[1] The University of Chicago

[2] University of Michigan

[3] Fermilab

Towards a Self-Driving Trigger: Adaptive Response in Real Time

Yuxin Chen, Zixin Ding, Shaghayegh Emami[2], Abhijith Gandrakota[3], Christian Herwig[2], David Miller, Jennifer Ngadiuba[3], **Giovanna Salvi[2]**, Cecilia Tosciri, Nhan Tran[3]

FastML 2025

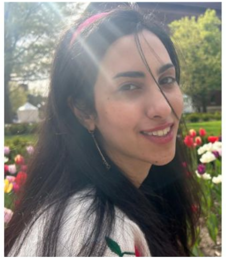
Who are we?

Team:

Yuxin Chen, Zixin Ding, Shaghayegh Emami, Abhijith Gandrakota, Christian Herwig, David Miller, Jennifer Ngadiuba, Giovanna Salvi, Cecilia Tosciri, Nhan Tran

THE UNIVERSITY OF CHICAGO UNIVERSITY OF MICHIGAN Fermilab



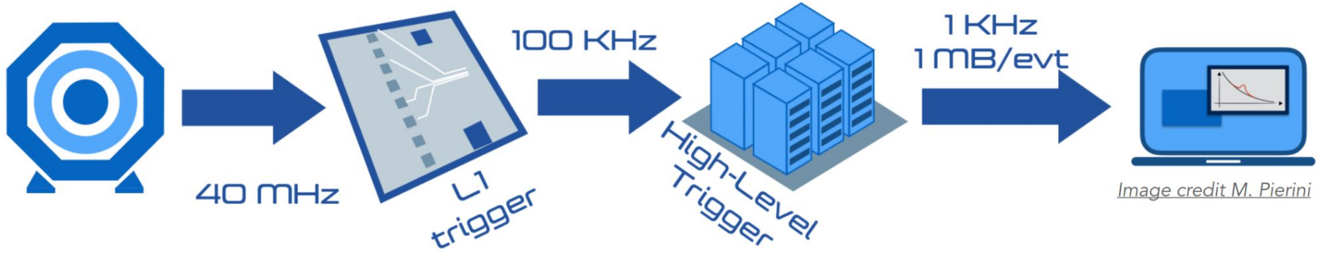


This work is supported by the 2023 Joint Task Force Initiative




Upcoming paper on an OpenData-based ecosystem

Motivation: the trigger challenge



CMS & ATLAS produces more data than we can handle

Only a tiny fraction of pp collision events are interesting enough to be recorded.
Only 0.25% acceptance in the first stage!

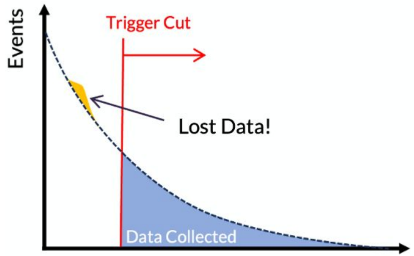
- Two-level trigger to capture interesting physics
- Need quick decisions, large data rate Must decide fast, under enormous data
- First-level acceptance: just 0.25%!
- Goals of a trigger:
  - Keep event rates manageable
  - Enhance sensitivity to BSM / rare SM processes

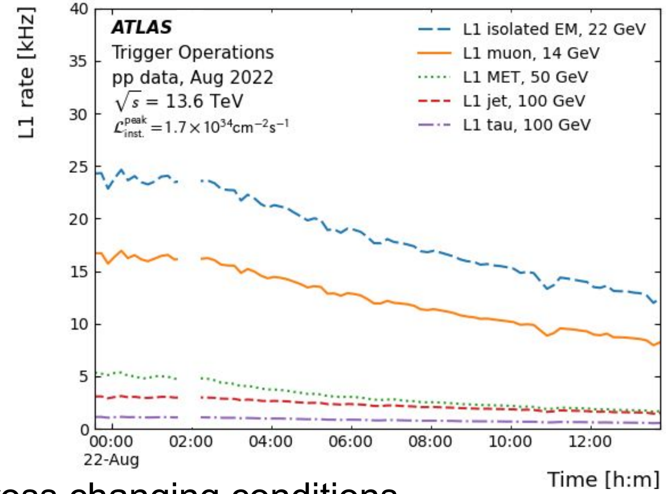
Why do we need an autonomous trigger system?

