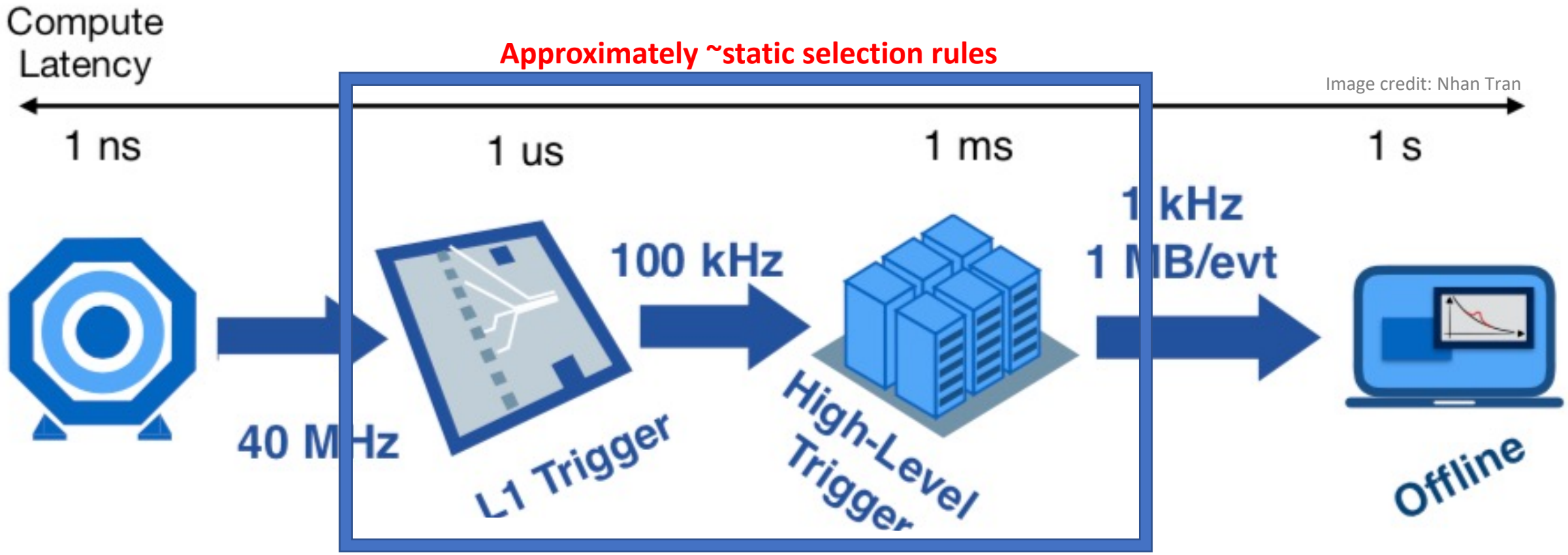
What would it take to build a self-driving trigger and DAQ system?

David W. Miller, University of Chicago

Please also see the more detailed and contextual talk last Friday:

https://indico.fnal.gov/event/46746/contributions/210791/

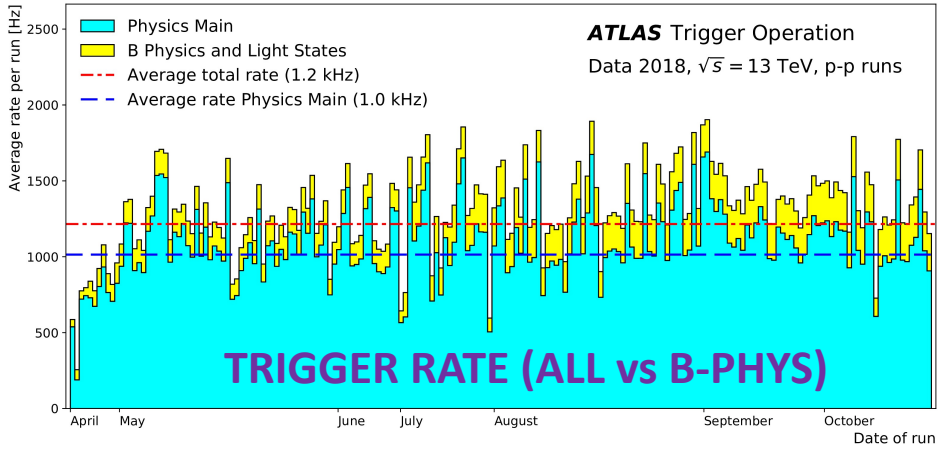


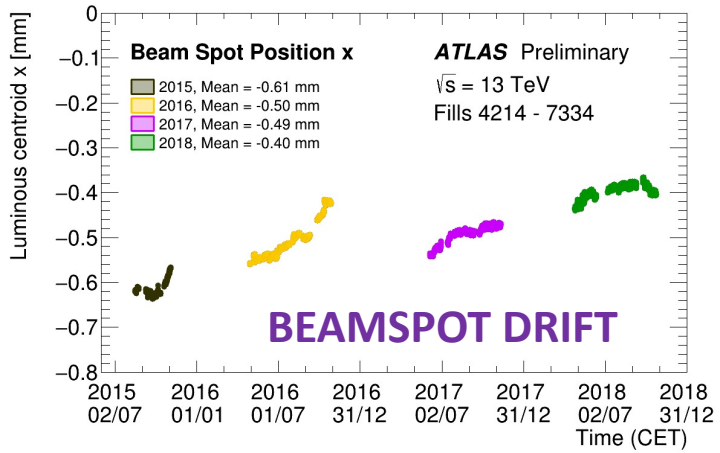
Data filtering and selection at hadron colliders

