PV-Finder for ATLAS: Exploring a deep learning approach for Primary Vertex Identification

Elliott Kauffman

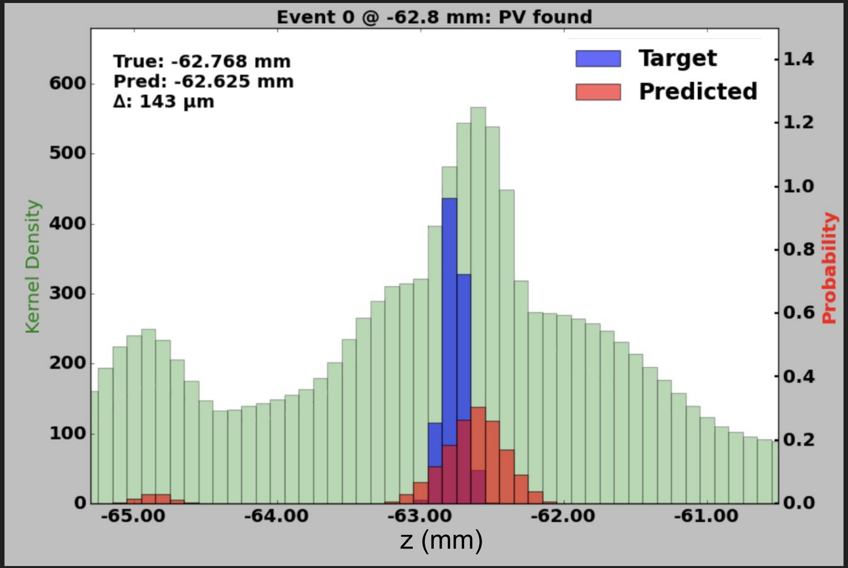
Mentors: Rocky Bala Garg[1], Henry Schreiner[2], Mike Sokoloff[3]

[1] Stanford University
[2] Princeton University
[3] University of Cincinnati

PV-Finder Overview

- Initially developed for use in LHCb data
  - They currently achieve efficiency values that exceed 98% for a pileup of 5.6 with a low false positive rate
- **Goal:** use machine/deep learning techniques to reconstruct primary vertices using detector hits and reconstructed tracks
- **Step 1:** Use reconstructed tracks to calculate **Kernel Density Estimators (KDEs)** for each event
- **Step 2:** Use truth information to calculate **labels** for use in the neural network
- **Step 3:** Use a convolutional neural network to **predict** the primary vertex locations

