Advances in developing deep neural networks for finding primary vertices in proton-proton collisions at the LHC

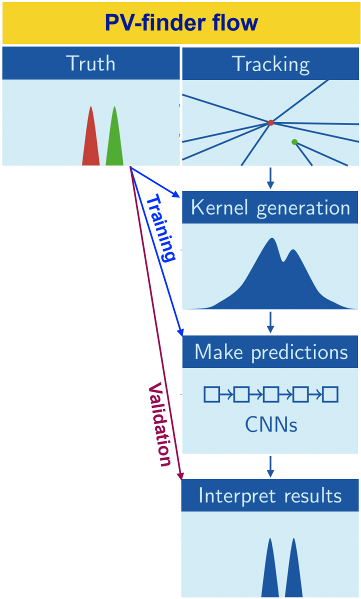
S. Akar[1], M. Elashri[1], R.B. Garg[2], E. Kauffman[3], M. Peters[1], H.F. Schreiner [3], M.D. Sokoloff[1], W. Tepe[1], L. Tompkins[2]

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

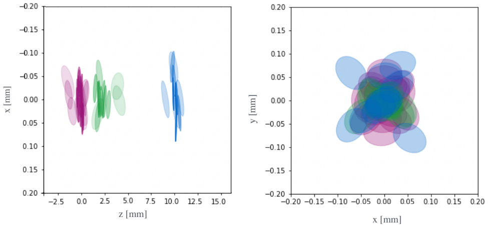
CHEP-2023, May 9, 2023

PV-finder: Key Concepts

hybrid algorithms use deep neural networks to predict primary vertex locations



`poca-ellipsoids`: the positions and error ellipsoids at tracks' positions of closest approach to the beamline. These are used to build Kernel Density Estimators along the beamline direction.

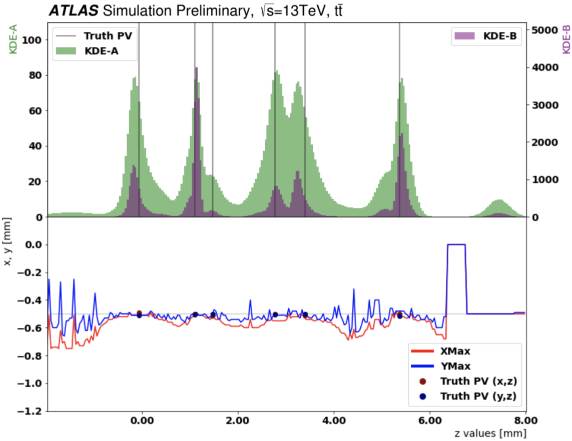


`target histograms`: proxies that are Gaussian distributions whose heights and widths reflect the expected PV resolutions.

CNNs are trained to predict distributions similar to the target histograms.

Heuristic algorithms extract PV positions from the predicted histograms.

Example KDEs for ATLAS data



- KDE-A $\approx \sum$ probabilities
- KDE-B $\approx \sum$ probabilities^2
- XMax and YMax $\approx$ max. probability point in x and y

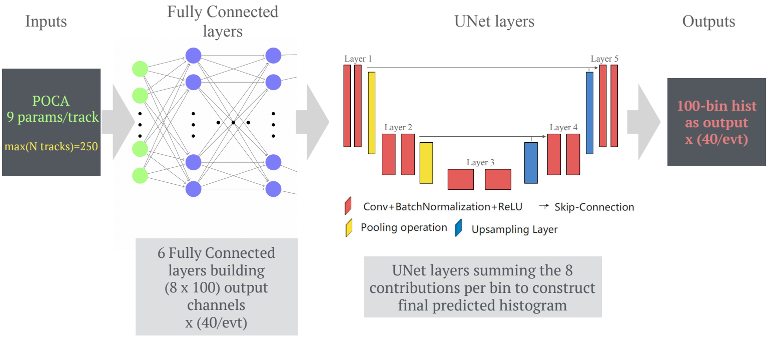
LHCb `kde-to-hist` models use similar KDEs;
better separated PVs due to lower pile-up; worse *z*-resolution.

