Improving the Analysis Grand Challenge (AGC) Machine Learning Workflow

Con Muangkod
University of Colorado Boulder

Mentors: Elliott Kauffman[1], Alexander Held[2], Oksana Shadura[3]
[1]Princeton University, [2]University of Wisconsin-Madison, [3]University of Nebraska-Lincoln

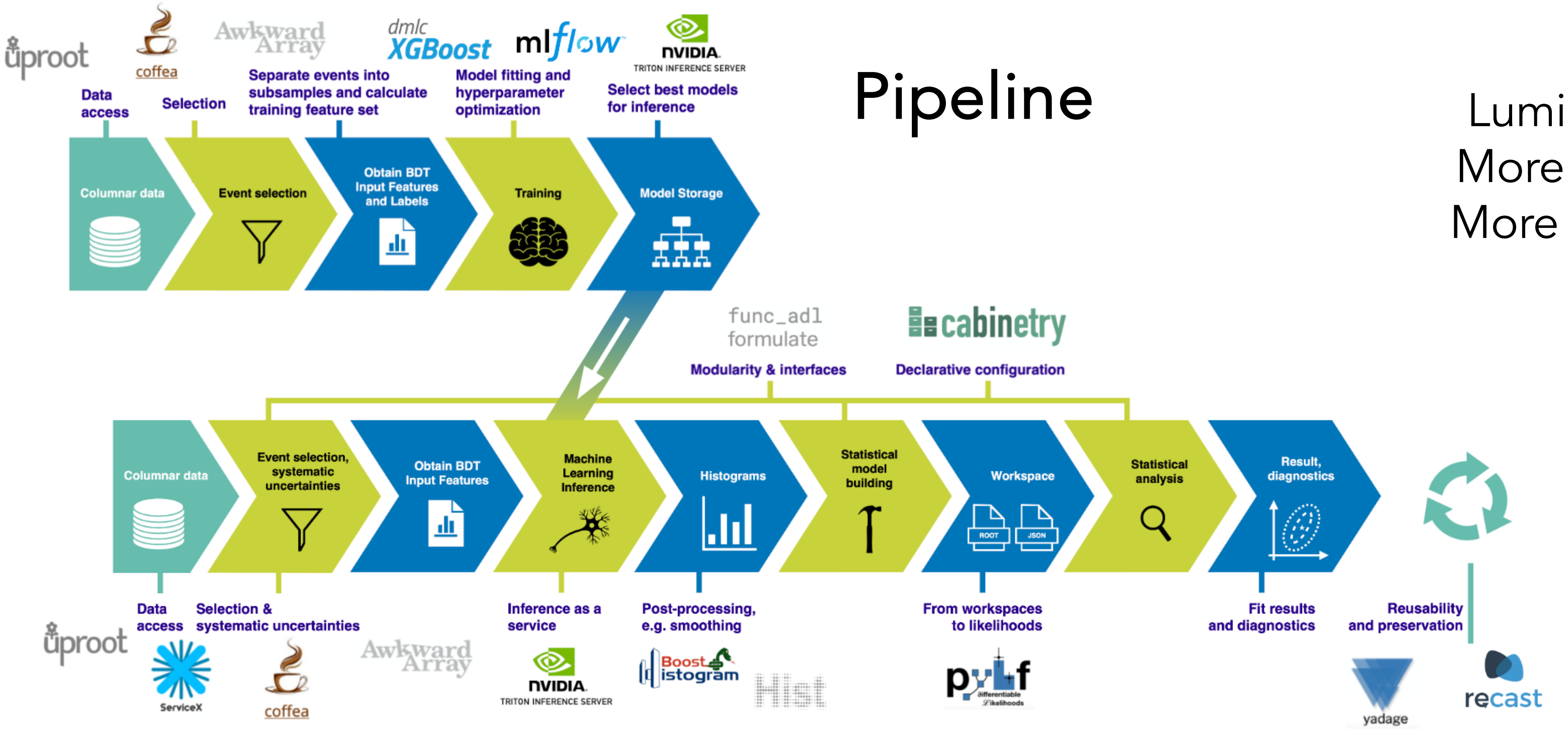
July 3rd, 2024

IRIS-HEP Undergraduate Fellow Program 2024

Analysis Grand Challenge (AGC)

AGC aims to investigate, develop, and improve the end-to-end analysis workflow, preparing for the **High-Luminosity LHC**. There are several tools and packages implemented to improve user experience, such as, the interactive interface, data access, event selection, histogram, statistical model, and interpretation.

