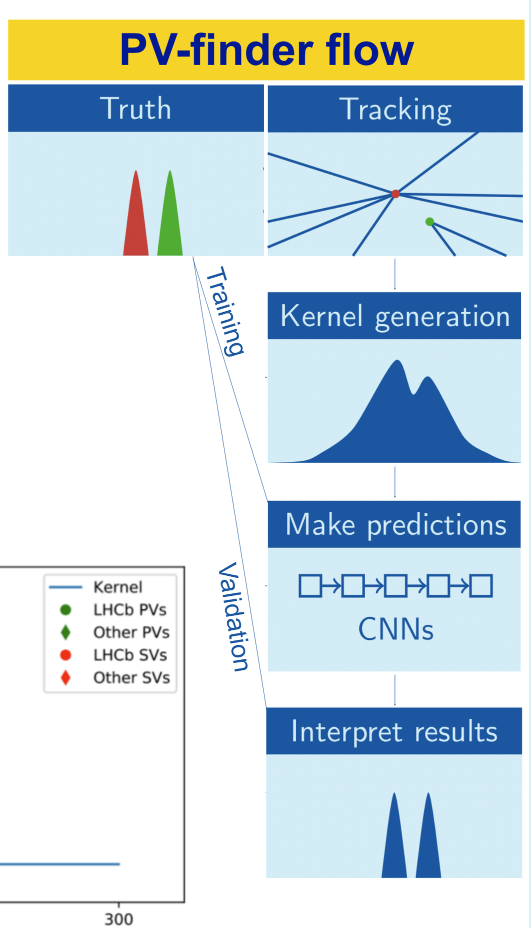
Figure 2: Run 3 LHCb Trigger Schematic

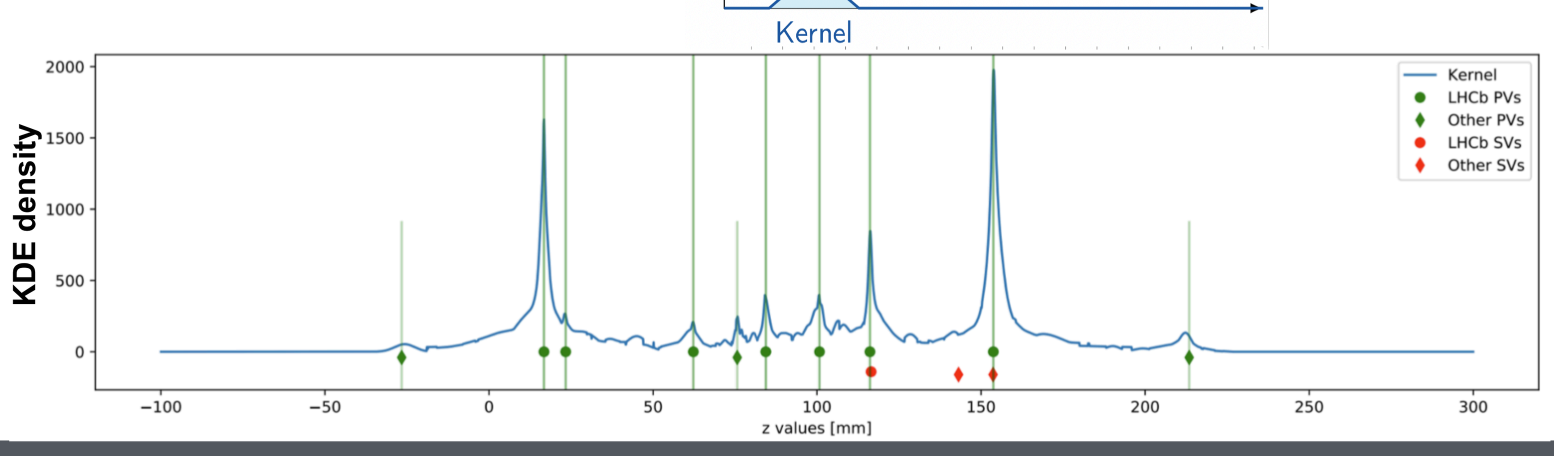
A hybrid ML approach to finding primary vertices