PV-finder: a new architecture for LHCb

end-to-end DNN, train using 40 $\times$ 10 mm intervals (LHCb simulation)

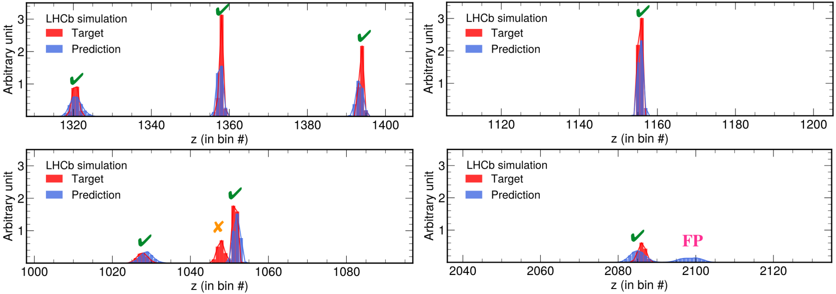
In the architecture illustrated below

- the fully connected layers are first trained as a `tracks-to-kde` model;
- the UNet layers are similar to a `kde-to-hist` model;
- the merged model is trained with all weights floating for FP32 arithmetic.



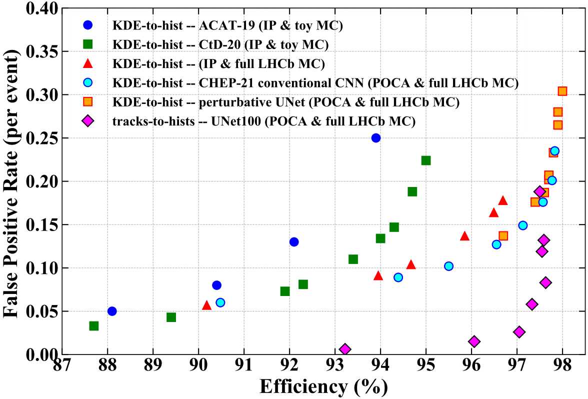
U-Net: Convolutional Networks for Biomedical Image Segmentation

Example Histograms
LHCb



- target histograms are Gaussian distributions with heights and widths calculated from "expected" resolution;
- predicted histograms are produced by the PV-finder inference engine.

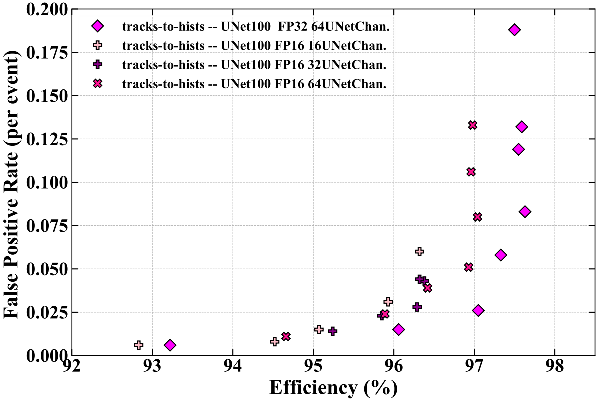
LHCb Run 3 simulation, ≈ 5.5 visible PVs per beam crossing



- The magenta diamonds show the results of using a `tracks-to-hist` deep neural network .AND. tuning the final heuristic algorithm used to select PV candidates.
- This model used here uses FP32 arithmetic and has 64 channels in the first layer of the UNet component.
- The efficiencies and false positive rates shown here adhere to LHCb definitions.
- For details, see ACAT-22 proceedings

PV-finder: LHCb Results using FP16 Arithmetic

LHCb Run 3 simulation, ≈ 5.5 visible PVs per beam crossing



- The magenta diamonds here are the same as shown on previous page.
- The models used for the additional points use FP16 arithmetic and they have 16, 32, or 64 channels in the first UNet layer.
- Note the reduced ranges of both the horizontal and vertical scales compared to the plot on the previous page.

PV-finder: KDEs for ATLAS

analytical approach demonstrating proof of principle of the underlying method

- Track $\rightarrow$ (d_0, z_0, θ, ϕ, $\frac{q}{p}$)
- Track at Point Of Closest Approach (POCA) $\rightarrow$ (d_0, z_0)
- 2x2 covariance matrix:

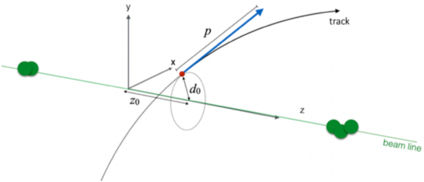
$$\Sigma = \begin{pmatrix} \sigma^2(d_0) & \sigma(d_0, z_0) \\ \sigma(d_0, z_0) & \sigma^2(z_0) \end{pmatrix}$$