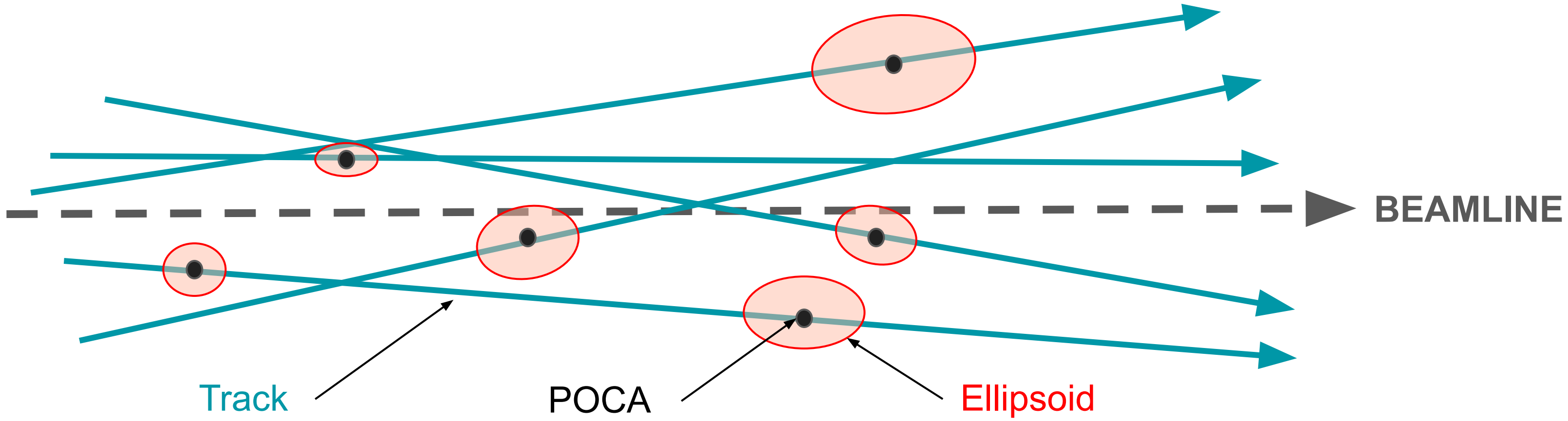
LHCb Monte Carlo



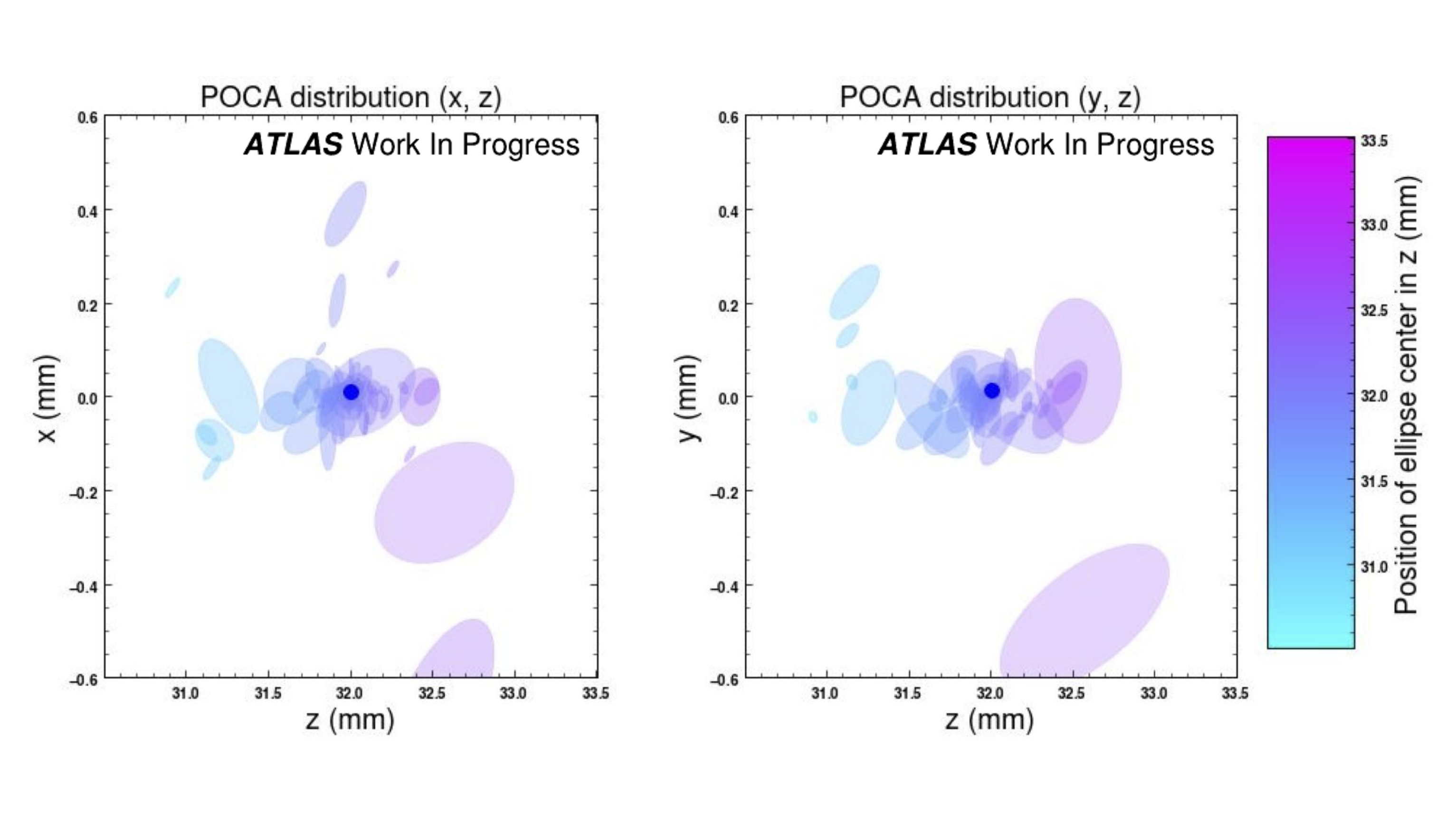
KDE: Track density as a function of z (along the beamline). Peaks correlate with vertex locations

Reconstructed tracks → POCA ellipsoids

- Using **truth-matched reconstructed tracks** for training purposes
- Each track is represented as a **3-dimensional ellipsoid**
  - The center of each ellipsoid is the track's **point of closest approach (POCA)** to the beamline
  - The size of each ellipsoid is proportional to the track's uncertainty
  - Smaller ellipsoids have a larger contribution
- Effect of using POCA ellipsoids: tracks only contribute significantly where they are closest to the beamline (ideally closest to their parent vertex)