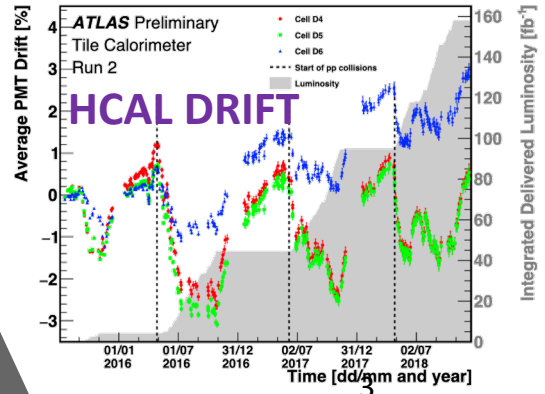
- Data reduction and selection levels ~10^{-5} required due to bandwidth constraints
- Hard real-time constraints necessitate fixed latency algorithms
- Data preparation of numerous data sources from front-end instrumentation
- Complex algorithms deliver variety of trigger and physics objects for accept vs. reject
- Huge selection menus ultimately determine data recorded vs. discarded

Problems with *status quo*

- Global optimization
- Changing conditions
- Known and unknown biases
- Data “waste”







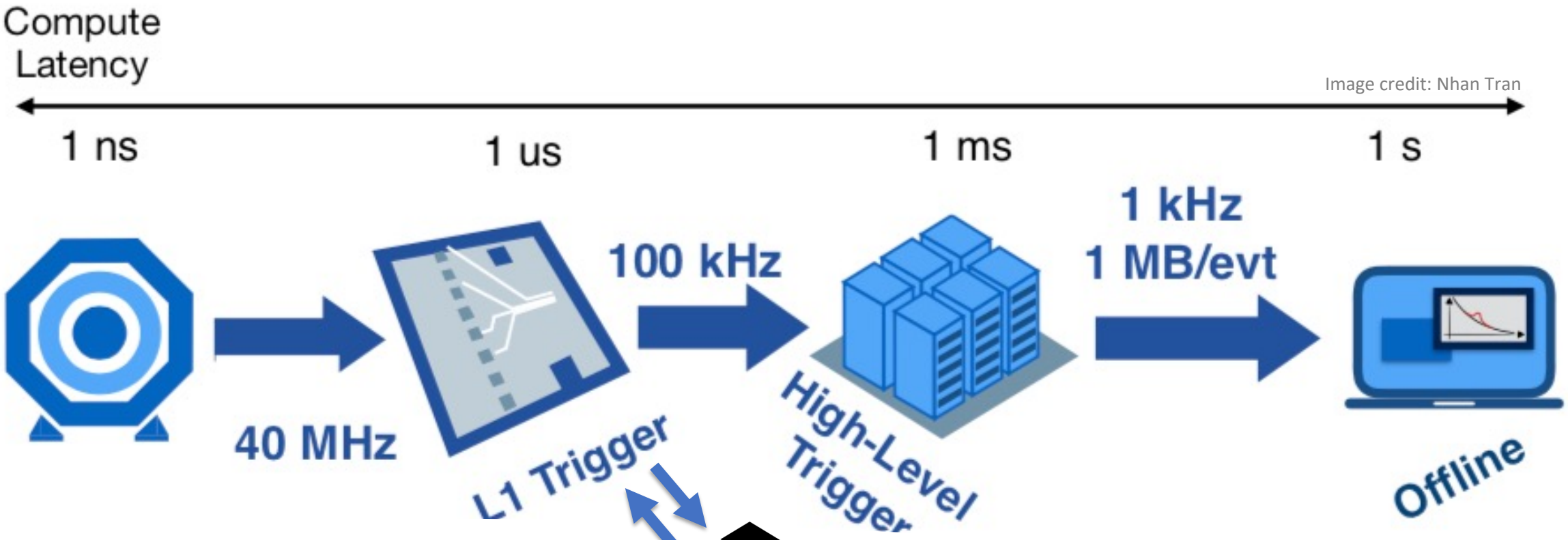
In 2018 alone:

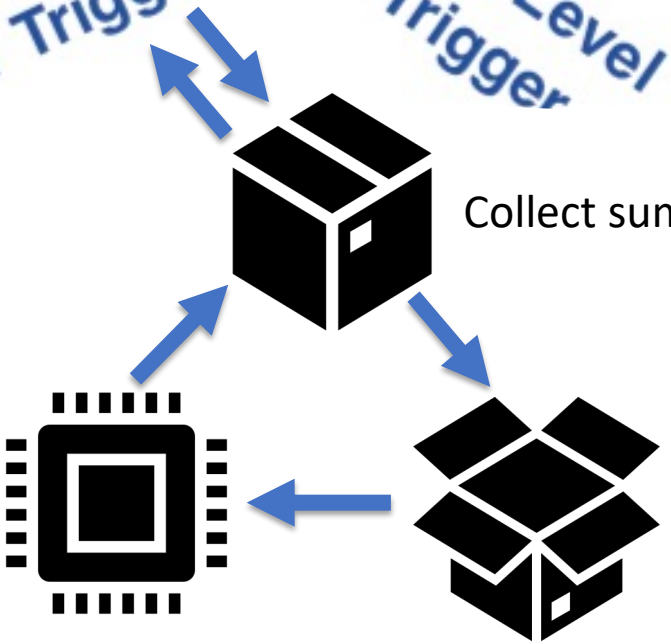
All events: 142 BILLION events (∼11 PB)

Triggered for physics: 6.4 BILLION events (∼6 PB)

Number of Triggers: ∼1200 triggers

What if the data processing and reduction pipeline could continuously learn to determine what data to save on its own??


Compute Latency
Image credit: Nhan Tran
1 ns
1 us
1 ms
1 s
40 MHz
L1 Trigger
100 kHz
High-Level Trigger
1 kHz
1 MB/evt
Offline


Collect summaries of data input to system
Compute updates and optimization of model
Continuous learning based on accept/reject

So, what would it take to build this?

Interpretable

Cost-aware

Traceable

What would these mean?

Interpretable

What are the most important information used in the selection?

What is the impact of those features?

How are those features varying in time?

Cost-aware

Are some features more costly than others?

What is the overlap and “value” of those features?