- **Using KDEs** (Kernel Density Estimators) **to reduce sparse 3D data (tracks parameters ; 41M pixels) to feature-rich 1D data – kernel densities in z**

$$\mathcal{K}(x,y,z) = \frac{\sum_{\text{tracks}} \mathcal{G}(\text{IP}(x,y)|z)^2}{\sum_{\text{tracks}} \mathcal{G}(\text{IP}(x,y)|z)} - \sum_{\text{tracks}} \mathcal{G}(\text{IP}(x,y)|z)$$



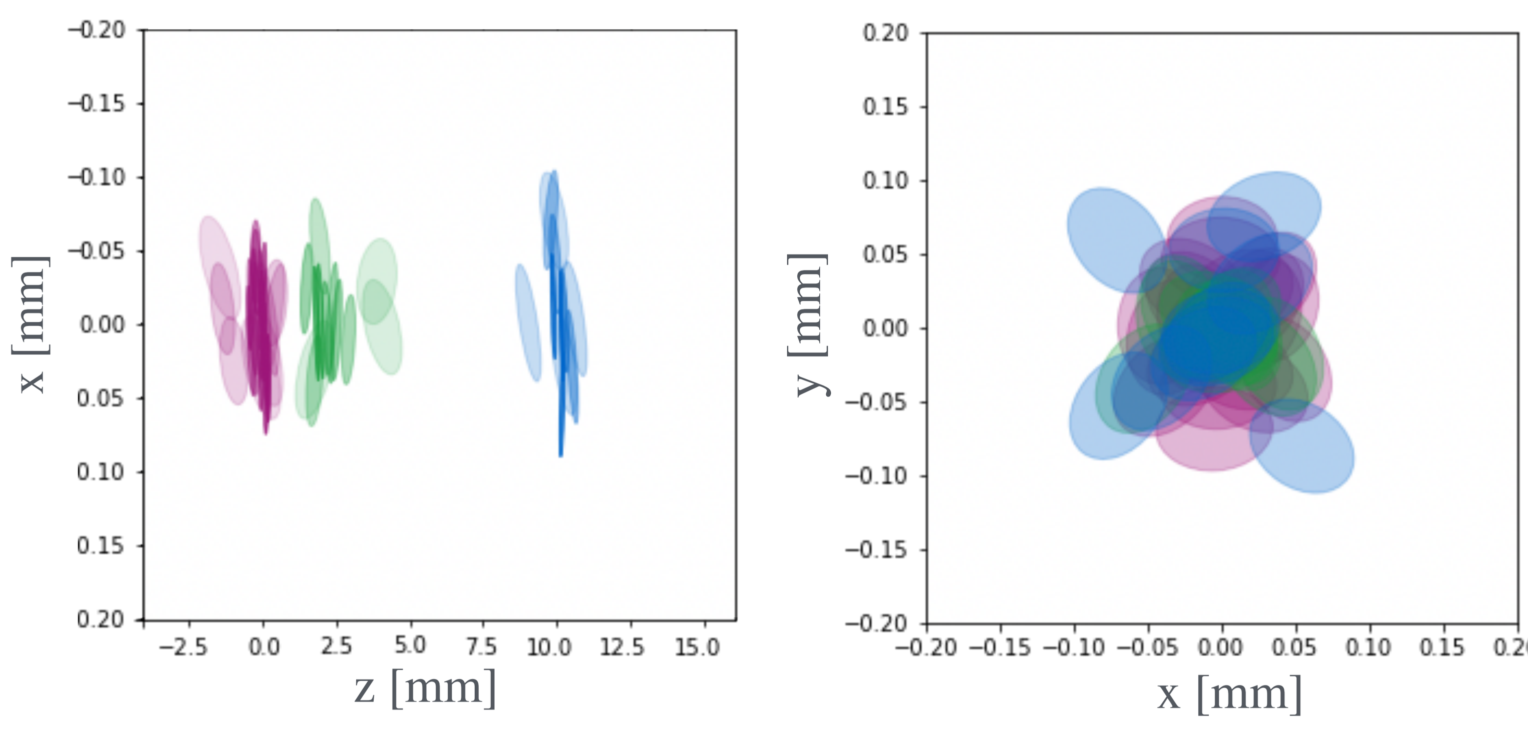




KDE distributions exhibit peaking structures near PV positions
Hand-written KDE computations expensive!