Pipeline



Data access

Selection

Separate events into subsamples and calculate training feature set

Model fitting and hyperparameter optimization

Select best models for inference

Columnar data → Event selection → Obtain BDT Input Features and Labels → Training → Model Storage

Modularity & interfaces

Declarative configuration

Columnar data → Event selection, systematic uncertainties → Obtain BDT Input Features → Machine Learning Inference → Histograms → Statistical model building → Workspace → Statistical analysis → Result, diagnostics

Data access

Selection & systematic uncertainties

Inference as a service

Post-processing, e.g. smoothing

From workspaces to likelihoods

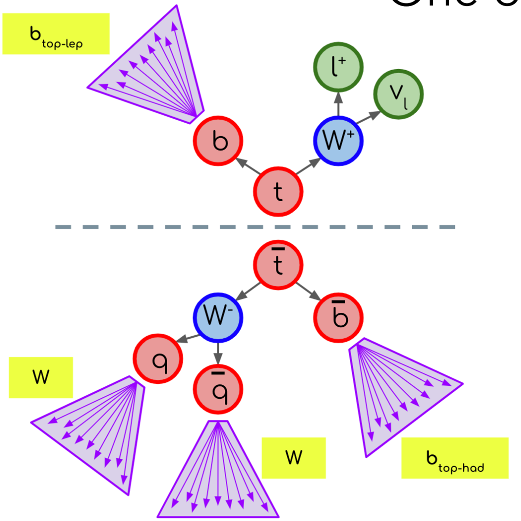
Fit results and diagnostics

Reusability and preservation

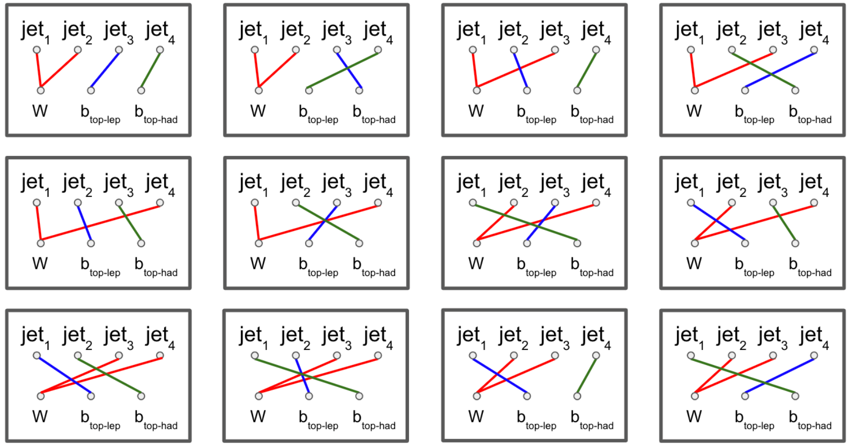
HL-LHC

Luminosity increased by 10
More events & backgrounds
More chance to observe rare phenomena

Semi-Leptonic ttbar production: Jets patron assignment



One of the common productions in proton-proton collision.



12 permutation for an event with 4 jets

Non-ML

Use **ak.combination** to construct tri-jets systems across jets permutation. Calculate combined pT of trijet, the highest pT candidate used to reconstruct top quark mass. Note: no prior jets assignment, the trijet is 2W and b_top from the hadronic side.