Traceable

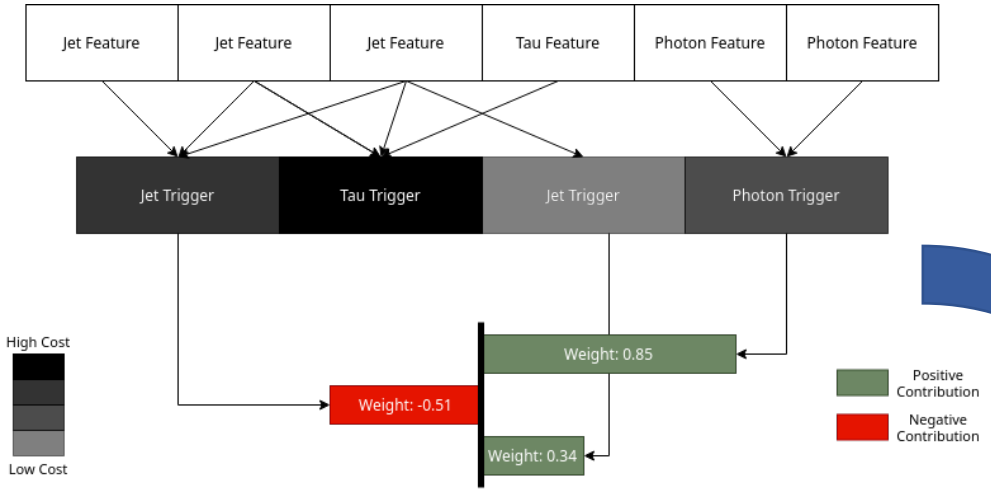
What updates are being proposed?

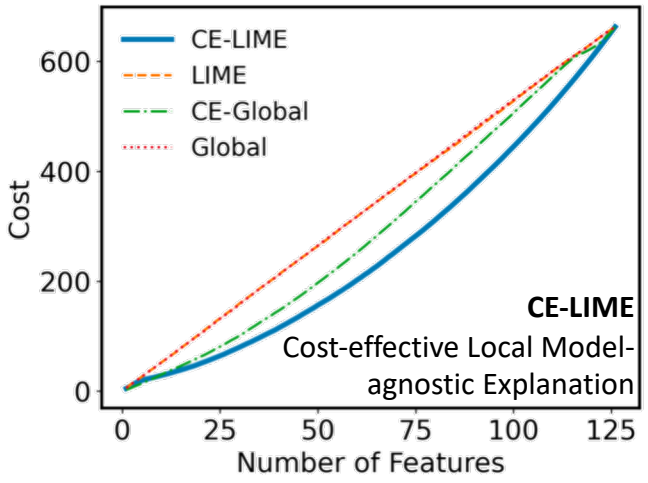
Can we track the proposed / implemented updates?

Can the effects of these updates be reliably simulated?

Progress on interpretation and cost-awareness

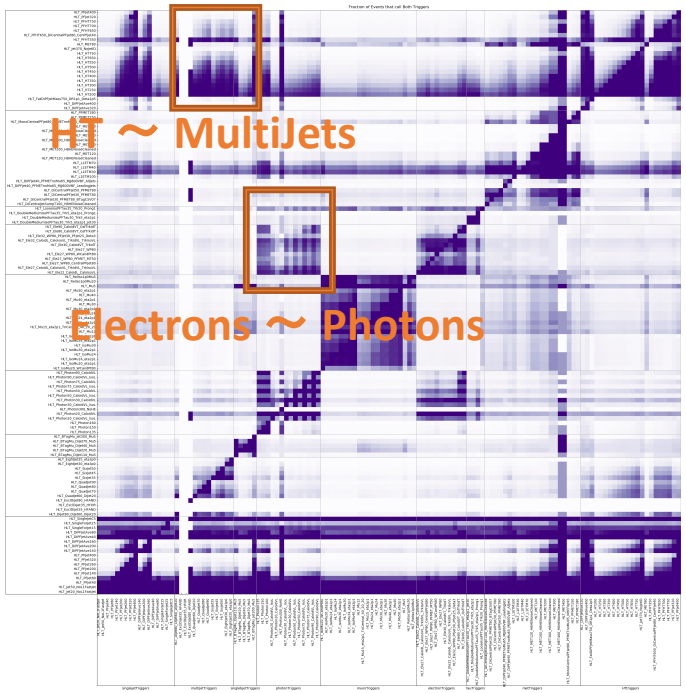
[NeurIPS 2020 WS] Self-Driving Trigger Paper



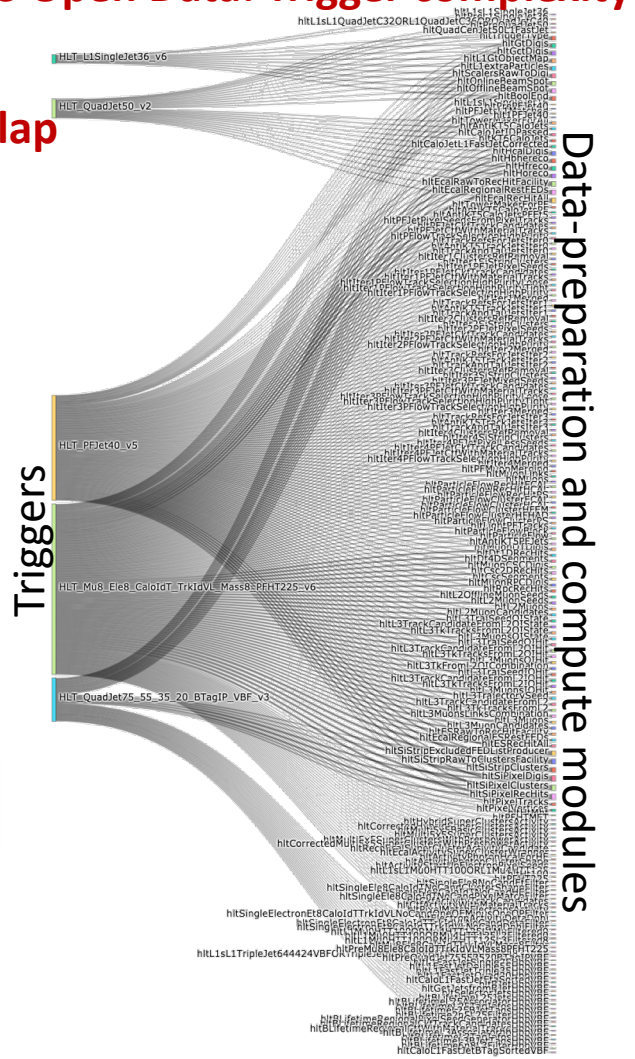


(d) Cost vs # Used Features, CMS Open Data

CMS Open Data: Trigger item overlap



CMS Open Data: Trigger complexity



Summary and conclusions

Our field is envisioning projects that span another 50 years, and so it is necessary that we allow ourselves to ask big questions!