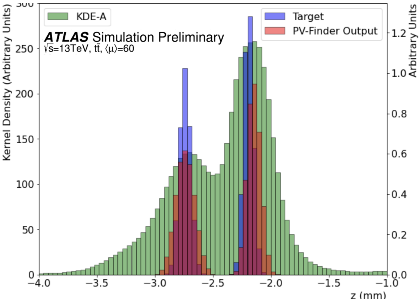
- Probability at a given (x,y,z) = (d, z) point:

$$\mathbb{P}(d,z) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}\Big((d-d_0),(z-z_0)\Big)^T \cdot \Sigma^{-1} \cdot \Big((d-d_0),(z-z_0)\Big)\right)$$

- KDE in a z-bin = Σ_{tracks} $\mathbb{P}(r,z)$ in that bin
- All reconstructed tracks passing quality cuts are considered

ATLAS Coordinate System





CPU-intensive-approach

PV-finder: ATLAS architecture

deeper and wider DNN trained using KDEs from $40 \times 20 \times 20\ \mu m^3$ voxels

- Analytical approach to compute KDEs from reconstructed tracks
- Two DNNs: UNet and UNet++ (UNet with denser skip connections)

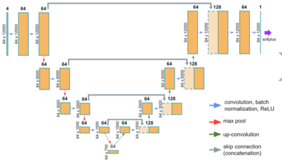
Inputs

Track parameters and their uncertainties (max. (NTracks) ~ 1000)
+
Truth vertices

tracks-to-KDE

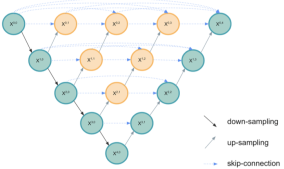
Analytically calculated KDE_A, KDE_B, XMax, YMax (4 1-D histgrams, 12000 bins)
+
Target truth histograms (1-D histogram, 12000 bins

UNet: KDE-to-hists



OR

UNet++: KDE-to-hists

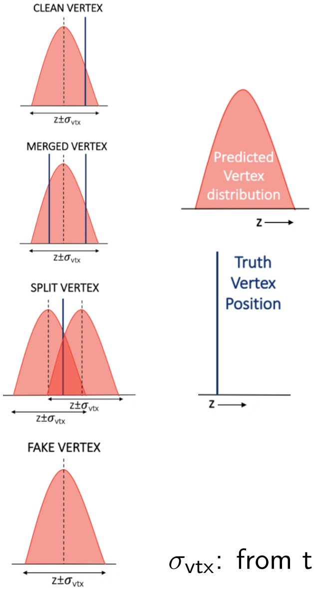


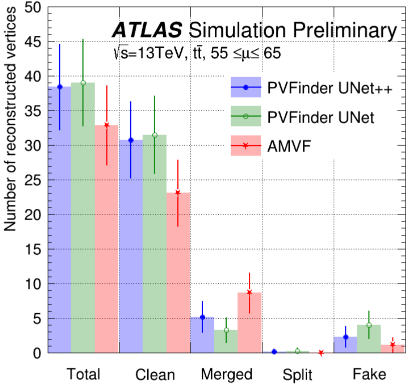
Output

Predicted PVs (1-D histogram, 12000 bins)

Vertex Classification Scheme

for comparison with ATLAS default algorithm, AMVF (Adaptive Multi Vertex Finder)



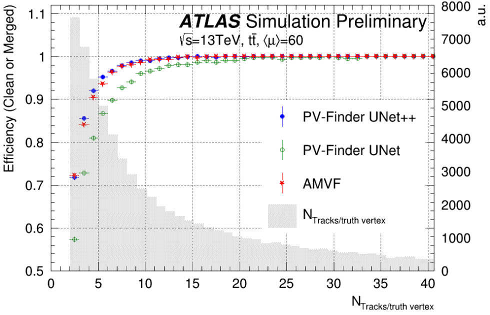


Vertical lines at tops of bars represent the R.M.S. widths of the distributions.

σ_{vtx}: from the z-resolution (next slide)

Efficiency

vertex reconstruction efficiency as a function of number of tracks associated to vertices



Efficiency: Number of truth vertices assigned to reconstructed vertices as "clean" or "merged" divided by the total number of reconstructable truth vertices

False Positive Rate: Average number of predicted vertices not matched to any truth vertex

Model	Efficiency	False Positive Rate (# / event)
PV-Finder UNet++	94.2%	1.5
PV-Finder UNet	88.7%	2.6
AMVF	93.9%	0.8

Resolution in z: PV-finder vs. AMVF

longitudinal separation between pairs of all nearby reconstructed primary vertices