- Fixed Menus are based on prior knowledge → based on cuts on physics signatures (e.g., MET > 50 GeV)
  - Subject to biases: are we missing new physics?

- Static **thresholds**→not responsive to changes of: *pileup, detector calibrations, non-collision backgrounds*

- Cuts are tuned for worst-case (e.g.high pileup)
  → too tight under normal conditions
  → this cause physics loss, inefficient data taking

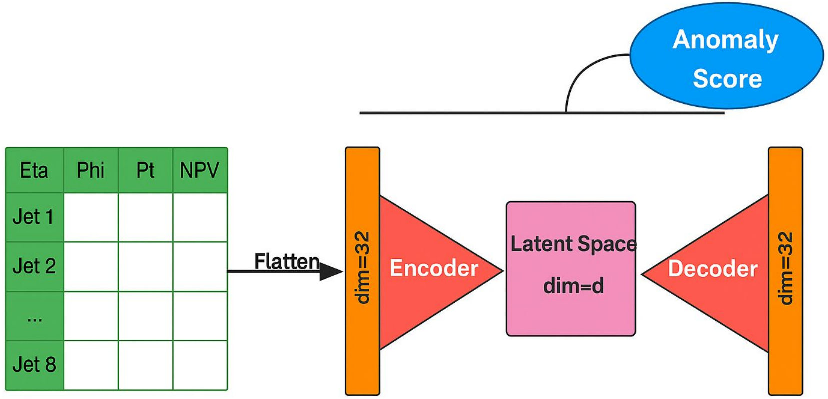




OUR GOAL:

Trigger that can adapt in real-time, maintaining high efficiency across changing conditions, able to take cost-aware trigger decisions.

Trigger Strategy & Dataset

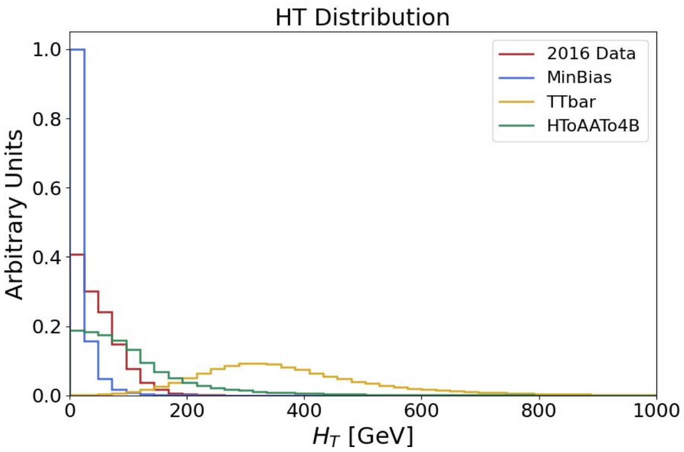


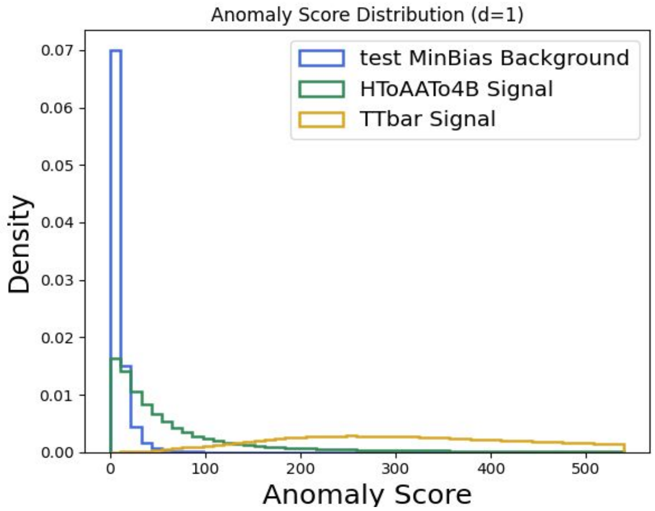
Two trigger inputs in a dual-path trigger structure:

- Conventional HT trigger sensitive to pileup
- Anomaly Detection→rare/unexpected signatures

We used Open CMS Data from 2016

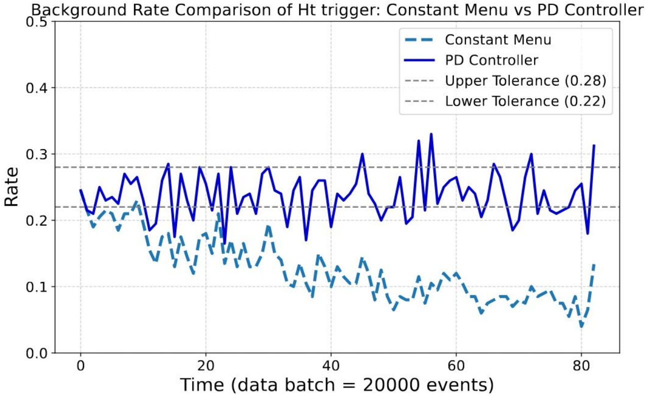
- Signal:
  - SM **ttbar**
  - BSM **H→AA→4b**
- Background:
  - **MinBias** (MC)
  - **ZeroBias** (data)

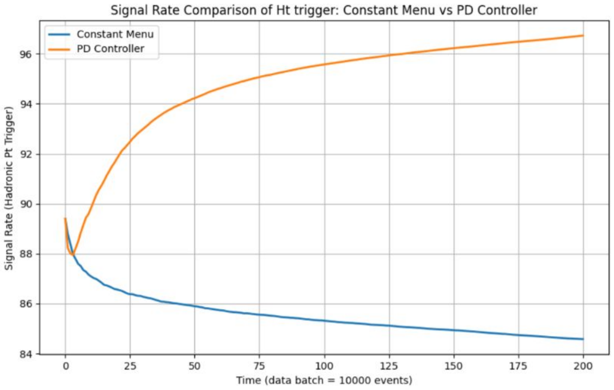


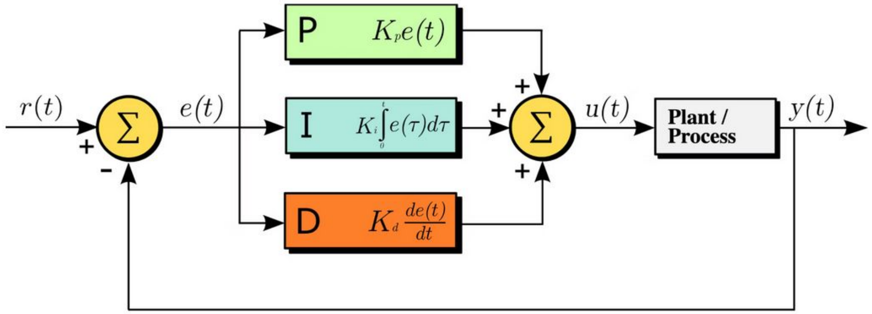


First Step toward Control: PID loop

PID loop can control one trigger at the time. Let's take HT for example. Tradeoffs: Signal efficiency, Bckg control.