The concept of an autonomous data filtering and processing system for high-throughput physics facilities is well-aligned with physics goals

We must ask what such a system could and should do for it to be useful, let alone feasible

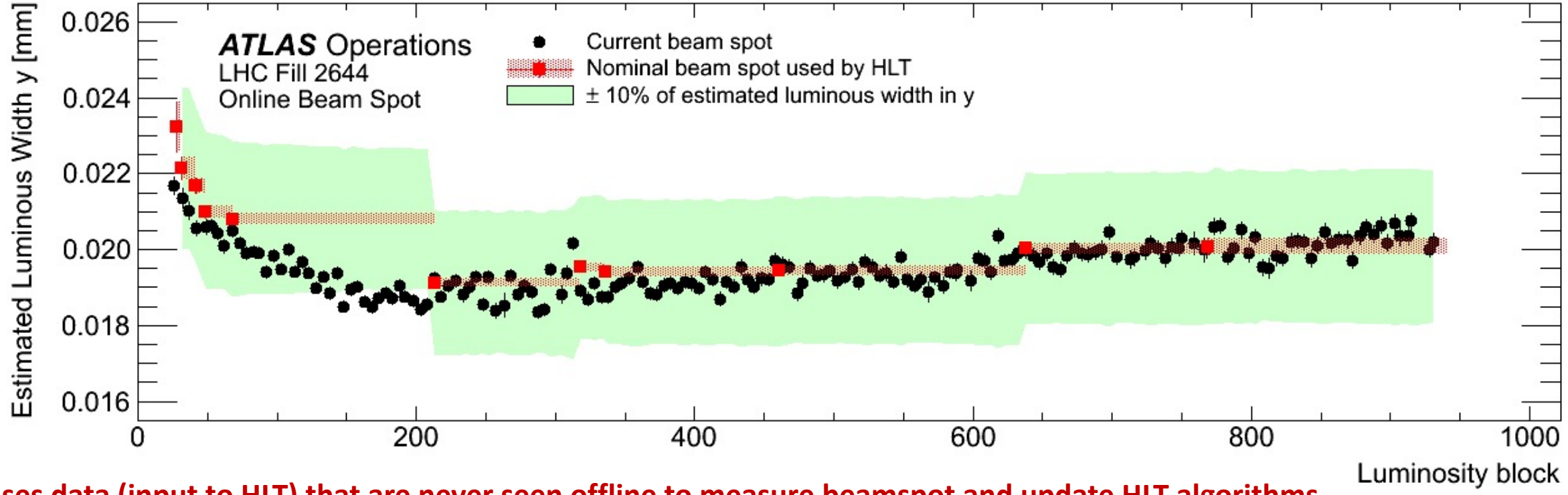
We have demonstrated some simple principles regarding interpretation of models and cost effectiveness using toy and open data

Expect results soon demonstrating proof-of-principle using realistic dataset and cost models and functions

Additional Information

ATLAS online beamspot measurement

(HLT = "High Level Trigger", algorithms run on commodity CPU + accelerators)



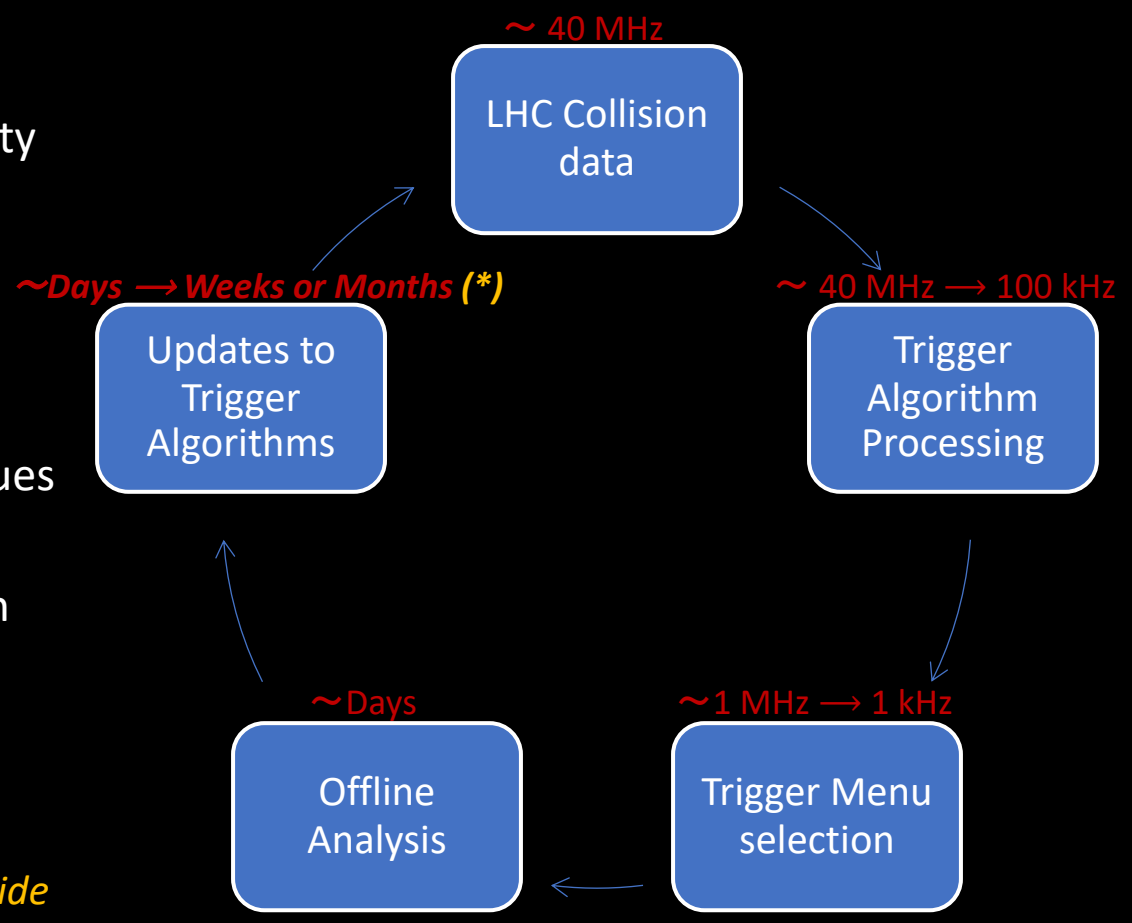
Uses data (input to HLT) that are never seen offline to measure beamspot and update HLT algorithms