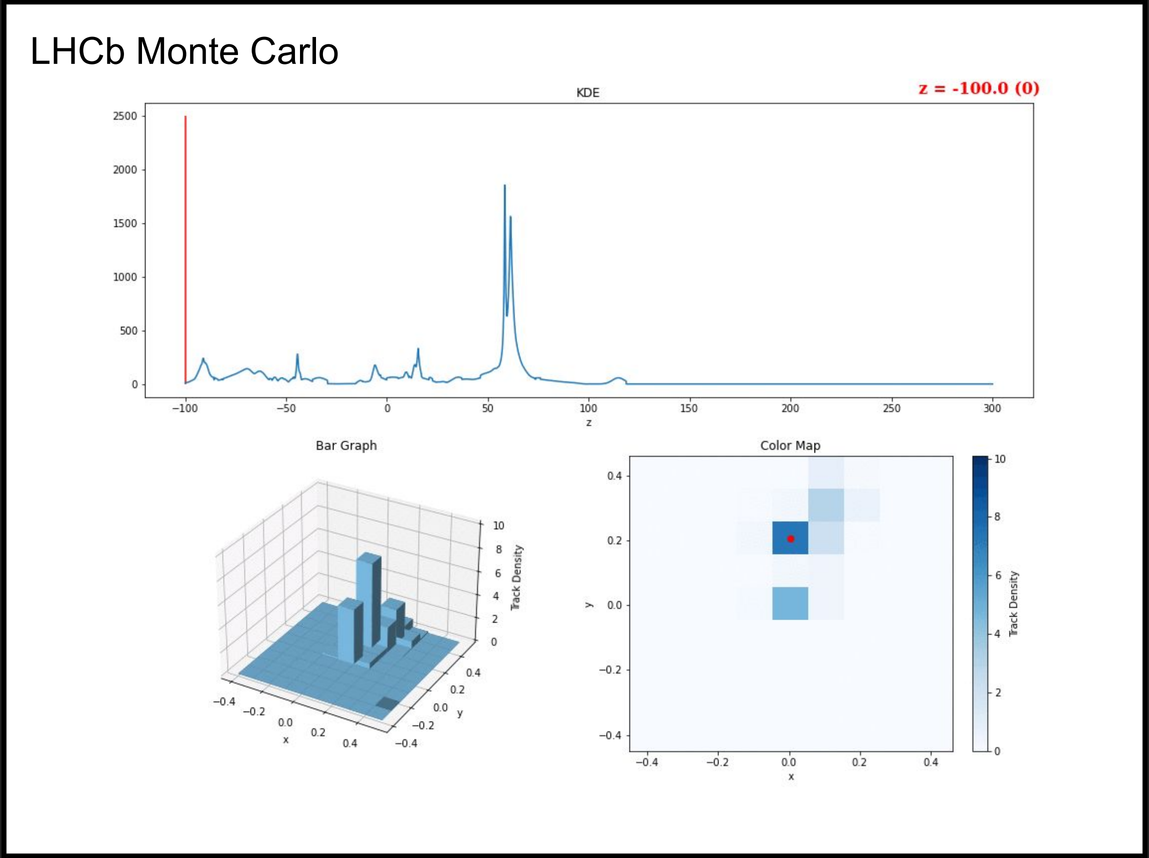


Visualization: POCA Ellipsoid Projections



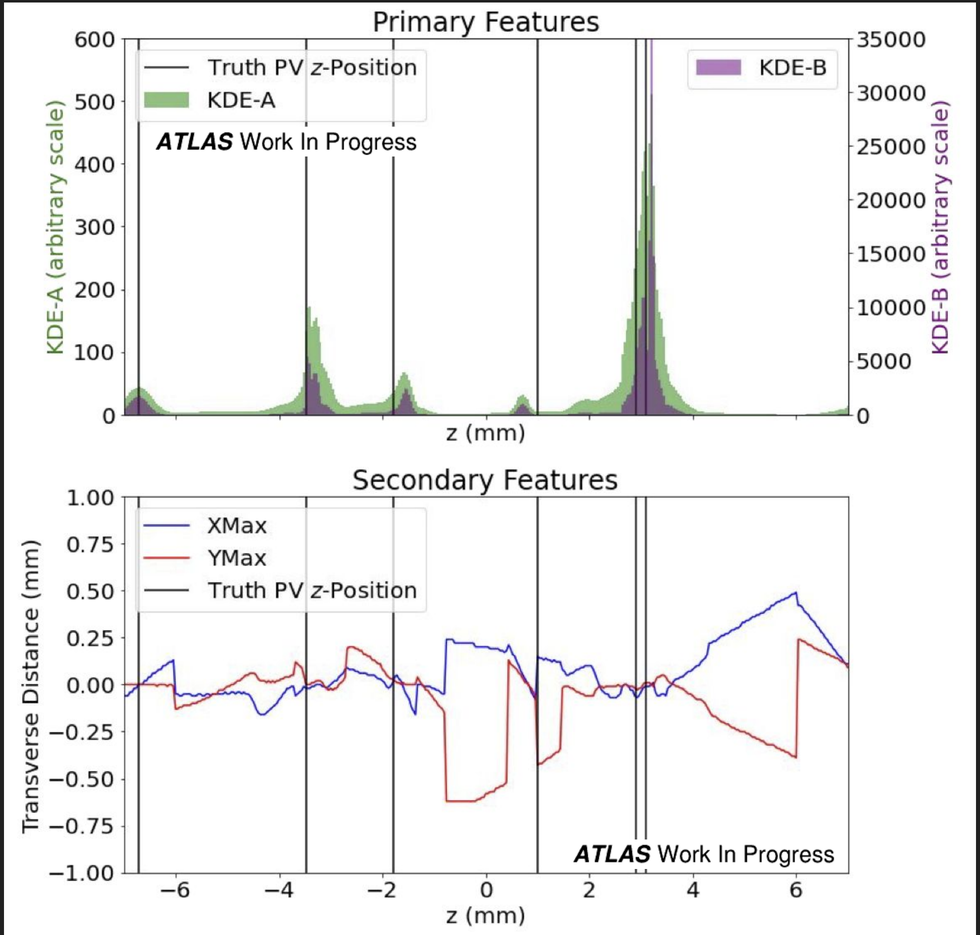
POCA Ellipsoids → KDE

- For each POCA ellipsoid, a Gaussian probability is calculated
- For each bin along the z-axis, the probabilities of the contributing tracks are summed
- For each z bin, we locate the point with the **maximum track density** and record it’s height and position