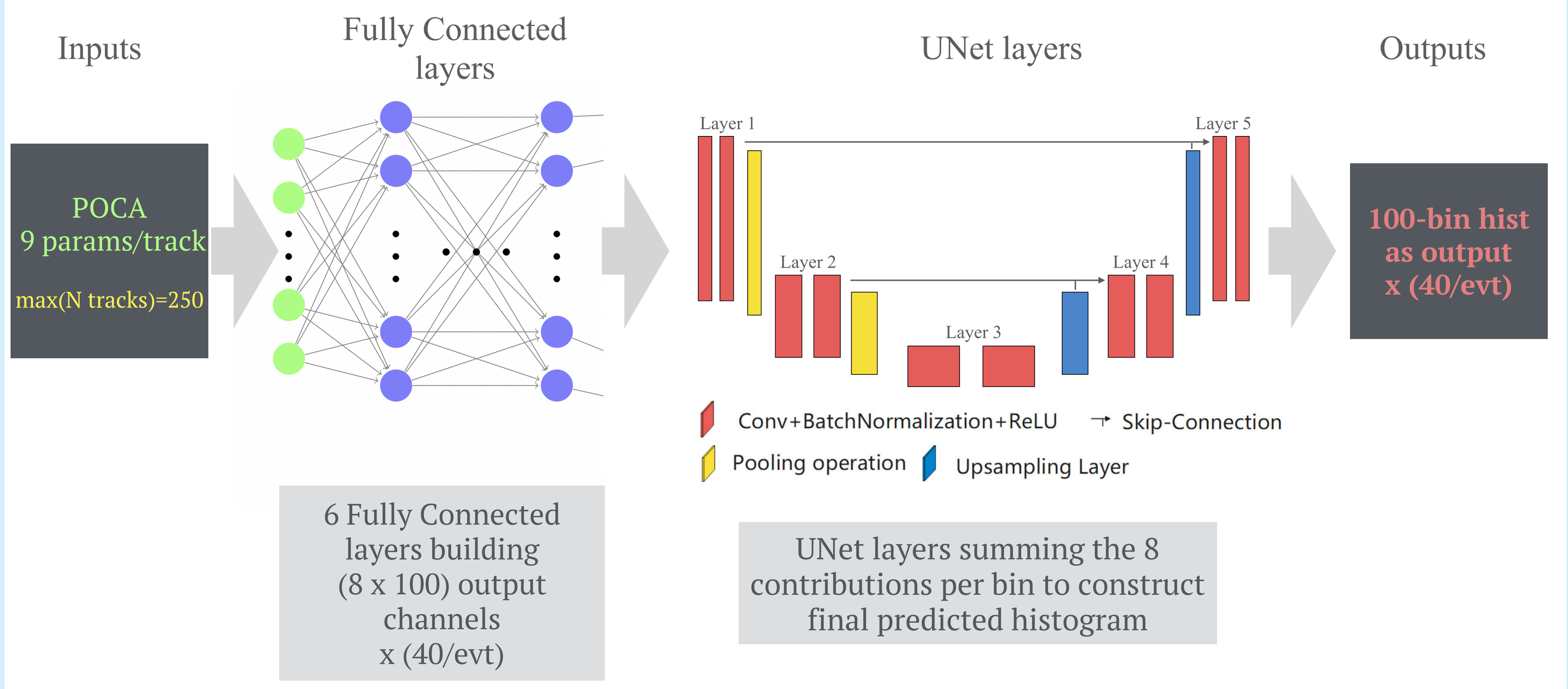
Updated input features

- Replaced **input tracks** information from **IP (impact parameter) to error ellipsoid at point of closes approach (POCA) to beamline:**