Very rough sketch of current approaches

Very good reasons for the stability (slowness) of updates:

- Well-understood trigger efficiencies and behaviors
- Modeling in simulation
- Logistics and bookkeeping issues in menu design and analysis
- Known and unknown biases in selection algorithms

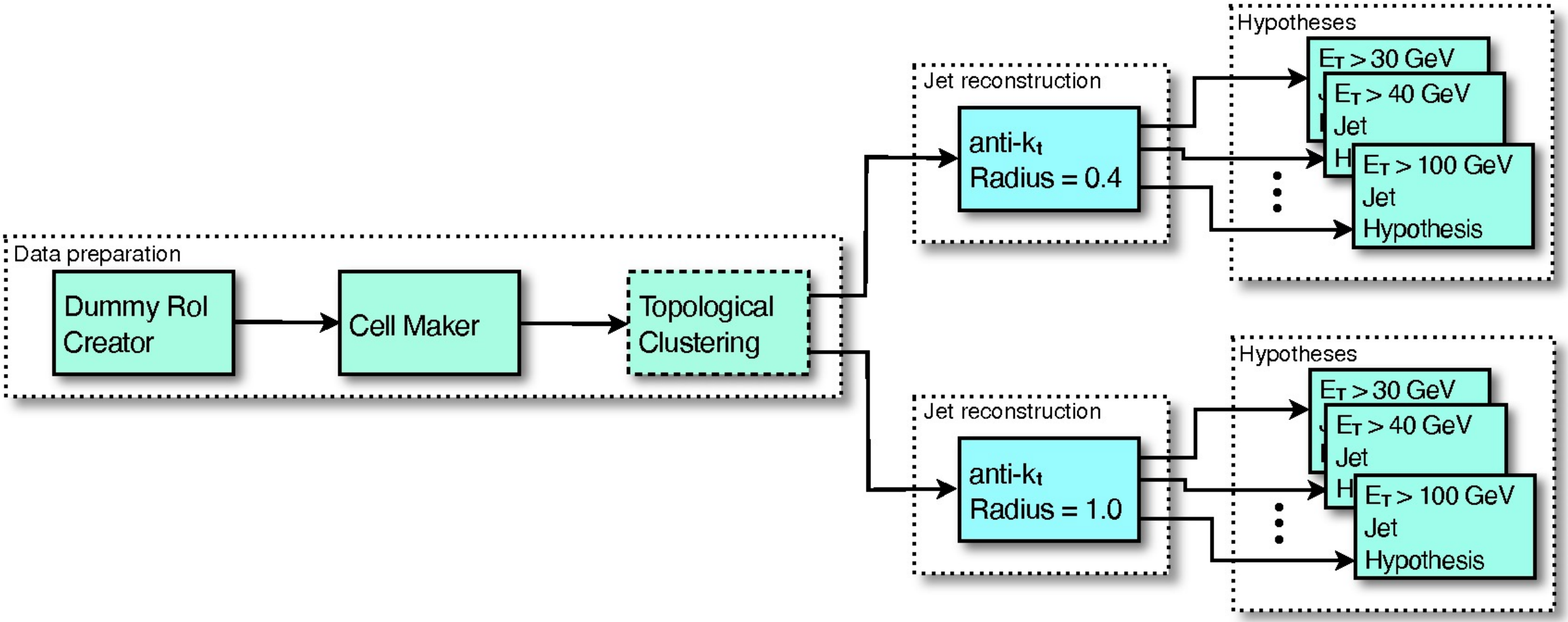
(*) Except for "prescales", see next slide