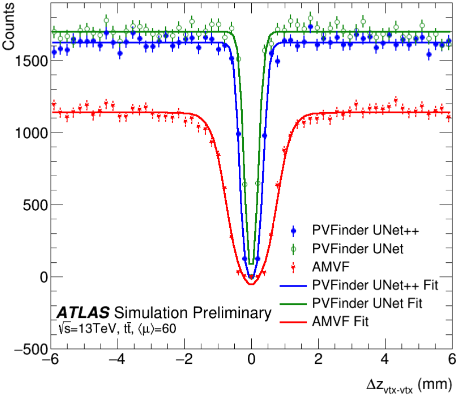
Fit Function:

$$y = \frac{a}{1 + \exp\left(b \cdot (R_{cc} - |x|)\right)} + c$$

where a, b and c are free parameters, R_{cc} is the cluster-cluster resolution, referred to as σ_{vtx}

Method	σ_{vtx} (mm)
PV-Finder UNet	0.23 ± 0.01
PV-Finder UNet++	0.37 ± 0.01
AMVF	0.76 ± 0.02

σ_{vtx}: half-width at the half-depth of the dip



Summary and Future Plans

LHCb:

- We have demonstrated that an "end-to-end" `tracks-to-hist` model performs better than a similar `kde-to-hist` model; using `FP16` arithmetic rather than `FP32` degrades performance slightly, as does reducing the size of the UNet component of the model.
- We plan to instantiate the existing `tracks-to-hist` inference engine inside `Allen`, the GPU-resident first level trigger, as a proof-of-principle;
- We will iterate the `tracks-to-hist` architecture to study performance trade-offs (efficiency vs. false positive rate on one hand versus memory footprint and number of calculations/throughput on the other);

ATLAS:

- We have demonstrated a proof-of-principle `kde-to-hist` model;
  - its vertex resolution is $2\times$ better than that of AMVF;
  - the efficiency and false positive rates are comparable to AMVF;
- This study has been documented in the form of an ATLAS public note;
- We plan to further optimize architectures for ATLAS;
- We plan to implement in ACTS (perhaps with GPU implementation).

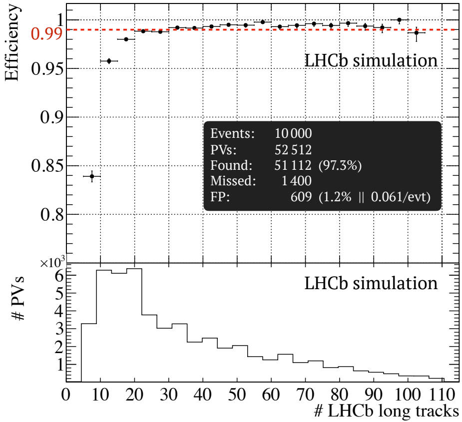
This work was supported, in part, by the U.S. National Science Foundation under Cooperative Agreement OAC-1836650. All of the machine learning training described here was done in `PyTorch` using `nvidia` GPUs.

Backup Material Follows

LHCb: Efficency vs nTracks

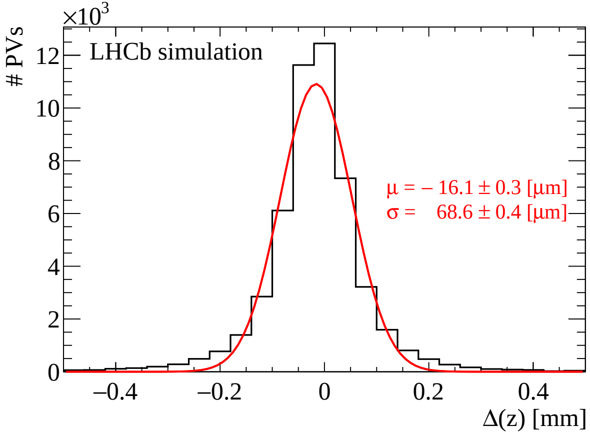
FP32 `tracks-to-hist` model

Efficiency for nTracks > 5 using a “tuned” heuristic algorithm for extracting PVs from target histograms.



