UNIVERSITY of WASHINGTON

A novel ML-based method of primary vertex reconstrucion in high pile-up conditions

Haoran Zhao, Alex Schuy, Zhijian Liu, Ke Li, Shih-Chieh Hsu, Scott Hauck, Phil Harris

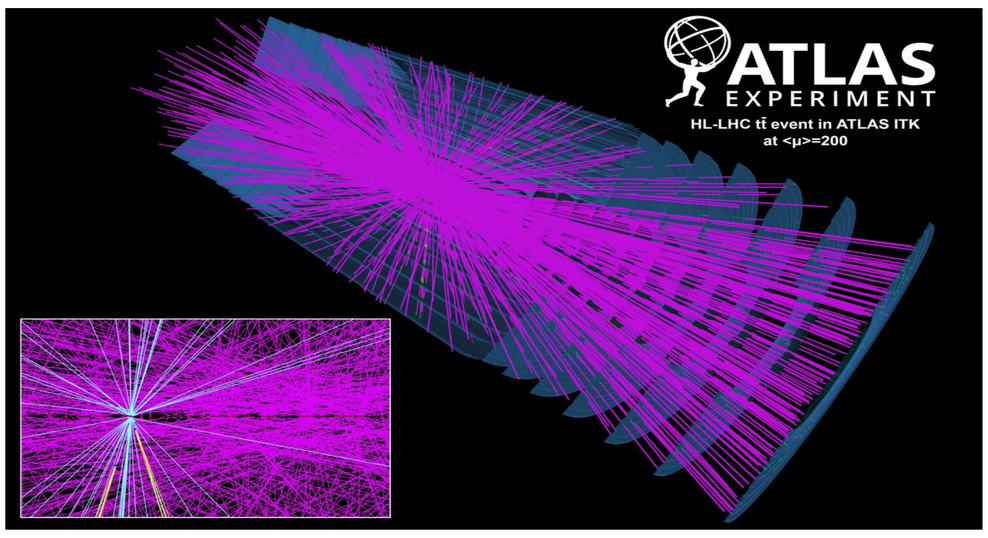
Introduction

A feasibility study of applying Sparse Point-Voxel Convolution Neural Network to do primary vertexing in a dense environment

Motivation: the challenges brought by High Luminosity LHC(HL-LHC)

More pile-up events in HL-LHC:

- Run 3: <μ> ~ 60 – 80
- HL-LHC: <μ> ~ 200 @ 7.5 x 1035 cm-2s-1



A ttbar event display in simulation of HL-LHC condition. Pink lines are tracks and red dots are primary vertices along the z axias. Expected an average of 200 vertices happening in a 25ns time window.

Primary Vertex Reconstruction

Primary vertex: the reconstructed location of an individual particle collision

Importance of primary vertex reconstruction:

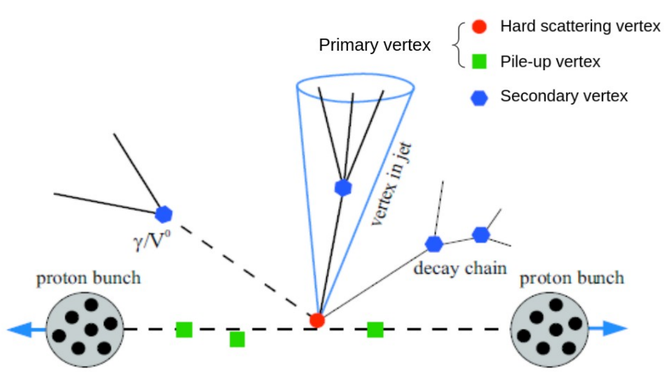
From Physics point of view, vertexing helps to:

- Identify decaying particles from their products (remove pile-up effects)
- Study the properties of those particles (mass, lifetime, couplings, ...)

From the instrumentation point of view, vertexing serves to:

- Determine the luminous region(beam spot)
- Characterize the detector positioning resolution

Vertex topologies in pp collisions



Collisions of two bunches of protons. Primary vertex include the hard scattering vertex(red round) and pile-up vertex(greed square).

