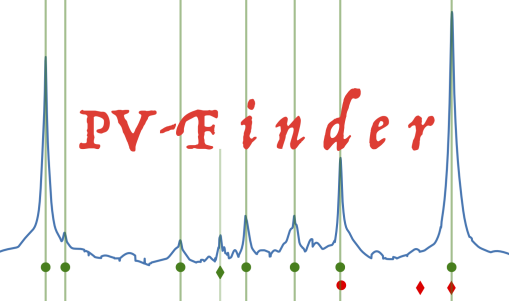
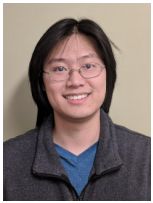


(Open) Source code of standalone package, including toy generation:

- https://gitlab.cern.ch/LHCb-Reco-Dev/pv-finder
- Runnable with Conda on macOS and Linux

iris hep
Institute for Research & Innovation in Software for High Energy Physics

NSF

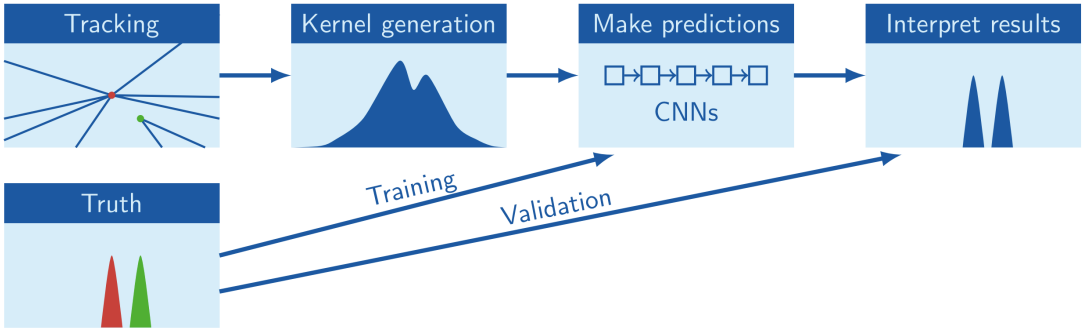


Gowtham Atluri[1] Kendrick Li[1] Henry Schreiner[2] Mike Sokoloff[1] Marian Stahl[1] Mike Williams[3]

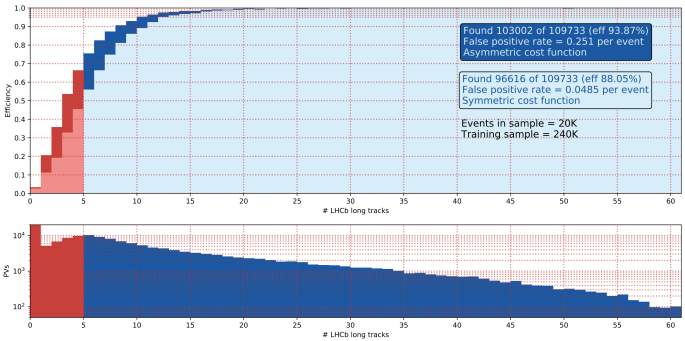
Simon Akar[1], Thomas Boettcher[3], Sarah Carl[1], Rui Fang[1], Michael Peters[1], Will Tepe[1], Constantin Weisser[3]

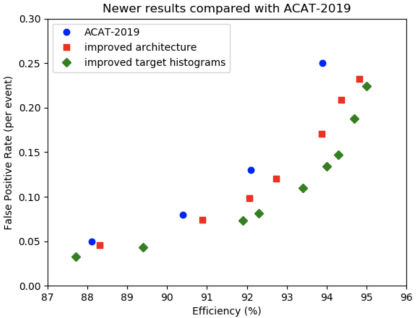
[1] Cincinnati, [2] Princeton, [3] MIT

- Challenge: in Run 3 LHCb the number of visible PVs will increase from $\sim$ 1.1 to 5.6
- Efficiency mainly driven by cluster search $\rightsquigarrow$ use machine learning
- The project is standalone, and uses toy data and it's own proto-tracking.
- A Gaudi compliant version using production tracking has recently been deployed in the Run 3 LHCb CPU software stack. This is the workflow in a nutshell:



Results





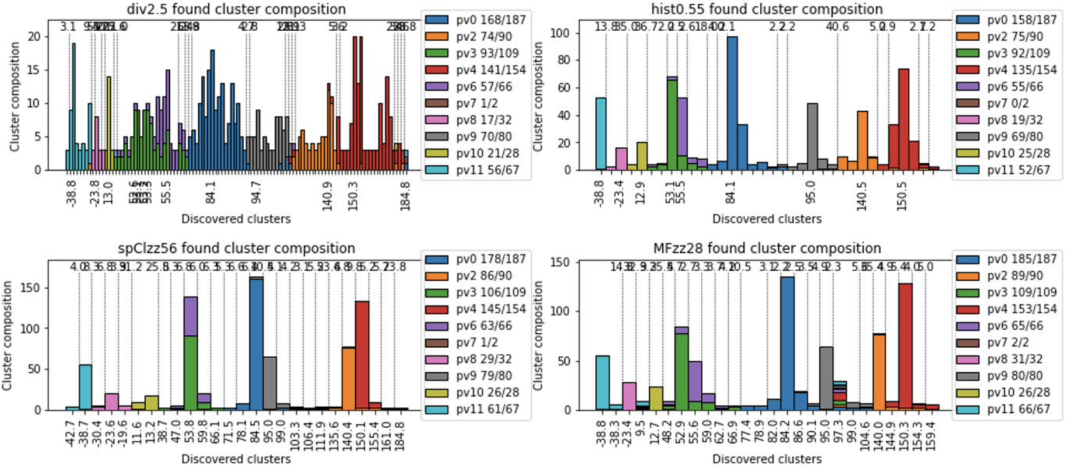
- Proof of principle established. Improved performance further by
  - modifying target histograms (learning proxies);
  - adding layers to CNN and adding x, y position information perturbatively.
- For a fixed efficiency of 94 %, the false positive rate is about 2× smaller.
- CNN performance is studied in detail from the computer science perspective.

Characterizing PVs and Efficiency

Multiplicity	Distance (closeness)	Spread (IQR)	SVs	Frequency of PVs with combination	Recall for these PVs
very small	close	wide	0 SV	2.86%	7.14%
very small	close	thin	0 SV	7.26%	12.68%
very small	very close	thin	0 SV	3.68%	13.89%
very small	close	very thin	0 SV	11.15%	20.18%
very small	far	very thin	0 SV	5.62%	23.64%
very small	very close	very thin	0 SV	8.38%	35.37%
very small	very far	very thin	0 SV	2.56%	40.00%
small	very close	very thin	0 SV	4.50%	40.91%
very large	very close	very thin	0 SV	2.45%	41.67%
small	close	thin	0 SV	2.04%	50.00%
small	close	very thin	0 SV	8.18%	61.25%
very large	close	very thin	2 SV	2.56%	64.00%
small	far	very thin	0 SV	2.76%	74.07%
very large	close	very thin	0 SV	3.78%	78.38%

- Study of inefficiencies in categories of track multiplicity, distance to neighboring PVs, spread of track-PV point-of-closest-approach distribution and associated secondary vertices (SVs).
- Uses PVFinder toy data, but a different cluster finding algorithm;
  Very efficient, or low frequency categories excluded from the table

- Track to PV association with different clustering algorithms:



Year 3 goals:

- Benchmark performance with standard LHCb simulation and software, and compare to the current baseline PV finding algorithm.
- Re-train the algorithm using full LHCb simulation in place of toy simulation.
- Deploy the algorithm in Allen.

Year 3+ goals:

- Develop an algorithm to assign tracks to PVs probabilistically.
- KDE generation too slow $\rightsquigarrow$ develop fast ML algorithm for KDE generation, then combine the two algorithms into one.
- Prune ML algorithms if necessary to fit into the HLT time budget.
- Improve existing algorithm by first characterizing PVs in categories of multiplicty, track variance, PV proximity and SVs; Develop targeted ML approaches to tackle inefficiencies.