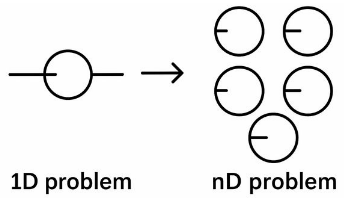
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t)$$

The PID controller computes corrections based on the deviation *e(t)* from a target value:

- keeps the bkg rate very near the target
- allows us to collect way more signal

Second Step: from 1D → ND Trigger Control

- **Problem:** Can no longer use use a simple PID solution anymore
    - need to dynamically adjust all thresholds, minimize in *n* dimension



- **Solution:** Simplify to **Cost Function** with set of constraints f_i and allowed deviations σ_i

$$C = \sum_{i=1}^{N} \frac{f_i}{\sigma_i}$$

Case 1: Maximize specific signal efficiency with fixed target

Case 2: Allocate bandwidth for specific trigger paths

Case 3: Minimize computation cost of triggers

- Perform local grid search on previous set of batches (lumi sections)
    - Find optimal parameters for the subsequent batch
    - We benchmark against an ***"ideal"*** controller that optimizes using future data.

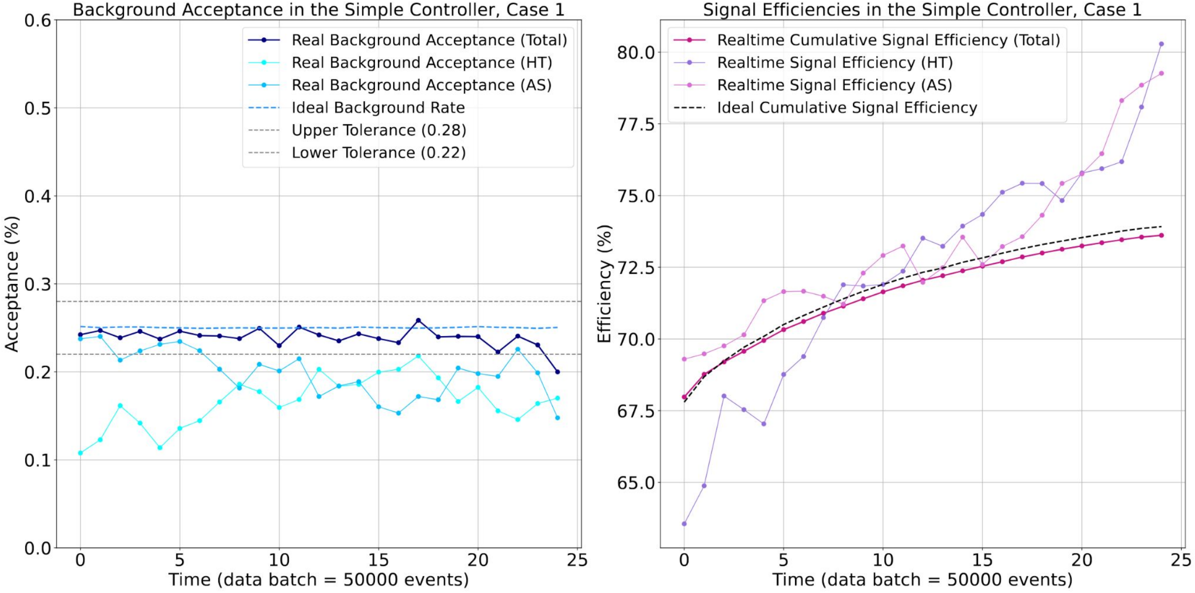
Case 1: Standard cost function

- Strategy relies on background-signal tradeoff→Maximize signal efficiency with constant data taking rate

$$\boxed{C_1 = w_0|r_b - t| + w_1(100 - r_s)}$$

Where r_b= total bckg rate; t= target rate; r_s= total signal efficiency w_0=100; w_1=0.2

- Signal efficiency improves over time, while the background rate remains constant