Vertex reconstruction can be thought of as comprising two components:

Vertex finding: the association of tracks to a particular vertex candidate

Vertex fitting: the reconstruction of the vertex position along with its covariance matrix, estimation of the quality of fit

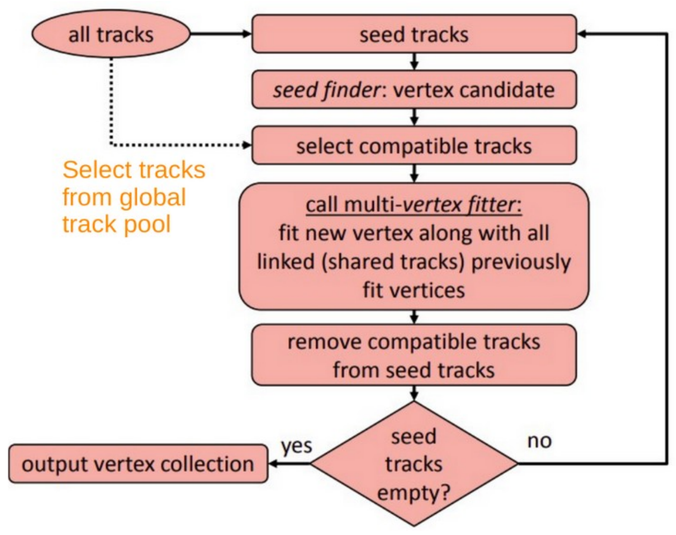
The goal of vertex reconstruction is to turn a set of **reconstructed tracks(recon'ed from tracker)** into a set of **reconstructed vertices,** i.e. clustering the tracks into different primary vertices.

Conventional algorithm: AMVF

Adaptive Multi-Vertex Finder (AMVF) is commissioned as the conventional algorithm at ALTAS, but not optimized for HL-LHC condition

AMVF Input: Track Parameter & Cov Matrix

AMVF Output: reco vertex x, y, z positions and errors, reco fitting quality: chi^2/(num. degree of freedom)



SPVCNN

SPVCNN is a state-of-the-art technique developed for self-driving cars which leverages **space voxelization** and **point-based method** to do clustering problem.

- **Voxel branch**: coarse-grained information
- **Point branch**: fine-grained information

SPVCNN overview

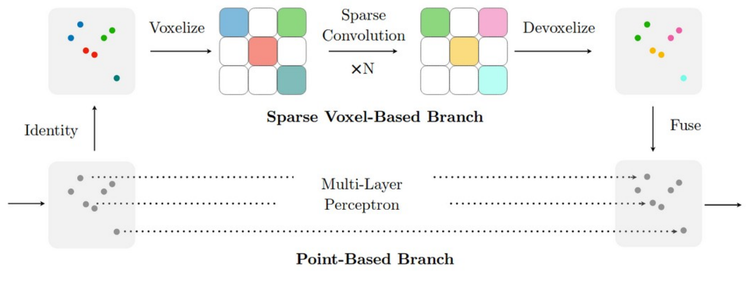
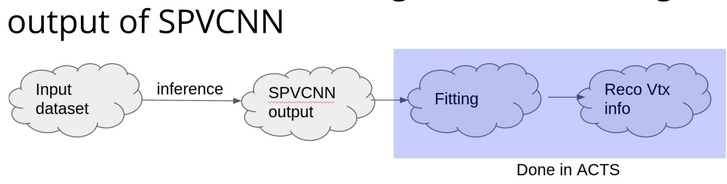


Fig. 2. Overview of Sparse Point-Voxel Convolution (SPVConv): it equips the sparse voxel-based branch with a lightweight, high-resolution point-based branch which can capture fine details in large scenes.

Workflow and dataset

Software: Acts Common Tracking Software

Workflow: do vertex fitting on the clustering output of SPVCNN



Dataset description:

> physics process: Full Hadronic ttbar 14 TeV

> 10k events generated by ACTS using Pythia8 event generation

> tracks are reconstructed by fast track simulaiton based on ACTS GenericML detector topology

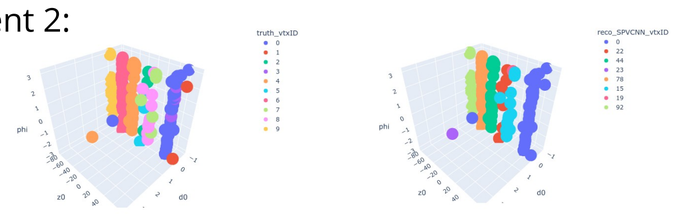
> input features: track parameters

Results

Event display:

Event 1:

Event 2:



Summary

> SPVCNN is a possible ML approach to do primary vertexing

> More tunings of SPVCNN are needed, possible explorations are:

> Coordinates and parameters used to voxelize the space, i.e track coordinates vs. Cartesian, voxel size

> Parameters used in prediction

> Physics performance, e.g. classifications of primary vertex, hard scattering vertex reconstruction efficiency, will be done

> Towards more complicated dataset, e.g. <μ> = 100, 200

Reference:
Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution, arXiv:2007.16100
Salzburger, A., Gessinger. el (2021). A Common Tracking Software Project (Version 10.0.0) [Computer software]. https://doi.org/10.5281/zenodo.5141419