    - **Coarse grid search** followed by **MINUIT** minimization starting from that point
- The **height** becomes the **KDE value** for that z bin
- The position of the maximum **(XMax, YMax)** are also used as features

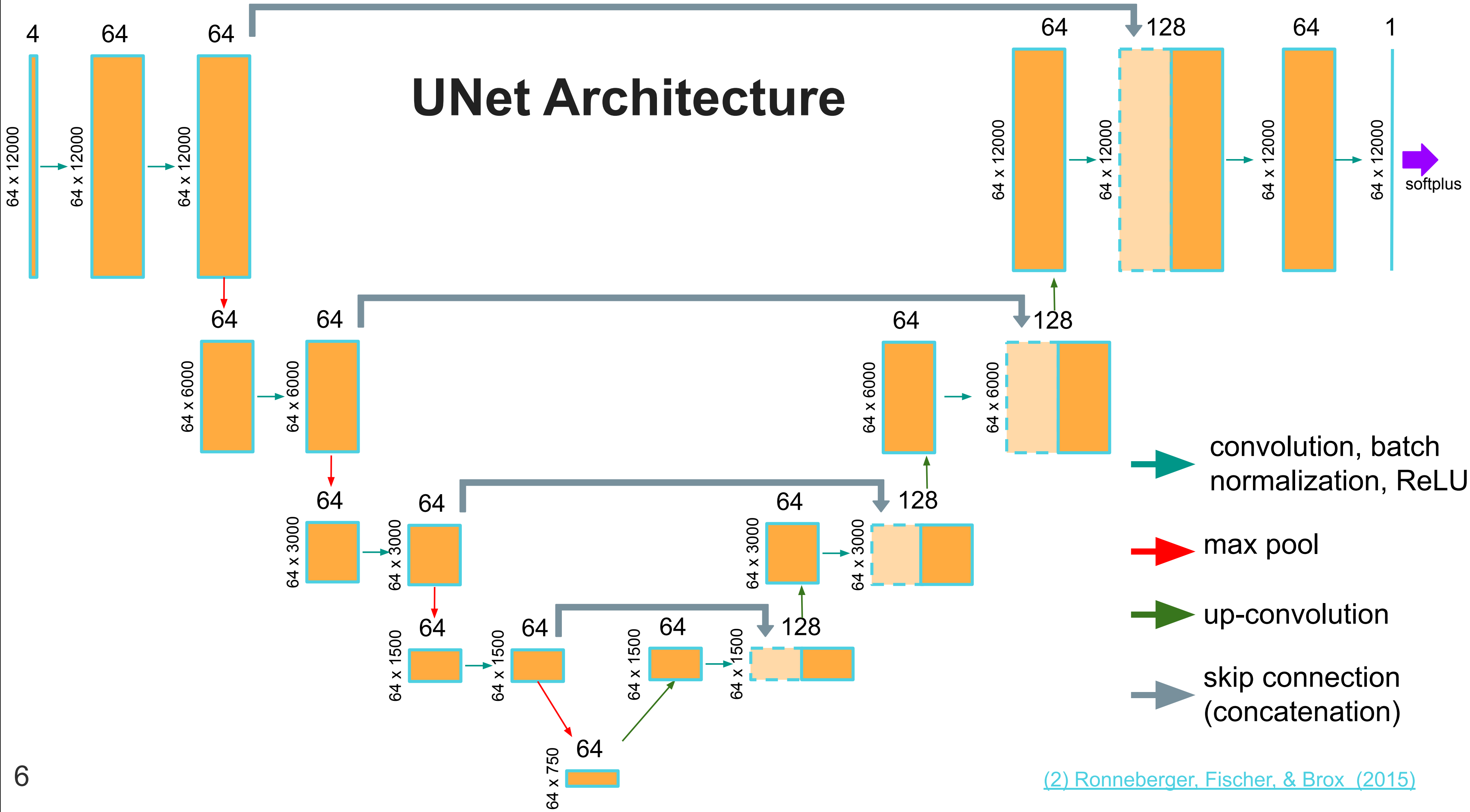


Summary of Neural Network Input Features

KDE-A: sum of probability
KDE-B: sum of (probability squared)
XMax: the x-coordinate of the KDE-A value for each z bin
YMax: the y-coordinate of the KDE-A value for each z bin