LHCb: Observed Resolution & Bias

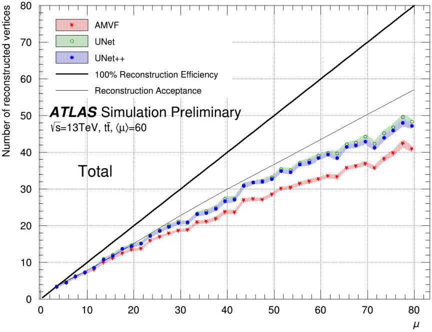
FP32 `tracks-to-hist` model

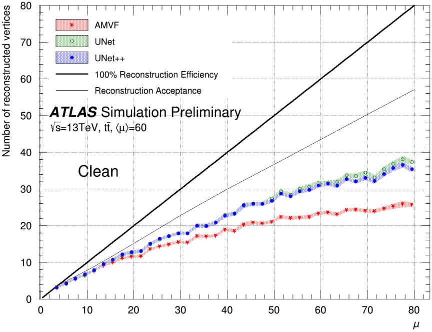


Resolution and bias derived from true PV positions and observed average positions of predicted peaks. This is not comparable to the ATLAS vertex separation resolution.

Vertex Classification

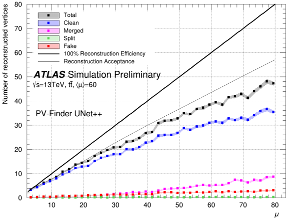
number of reconstructed vertices as a function of pile-up (simultaneous proton-proton interactions at each bunch-crossing)

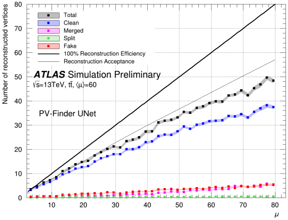


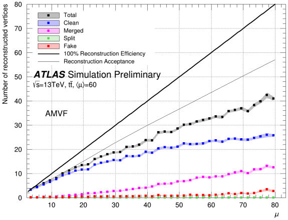


ATLAS: Vertex Classification (Comparison between three approaches

Number of reconstructed vertices as a function of pile-up

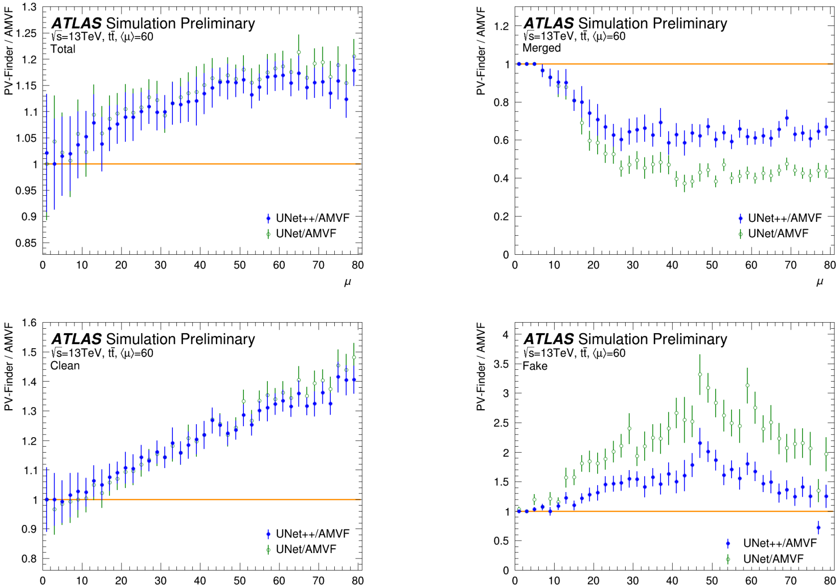






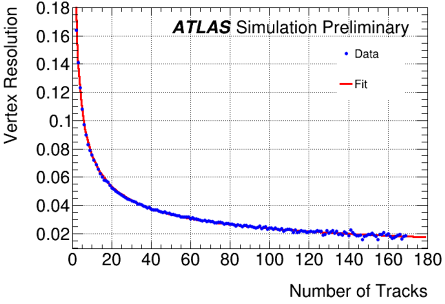
ATLAS: Ratio plots

UNet++/AMVF ratio and UNet/AMVF ratio

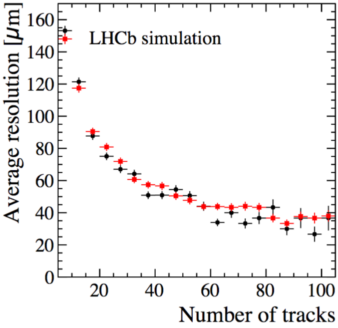


ATLAS & LHCb Expected Vertex Resolution

used for building target histograms



Simulated vertex resolution data with fit used to define target Gaussians. For details, see Fig. 12 in the ATLAS public note.



The red dots correspond to the Run 3 simulated vertex resolution data used to define target Gaussians. See Fig. 28 in the LHCb VELO Upgrade TDR.