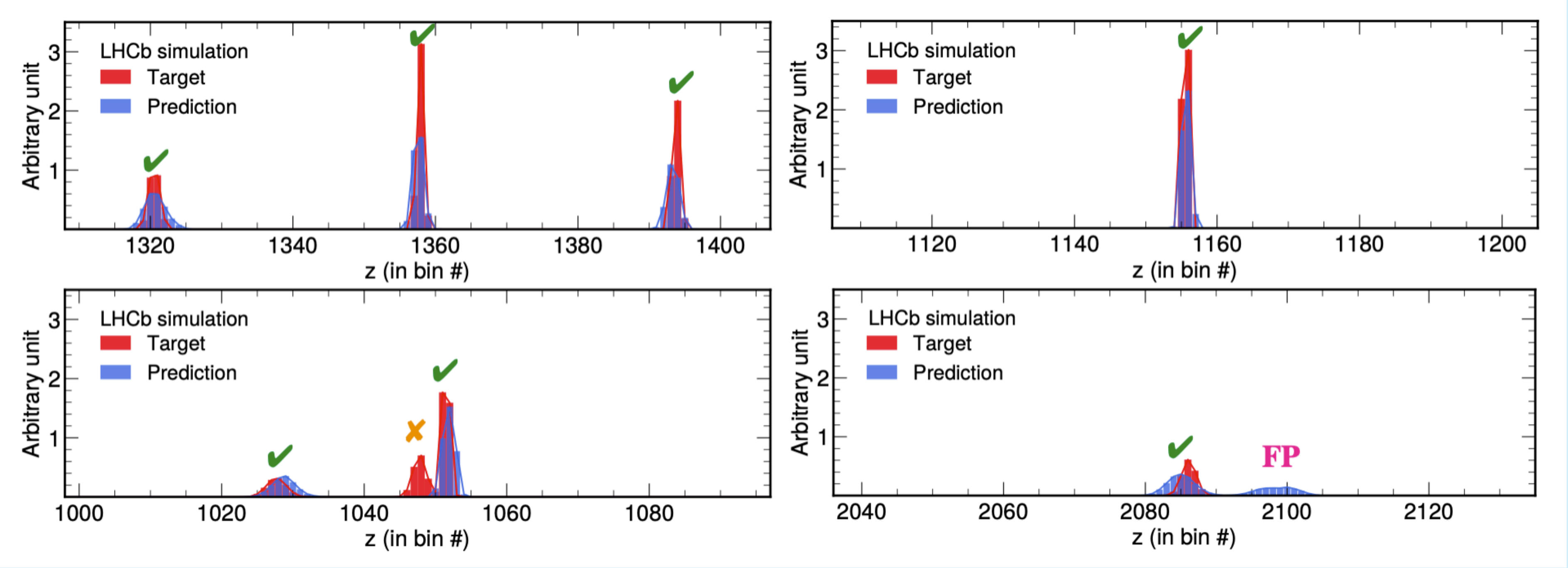


Each track represented as POCA-ellipsoid with 9 parameters
defining central position (3 pars.) and volume/uncertainty (6 pars.)

State of the art architecture [implemented using PyTorch]