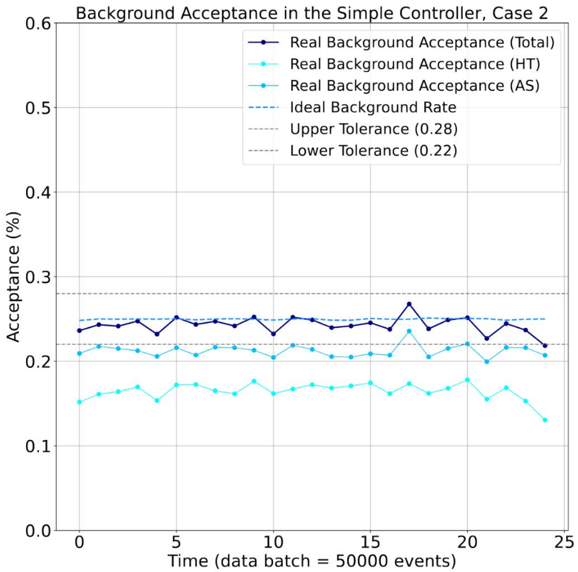


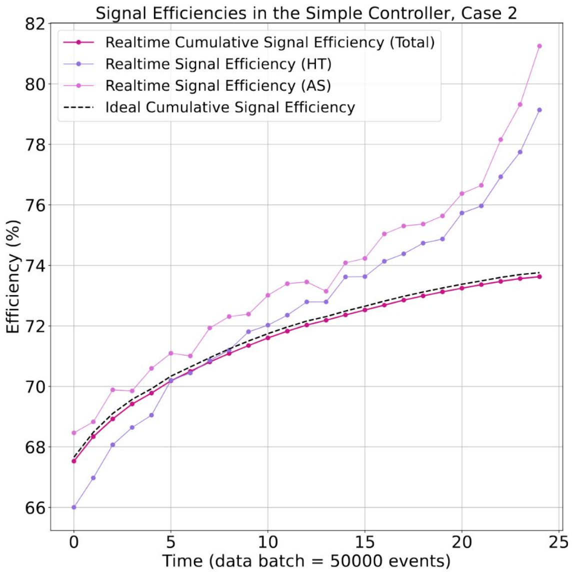
Case 2: Anomaly focused cost function

- Exclusive rate for Anomaly detection, while requiring high performance on well understood signal

$$C_2 = w_0|r_b - t_b| + w_1|r_{t\bar{t}} - 90| + w_2|r_{\mathrm{AS}}^{(\mathrm{ex})} - p \cdot t_b|$$

- where r_{tt}= total efficiency of tt, r_{AS}(ex)= exclusive rate for anomaly detection, p= fraction of allocated bandwidth. Here w=[100,0.2,25] and p=0.3





Case 3: Computationally focused cost function

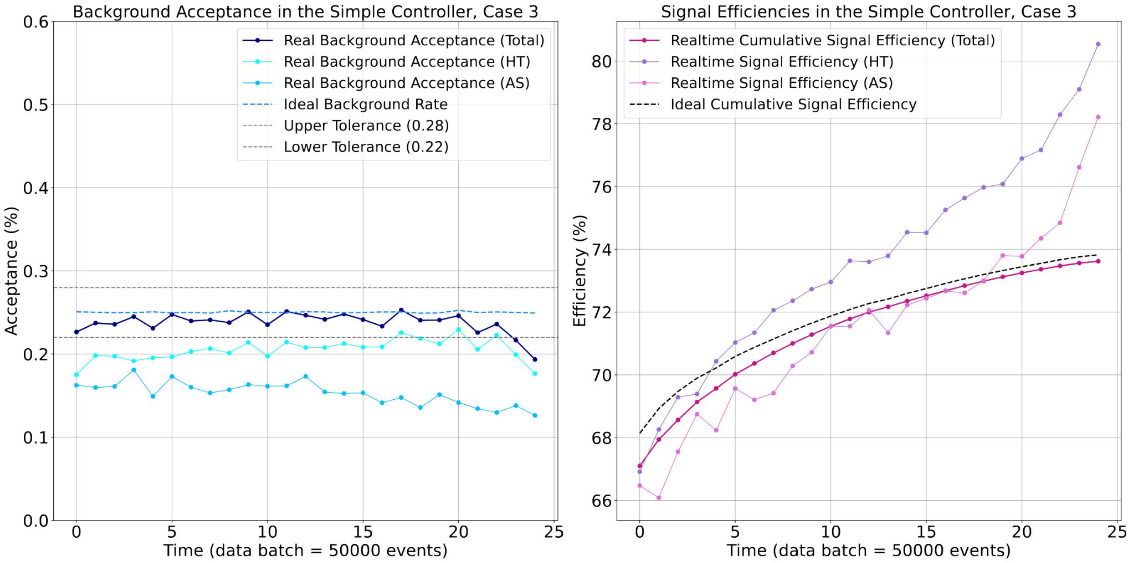
$$C_3 = w_0 |r_b - t_b| + w_1 |r_s - 100| + \sum w_i \cdot f_{comp,i}$$