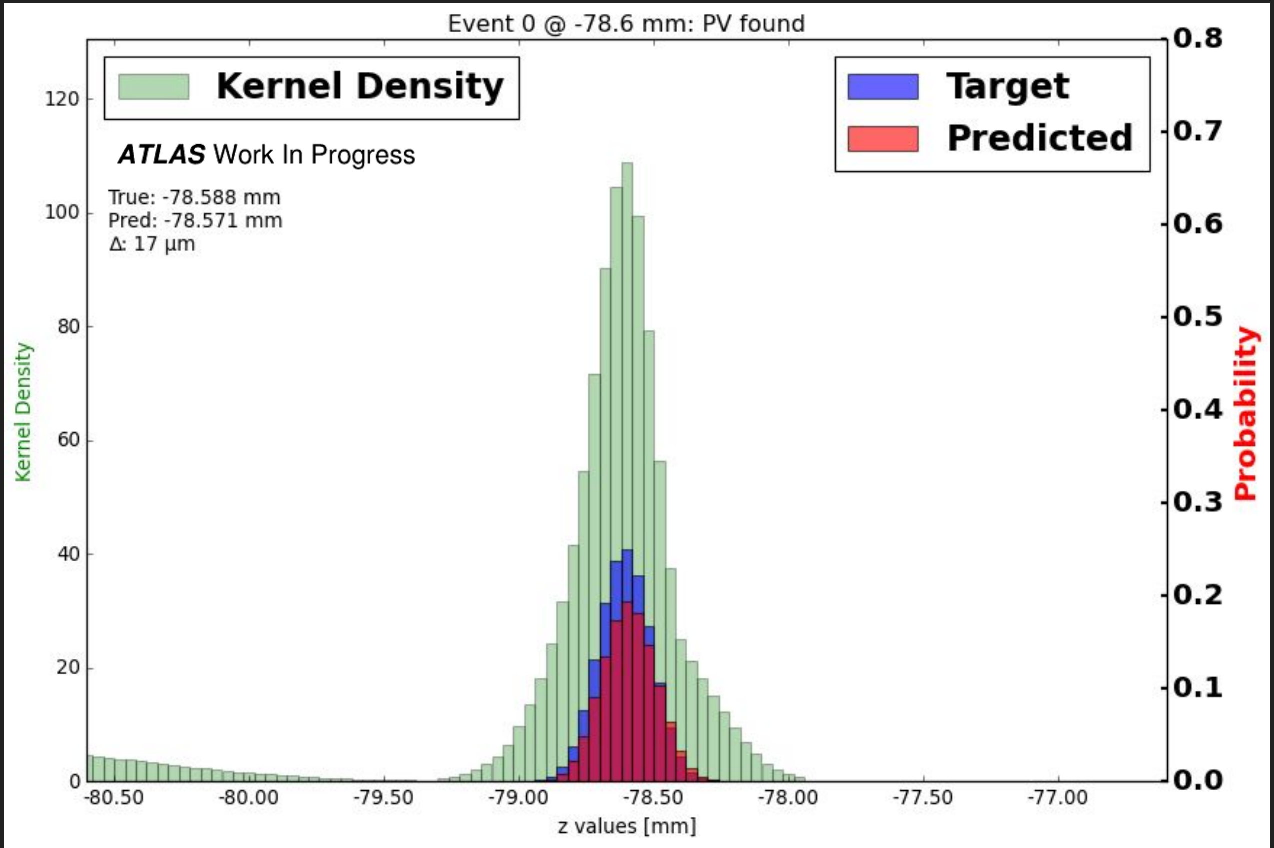
UNet Architecture



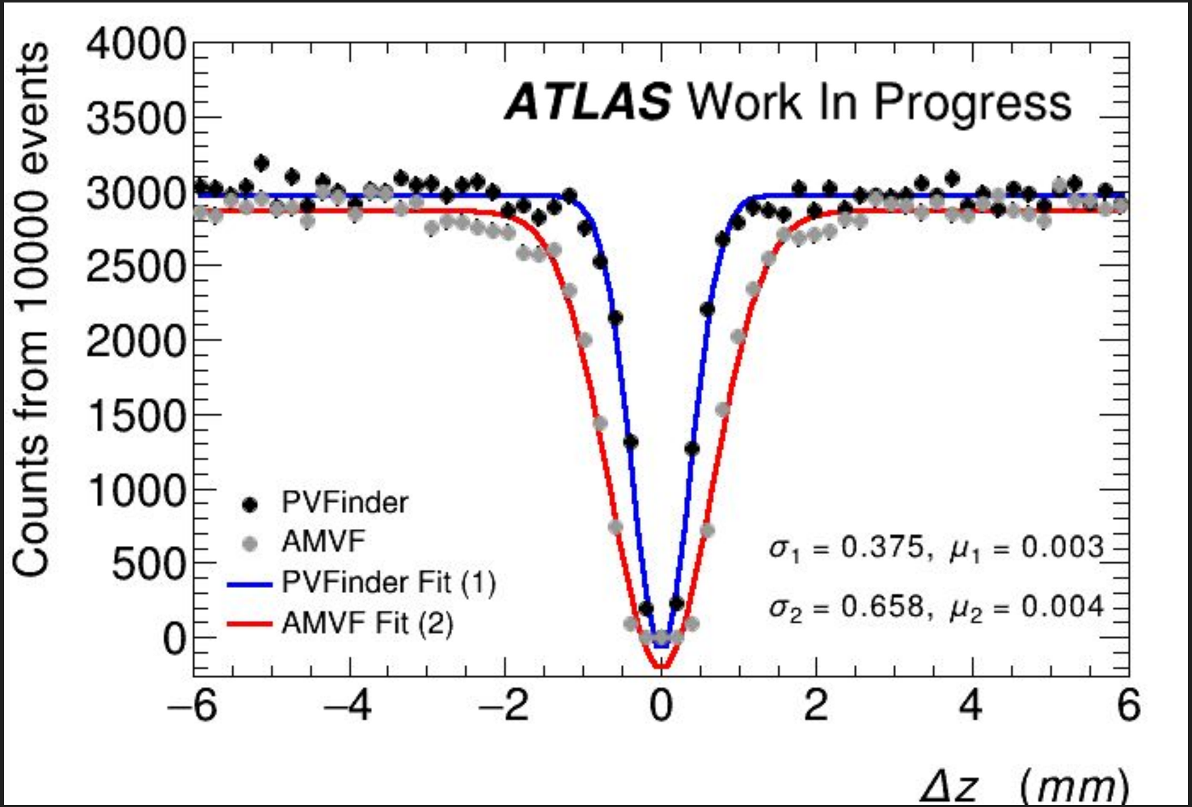
[(2) Ronneberger, Fischer, & Brox (2015)](2)

Training Results

ATLAS Run 2 ttbar sample (40000 training/10000 validation)



Vertex-Vertex Resolution (Comparison to AMVF)