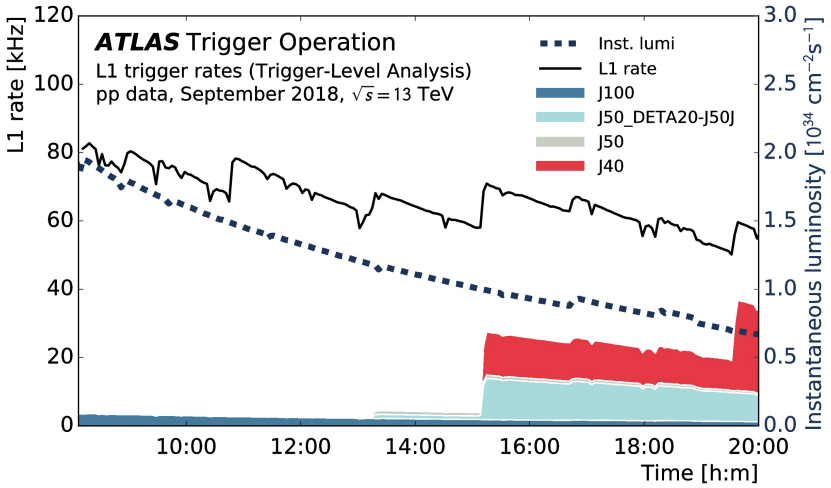


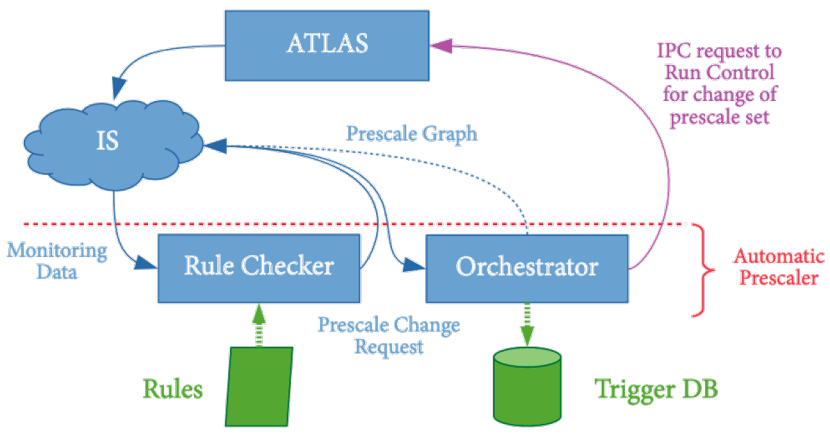
Before we would even consider allowing for continuous updates *(or intermittent but autonomous)* we would insist on knowing:

- What has been learned such that an update is merited?
  - ***Interpretability***
- What are the impacts of those updates?
  - ***Cost and benefit***

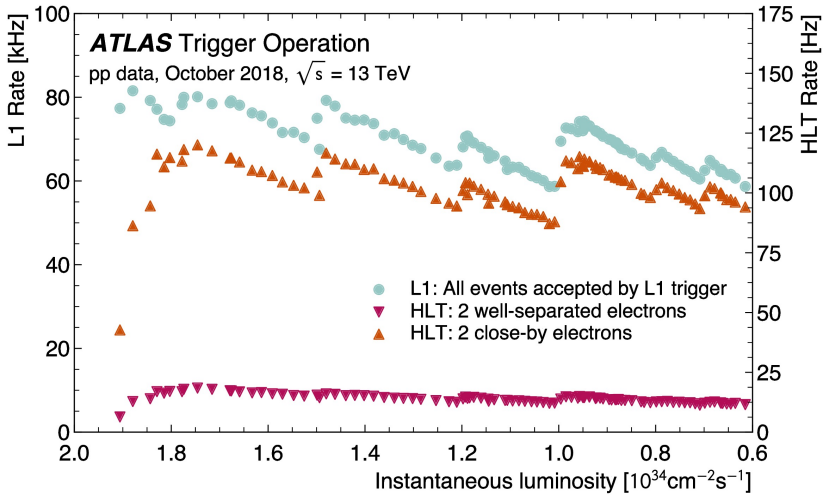
Typical (HLT) trigger algorithm workflow