Two components included:

- Event-level computational cost: reflects the complexity of the event (quantified by jet multiplicity)

- Trigger-path-level computational cost:
  - Each trigger path activates different algorithms at HLT with different CPU cost,
  - We assign each path a representative cost.

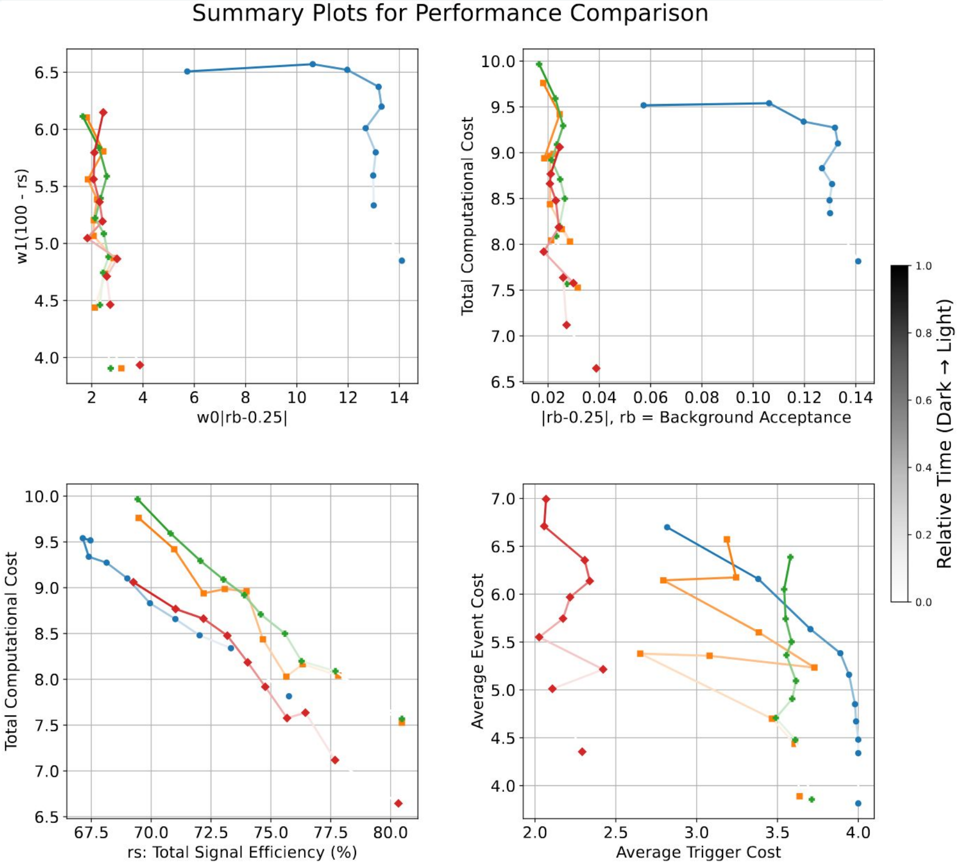
  w = [100, 0.2, 2, 2]



Discourages configurations that unnecessarily exploit expensive trigger paths

Comparison of different strategies

- **Fixed Menu**
  - Progressive degradation
- **Stable Cost Agent**
  - Stable rate with increasing signal efficiency
- **Anomaly Focused Agent**
  - Higher performance, but higher computational cost
- **Computational Focused Agent**
  - Improved performance with optimized resource utilization