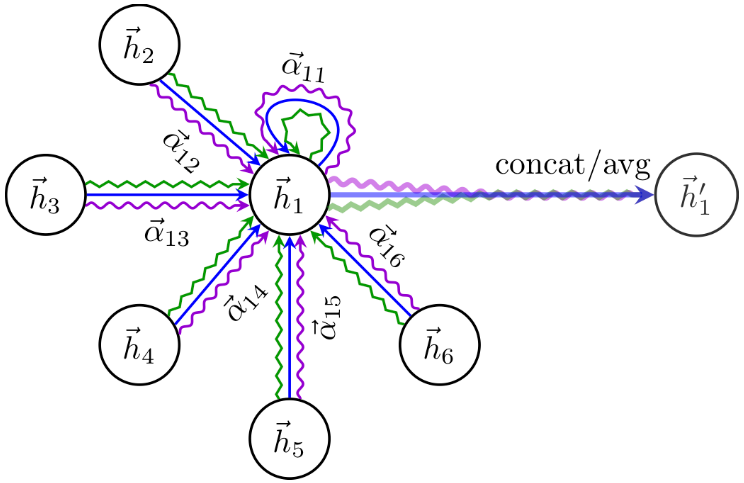
Machine Learning (ML) implementation

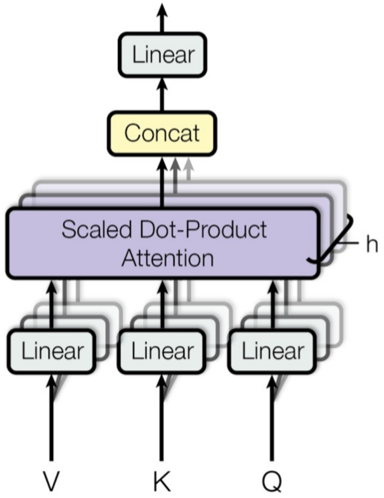
Use **Boosted Decision Trees** (BDT) for predict label classes and match it with the truth table. Consider all permutations in each event, the highest score give correct jet assignment. 20 features are used for the training; pT, btagCSVV2, qgl, combined mass, combined pT, and deltaR.

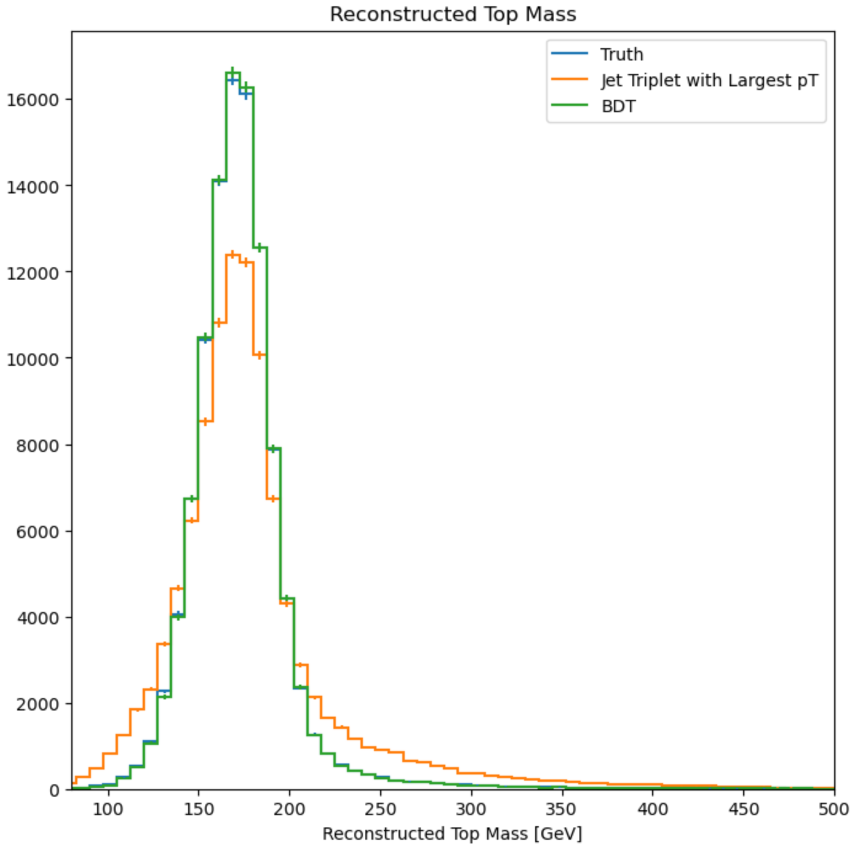
Graph Neural Network (GNN)

Another technique of representing HEP data as a graph with nodes and edges.
For the ttbar analysis, the nodes could be jets across events and edges connection would represent jets that are in the same event. Node features include pT, mass, eta, phi, btagCSVV2, qgl, jet-lepton combined mass, etc. Edge features, such as, deltaR, combined pT and mass can be implemented (with computational cost)

Note: jets across events are flattened. No need to consider every permutation like BDT. Less computational expensive







Multi-head attention for independent parallel computation.

Objectives

- The main goal is to implement the GNN technique into the pipeline.
- More complex GNN architectures can investigated to reflex physics phenomena.
- We need more robust and refine model for the future analysis which could be more computational expensive. Using GNN is a start.

Machine-Learning Tools

Scikit-Learn: a library built in python, providing models and metrices used in the learning algorithm.

XGBoost: parallel tree boosting to build model and optimization

DASK: scalability of machine learning, distributed and parallel training and prediction.

NVIDIA Triton Inference: inference infrastructure for model deployment. It allows fast and scalable workloads.

Pytorch: the main framework used for building GNN architecture. It helps with preprocessing data, defining layers and operations