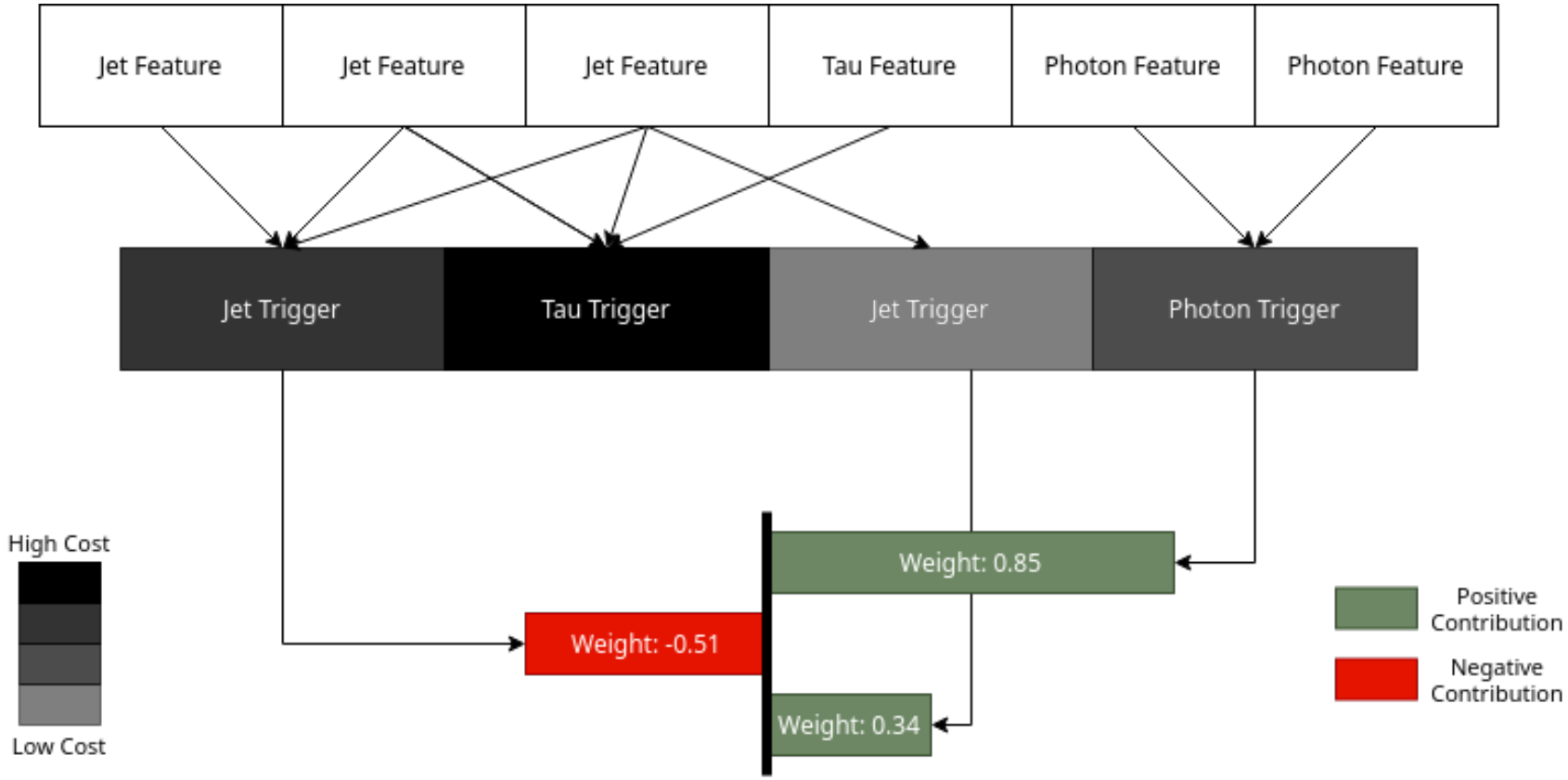


G. Galster,Relational databases for conditions data and event selection in ATLAS, (2019), PhD Thesis



Automatic prescaling in ATLAS

"Prescale" = Rate limiter applied to a trigger in pseudo-random fashion $\longrightarrow \text{Trigger Rate} = \frac{1}{\text{Prescale}}\sigma\mathcal{L}$



Cost-effective "explanation" of an event

For this single event, the Tau Trigger has the highest cost and thus the weight associated with the Tau Trigger was driven to 0.

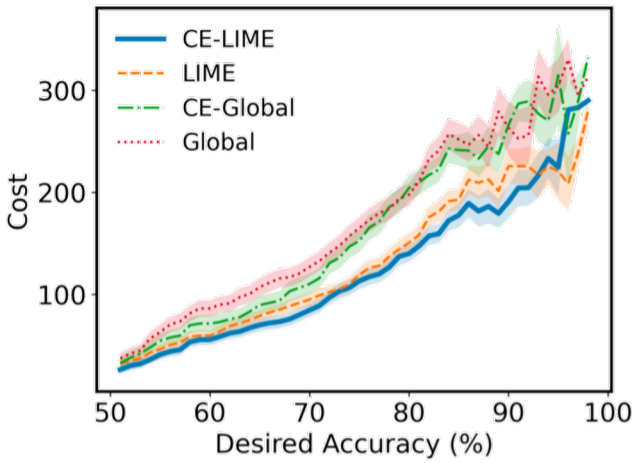
The remaining weights result in the final cost-effective explanation of the event, with the weights with the highest ***absolute value*** being the most important.

Demonstration of trigger (cost) optimization

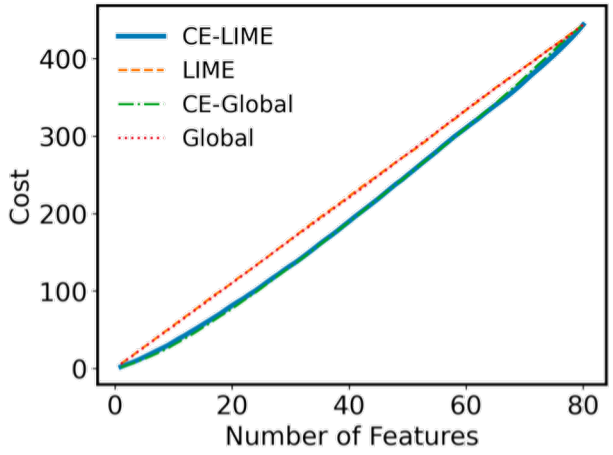
[NeurIPS 2020 WS] Self-Driving Trigger Paper

We modified the interpretability framework on the previous slide to account for ***cost***.

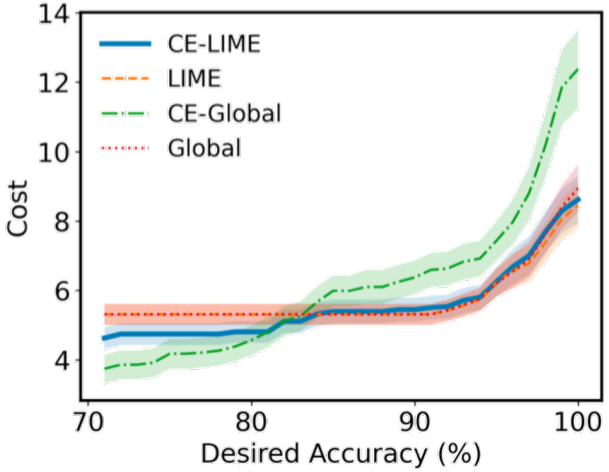
- **Toy dataset**
  - Randomly generate trigger items ("features") and associated resource costs ("cost")
  - Minimize the total cost while maintaining the physics result (accept or reject!)
- **CMS Open Data**
  - Use triggers and events from CMS open data
  - Still randomly assign costs (proof-of-principle)