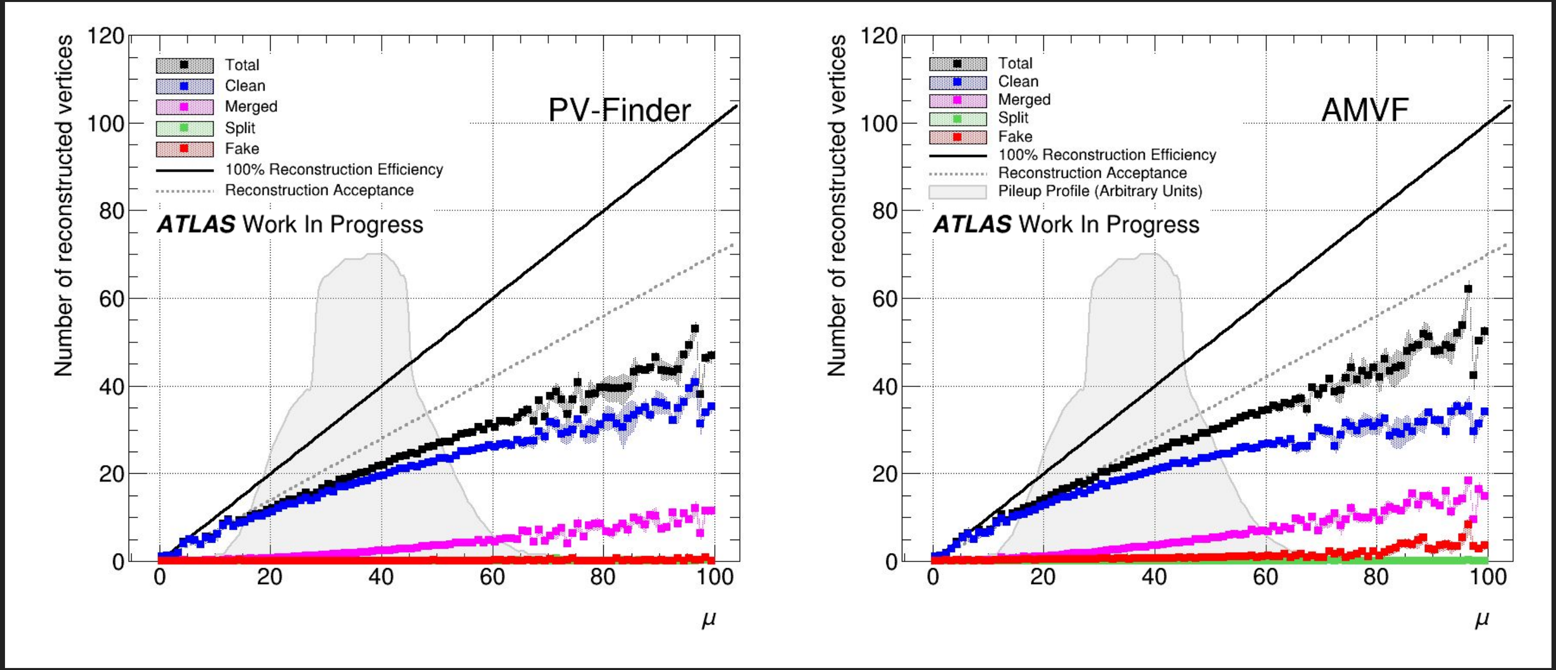


Classification Scheme

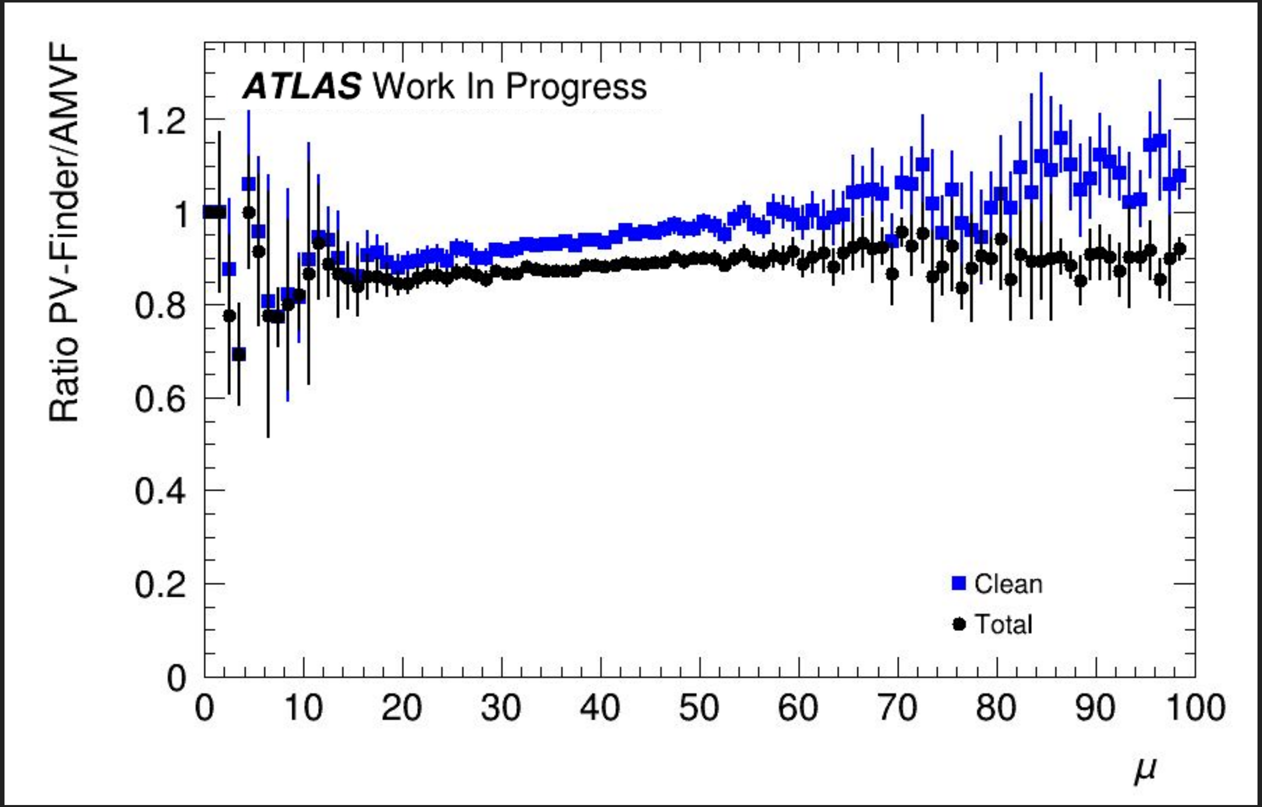
Developed to provide comparison to standard ATLAS method (AMVF)[3]

1. Iterate through reconstructed vertex positions
    a. Find list of truth vertices with a *z*-position within $\sigma_{\text{vtx-vtx}}$ of the reconstructed vertex *z*-position
    b. If this list is empty, the reconstructed vertex is classified as **fake**
    c. If this list has one entry, the reconstructed vertex is classified as **clean**, and that truth vertex is assigned to the reconstructed vertex
    d. If this list has more than one entry, the reconstructed vertex is classified as **merged** and the truth vertices are assigned to the reconstructed vertex
2. Iterate through the truth vertex assignments
    a. If a truth vertex has more than one assignment to a **clean** reconstructed vertex, then all but the closest reconstructed vertex are reclassified as **split**.

