Project Proposal:
Adapting a Machine Learning Algorithm for Enhanced Performance in ACTS

Layan AlSarayra

May 25th, 2023

1 Project Summary

This project aims to lay the foundation for adapting a machine learning algorithm for primary vertex (PV) identification within ACTS framework to improve the efficiency and accuracy of the PV identification. The Objective is to generate Kernel Density Estimation (KDE) code and prepare the output to be processed using UNet/UNet++ Neural Network. To assess the performance, the results obtained will be compared to the previously established results from AMVF. PV identification is a crucial step in HEP research; it provides necessary information about particle trajectories and interactions, therefore adapting a machine learning algorithm for it within the ACTS framework has the potential to significantly enhance the efficiency and accuracy of the identification process. By leveraging available resources, improvements in pattern recognition and features detection our work will directly impact the quality of analysis and interpretation of data. An important consequence of this project is that the methods and techniques developed will have broader applicability to be used in other scientific fields that involve tracking and vertex reconstruction beyond the specific domain of HEP. This work contributes to the continuous development and improvement of ACTS, enabling researchers to benefit and build on the developed code.

2 Work Plan

During the 12-week time interval I plan to achieve 4 main goals:

1. Running ACTS and producing all tracks and vertices in the events. This includes setup and configuration for ACTS on my environment, gathering the input data, event reconstruction by analyzing tracks and vertices, extracting the track parameters at the point of closest approach (POCA), and validating the result.

2. Generate KDEs by transforming the data obtained from the POCA and transforming the result into a suitable format for Machine Learning algorithms. This is done by applying Gaussian kernel functions to each data point or track parameter value.

3. Running UNet/UNet++ on the generated KDEs. The end product should contain files like: UNetPrimaryVertexFinder.hpp.

4. Write up a documentation that contains the work done, publish and source the code, and compile used sources.

3 Deliverables

Software deliverables include developed code in C++ and ROOT for ACTS on lxplus using any required libraries for our objectives,as well as sourcing the code on Git/GitHub. Together with the support of my mentors and supervisors Rocky Bala Garg, Lauren Tompkins, Bastian Schlag, we will compile a tutorial that contains software setup instructions, implementation details, and usage guidelines for the developed code. By the end of this phase, we aim to have a well-documented and thoroughly tested project, ready for future contributions in the field of data analysis and machine learning for HEP.

4 Timeline

Week	**Tasks**
1-2	Set up the development environment, familiarize myself with ACTS, and generate tracks and vertices data.
3-4	Implement the code to write tracks to POCA code (and validate).
5-6	Develop the POCA to KDE code (and validate).
7-8	Run UNet/UNet++ on the generated KDEs and compare it with respect to the previously generated result.
9	Compare ACTS UNet/UNet++ implementation with ACTS AMVF implementation.
10-11	Finalize the documentation, tutorials, and code contributions. Review and polish the project's paper.
12	A contingency period as a buffer to account for potential setbacks that may arise during the project.