DIANA Fellowship Proposal: Adapting a Machine Learning Algorithm for use in the CMS Experiment

PV-FINDER is a hybrid deep learning algorithm designed to identify the locations of proton-proton collisions (primary vertices) in the Run 3 LHCb detector. The underlying structure of the data and the approach to learning the locations of primary vertices may be useful for other detectors at the LHC, including CMS. The algorithm is approximately factorizable. Starting with reconstructed tracks, a kernel density estimator (KDE) can be calculated by a hand-written algorithm that reduces sparse point clouds of three dimensional track data to rich one dimensional data sets amenable to processing by a deep convolutional network. This is called a `kde-to-hist` algorithm and its predicted histograms are easily interpreted by a heuristic algorithm. A separate `tracks-to-kde` algorithm uses track parameters evaluated at their points of closest approach to the beamline as input features and predicts an approximation to the KDE. These two algorithms can be merged and the combined model trained to predict the easily interpreted histograms directly from track information.

The incumbent will work under the joint supervision of Henry Schreiner (a CMS physicist, Princeton) and Mike Sokoloff (an LHCb physicist, Cincinnati) to adapt `tracks-to-kde` algorithms to process CMS data rather than LHCb data. Candidates for this position should have some prior knowledge of scientific Python. Knowledge of deep neural networks and machine learning frameworks such as PyTorch and TensorFlow will be considered favorably, as will be experience with CMS software. The anticipated duration of the project is the three month period May - July, 2021, although there is flexibility related to the start and finish dates.

A timeline with milestones is provided on the next page.

Timeline

weeks 1-2 in parallel, become familiar with descriptions of tracks and primary vertices in simulated CMS data and study the formats of data used in pv-finder

weeks 3-4 in parallel, (i) learn to run existing PyTorch notebooks that train `kde-to-hists` models using “toy Monte Carlo” data sets that simulate LHCb data, (ii) re-format simulated CMS data so it can be read and processed using scientific Python, and (iii) use this data to study characteristics of CMS primary vertices and and tracks;

weeks 5-6 produce KDEs using CMS data and compare these to primary vertex positions to understand what granularity will be required by the `kde-to-hists` models; re-design existing PyTorch models and notebooks to process CMS data;

weeks 7-8 begin to train a first `kde-to-hists` model using CMS data;

weeks 9-11 begin to formally document work done to date and train more sophisticated `kde-to-hists` models.

weeks 12-13 document work done for public consumption and present results publicly.

At the end of the project, the student will present the work done at an IRIS-HEP topical meeting.