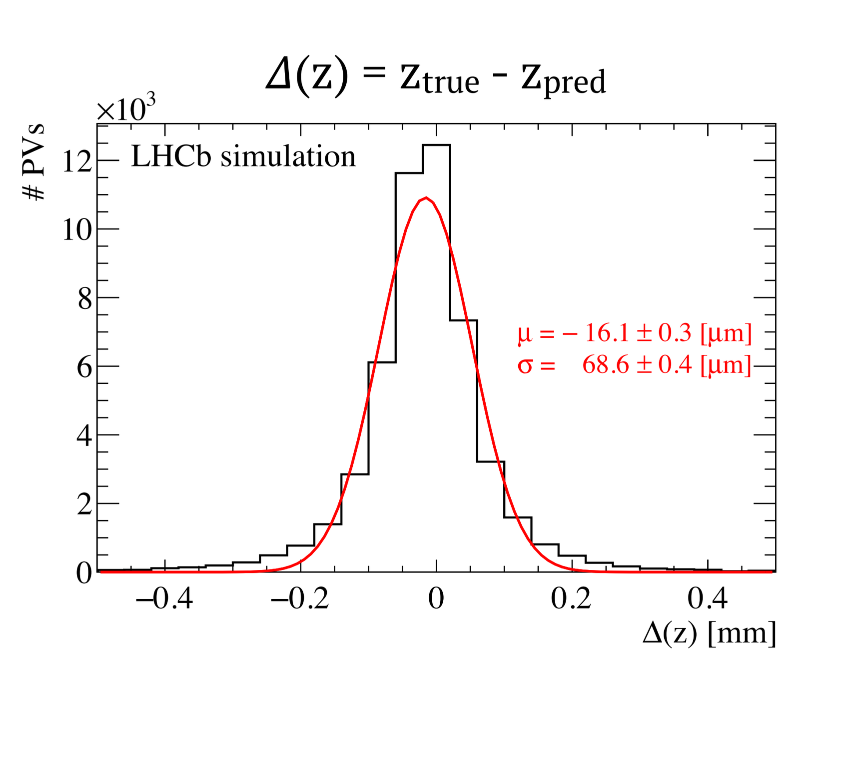
Target histograms as proxies to learn

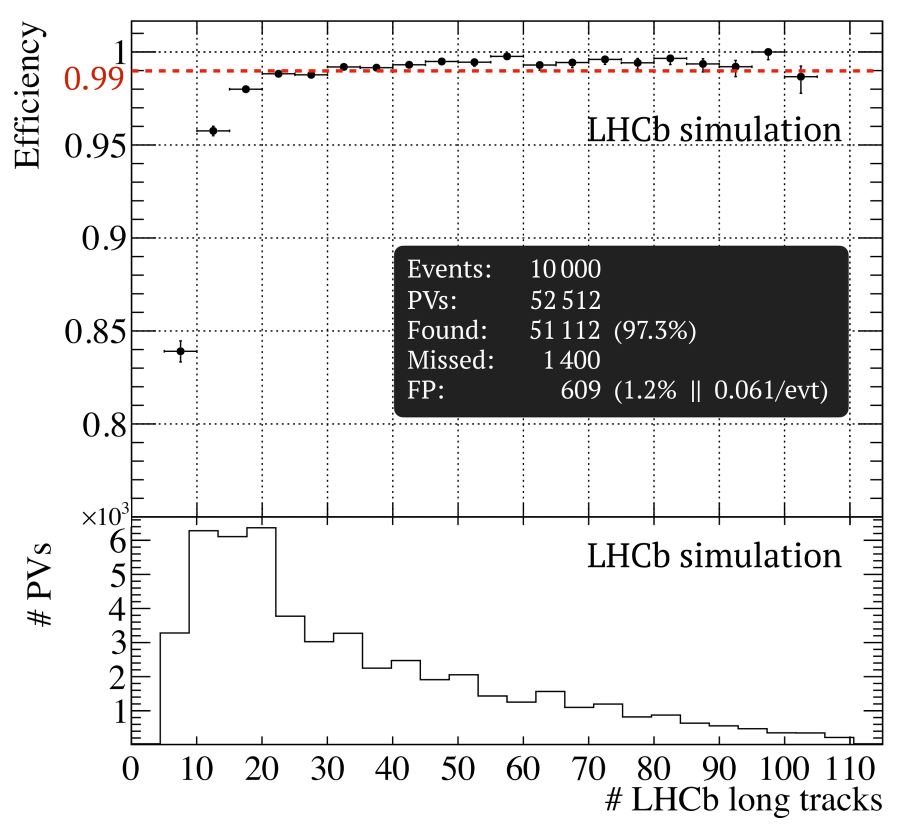


Performances: predicted position and efficiency

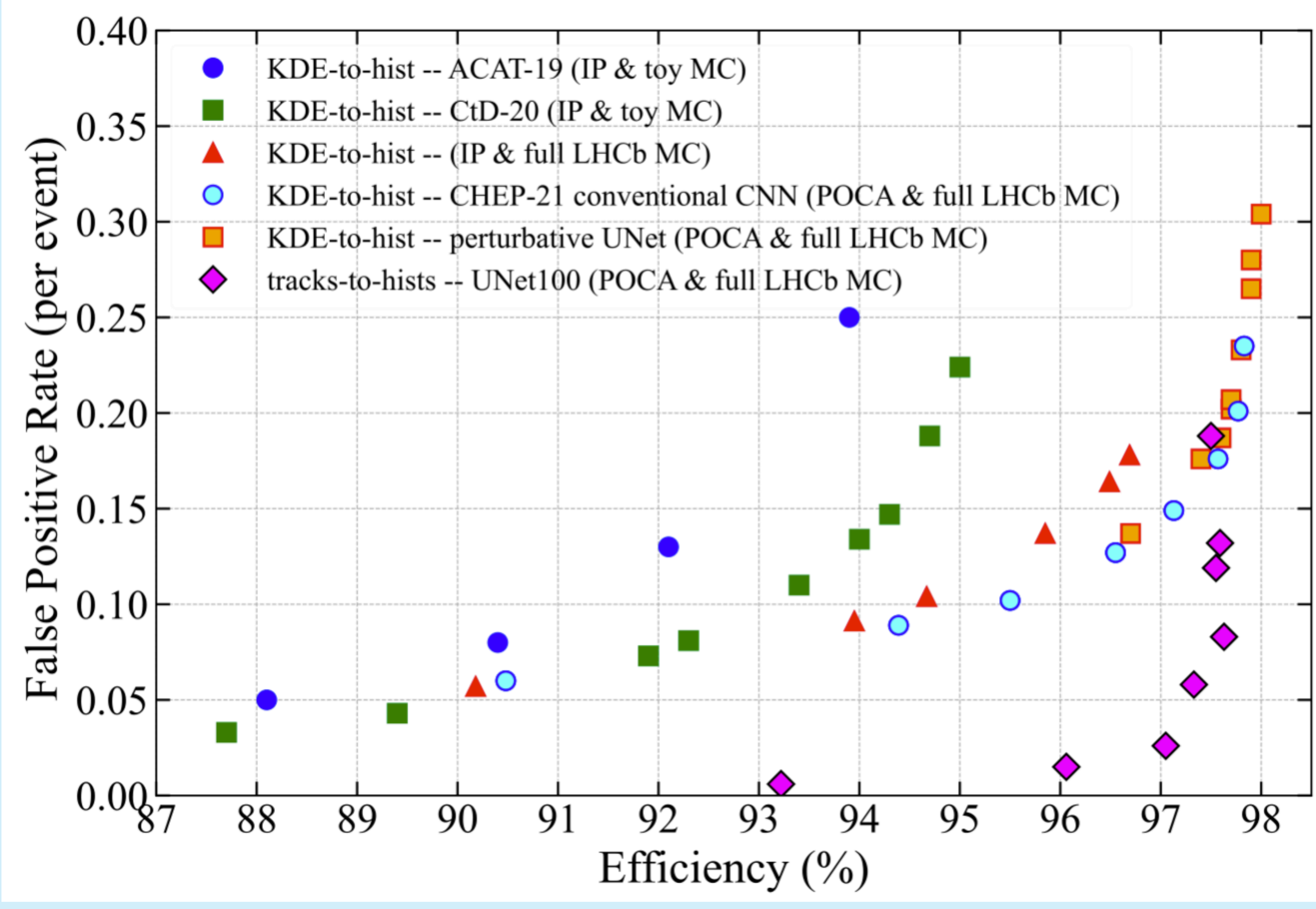
Predicted PVs position:
- from mean predicted hist
- small bias on $\Delta(z)$ of ~16 μm

Efficiency:
- matched if true PV in $\pm 5\ \sigma(z)$, with $\sigma(z)$ variance predicted hist





Performances evolution (Eff. vs FP rate) and ongoing studies



- **First end-to-end** (tracks-to-PVs) **DNN algorithm** with **high efficiencies** and **improved false positive rate** w.r.t. previous PV-finder models
- Ongoing deployment of inference engine in LHCb software stack
- Ongoing studies of PV-finder applications to other experimental conditions