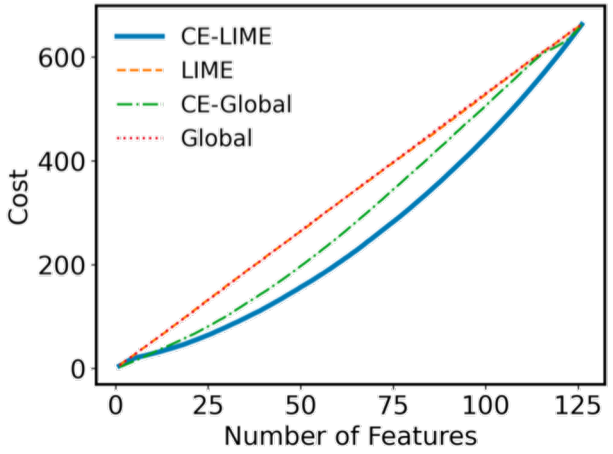
(a) Cost vs Performance, Toy Dataset



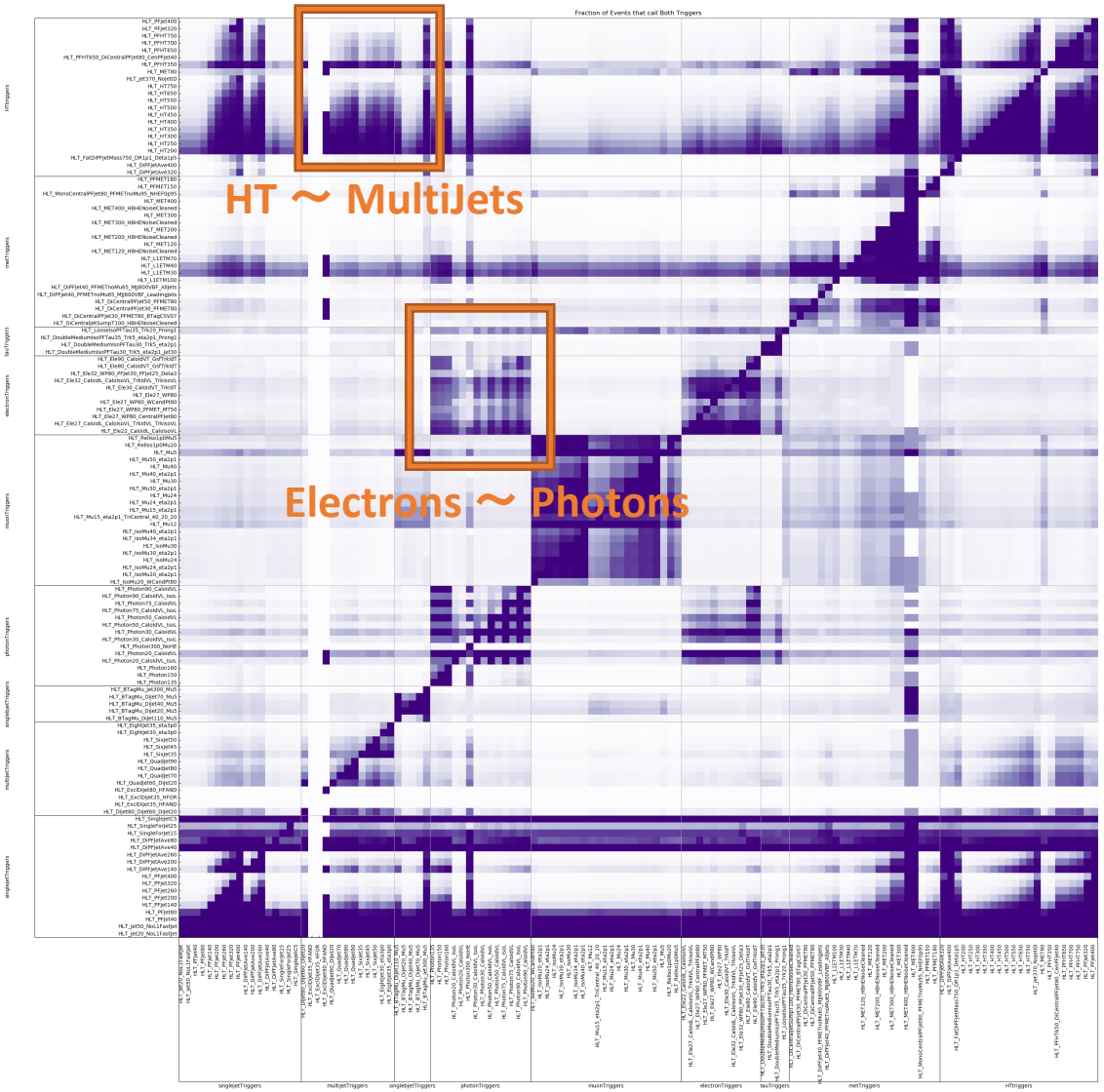
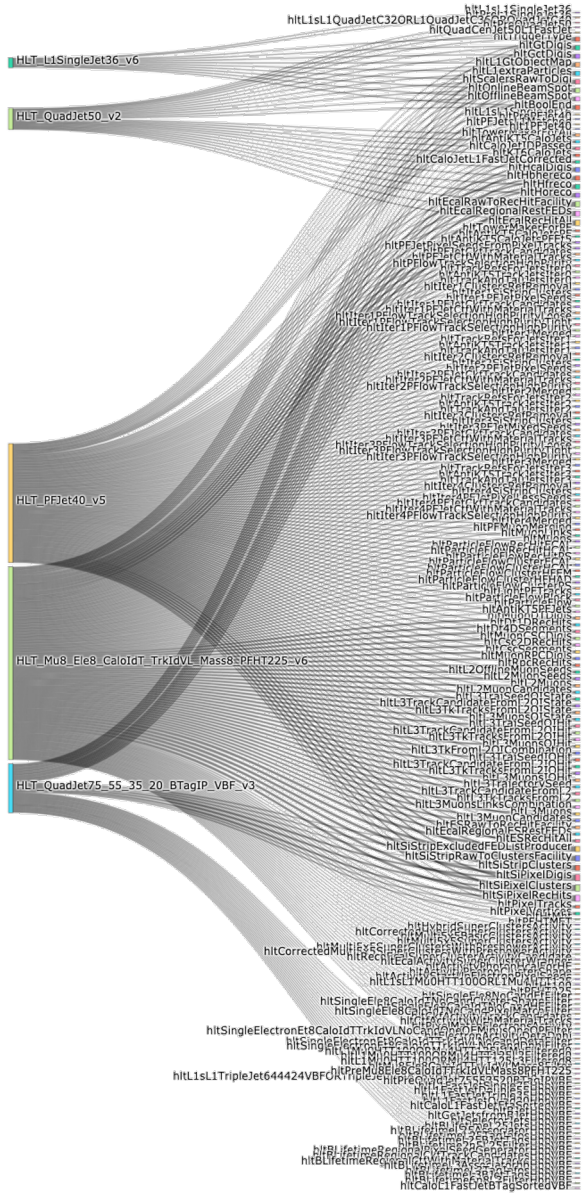
(b) Cost vs # Used Features, Toy Dataset



(c) Cost vs Performance, CMS Open Data



(d) Cost vs # Used Features, CMS Open Data

Modeling trigger “cost” using CMS open data

[NeurIPS 2020 WS] Self-Driving Trigger Paper