Number of Reconstructed Vertices (PV-Finder vs AMVF)

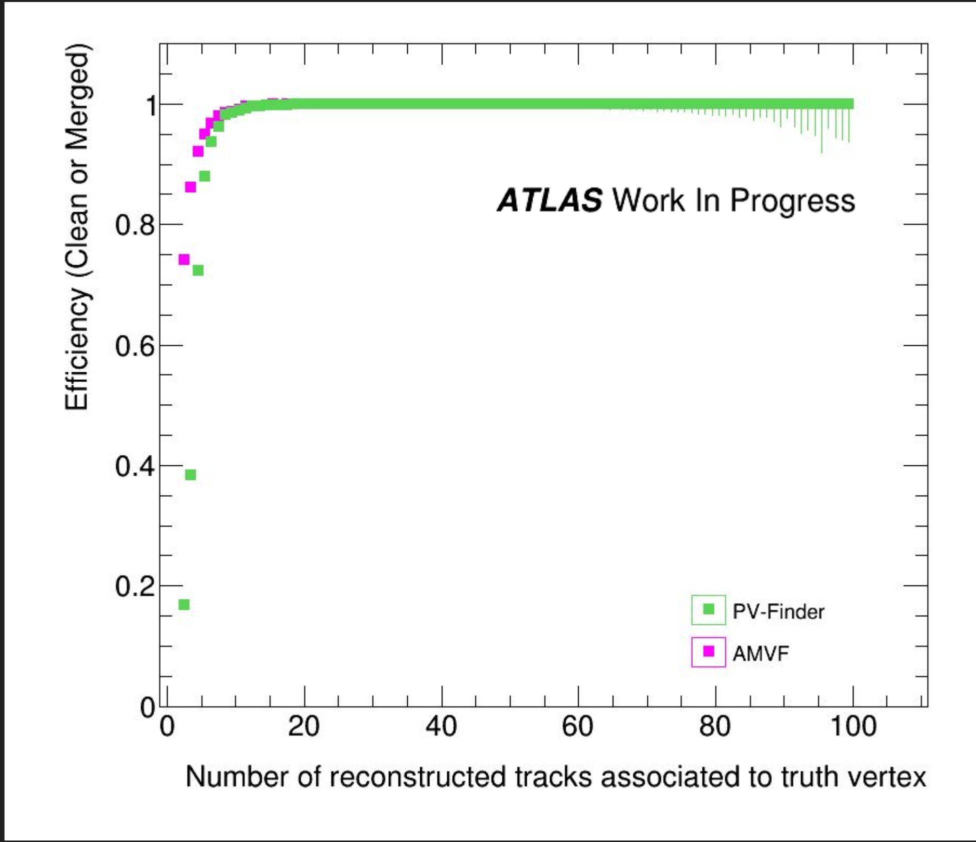


Ratio of the Number of Reconstructed Vertices (PV-Finder to AMVF)



Efficiency as a function of number of associated truth-matched reconstructed tracks

Efficiency = (# clean + # merged) / (total)



Conclusions

- PV-Finder appears to achieve better vertex-vertex resolution than AMVF
- AMVF performs better for PVs with lower track multiplicity
- Under this new classification scheme, PV-Finder produces more clean reconstructed vertices at high pileup
- Future Studies
  - Fine-tune neural network parameters and input features to increase performance
  - Train on lower-multiplicity tracks to try to increase performance for PVs low track multiplicity
  - Use PV-Finder output as seeds for assigning reconstructed tracks to reconstructed PVs
    - Necessary for physics analysis
    - Will allow a better comparison to AMVF (can use their classification method)

Acknowledgements

I would like to thank the following individuals for their contributions to this project: Rocky Bala Garg (Stanford University), Lauren Tompkins (Stanford University), Mike Sokoloff (University of Cincinnati), Simon Akar (University of Cincinnati), Henry Schreiner (Princeton University), Michael Peters (University of Cincinnati), Will Tepe, and Rida Shahid (DIANA Fellow).

This work was supported by the National Science Foundation under Cooperative Agreement OAC-1836650 and under Award OAC-1450319.

References

(1) Progress in developing a hybrid deep learning algorithm for identifying and locating primary vertices, S. Akar, G. Atluri, T. Boettcher, M. Peters, H. Schreiner, M. Sokoloff, M. Stahl, W. Tepe, C. Weisser and M. Williams, EPJ Web Conf. 251 04012 (2021) (08 Mar 2021).

(2) U-Net: Convolutional Networks for Biomedical Image Segmentation, O. Ronneberger, P. Fischer, and T. Brox, arXiv:1505.04597 (accepted at MICCAI 2015) (18 May 2015).

(3) ATLAS Collaboration, *Development of ATLAS Primary Vertex Reconstruction for LHC Run 3*, ATL-PHYS-PUB-2019-015, 2019

Other PV-Finder Papers

- An updated hybrid deep learning algorithm for identifying and locating primary vertices, S. Akar, T. J. Boettcher, S. Carl, H. F. Schreiner, M. D. Sokoloff, M. Stahl, C. Weisser, M. Williams, arXiv:2007.01023 [physics.ins-det] (Submitted to CTD2020) (02 Jul 2020).
- A hybrid deep learning approach to vertexing, R. Fang, H. Schreiner, M. Sokoloff, C. Weisser and M. Williams, J.Phys.Conf.Ser. 1525 012079 (2020) (19 Jun 2019).