Summary

- Current trigger systems don’t
    - adapt to changing detector conditions
    - take action without human intervention

- We implemented a step-by-step approach to develop a self-driving trigger framework that:
    - integrates traditional and anomaly detection that can expand the physics reach by targeting rare or unconventional signatures.
    - real-time control strategies: dynamically adjust trigger thresholds and allocations in response to changing conditions

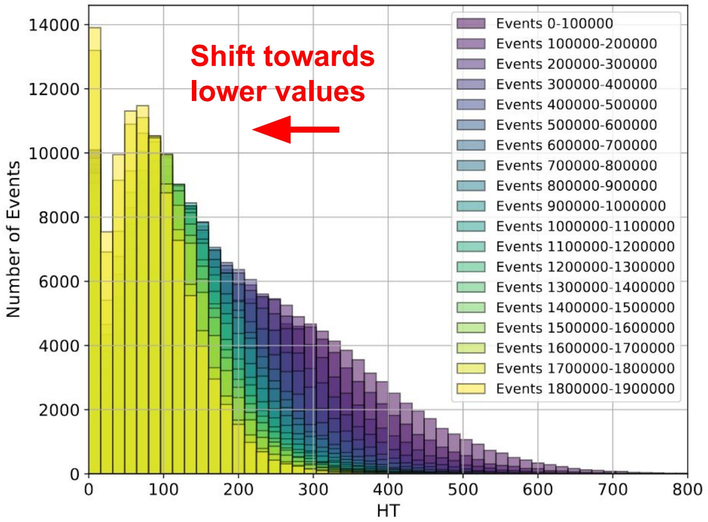
- Next steps:
    - scale to more trigger paths and test it in real time
    - Introduce an agent using reinforcement learning to adjust thresholds dynamically!

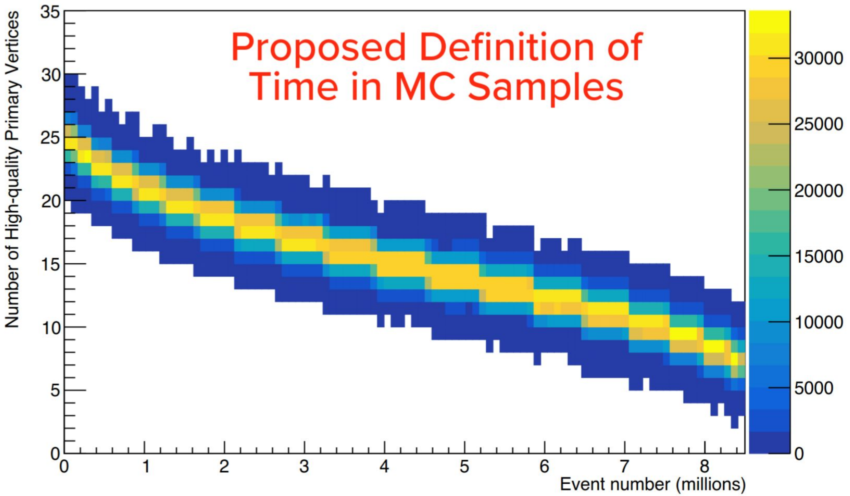
Stay Tuned!
Paper coming out soon!

THANK YOU FOR THE ATTENTION!

Time information

- Rates exponentially decay with decreasing luminosity during a LHC fill (also reflected in HT distributions)
- Since timing information doesn't exist in MC samples, to emulate this behavior, we sorted events based on NPV and smeared the values to give them a slight noise.





TRAP

TRAP prepare chains with different thresholds: 50, 40, 30 GeV, that start with high prescales and are progressively de-prescaled as the luminosity decreases.

AUTOENCODER

Encoder: Takes the input data and compresses it into a smaller, lower-dimensional representation, latent space.
Decoder: Takes that compressed representation and tries to reconstruct the original input as closely as possible.

Anomaly score: reconstruction error (MSE) of jets/event features.

Comparison with Fixed Menu