Towards an Interpretable Data-driven Trigger System for High-Throughput Physics Facilities

Or

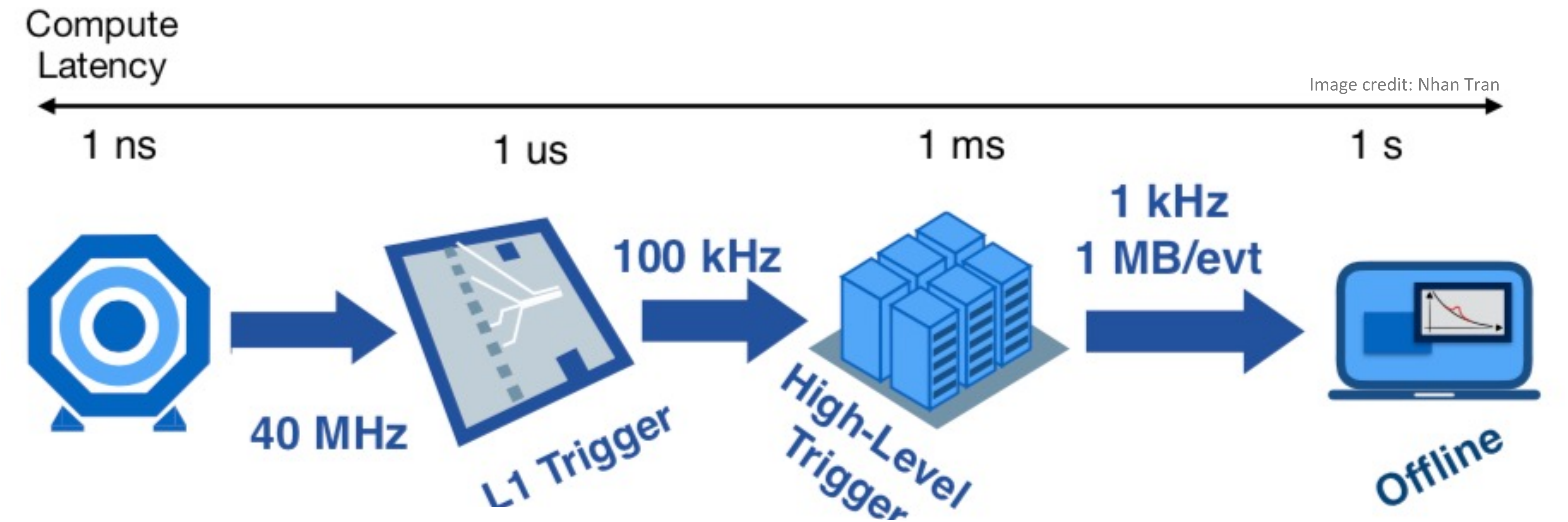
Progress towards a self-driving data filtering and processing system

David W. Miller, University of Chicago

along with collaborators:

Yuxin Chen, Kristin Dona, Chinmaya Mahesh, Cecilia Tosciri

(also: Andrew Chien & Nhan Tran)

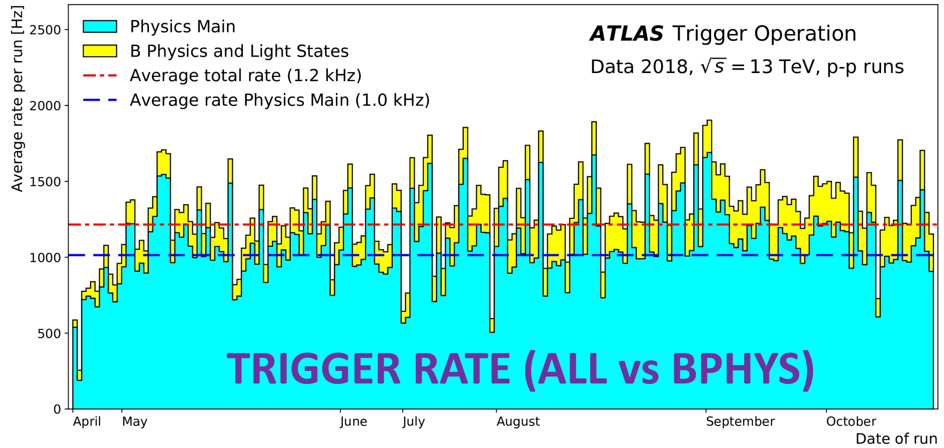


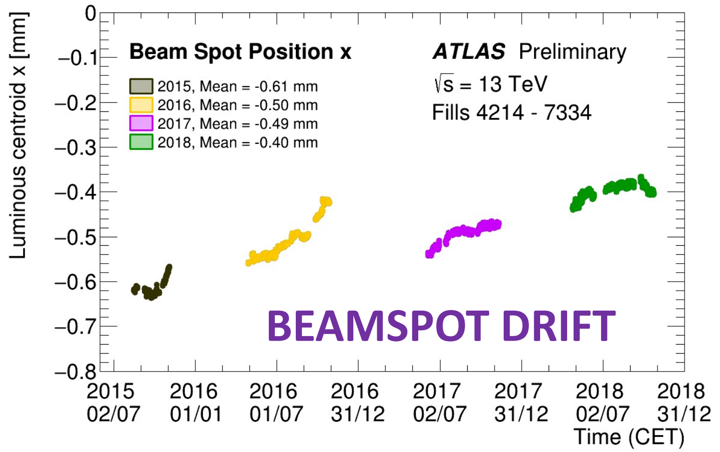
Data filtering and selection at hadron colliders

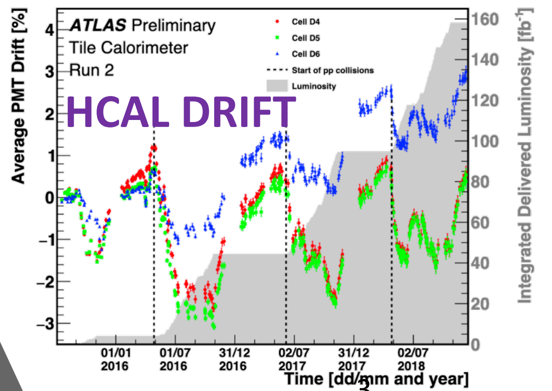
- Data reduction levels of 10^{-5} required due to bandwidth constraints
- Hard real-time constraints necessitate fixed latency algorithms
- Data preparation of numerous data sources from front-end instrumentation
- Complex algorithms deliver variety of trigger and physics objects for accept vs. reject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

The problem(s)

- Triggers not necessarily globally optimized for both physics and resource usage
- Accelerator conditions vary with time
- Detector conditions vary with time
- Trigger menus have both known and unknown biases
- Most of the data is never used, despite being processed

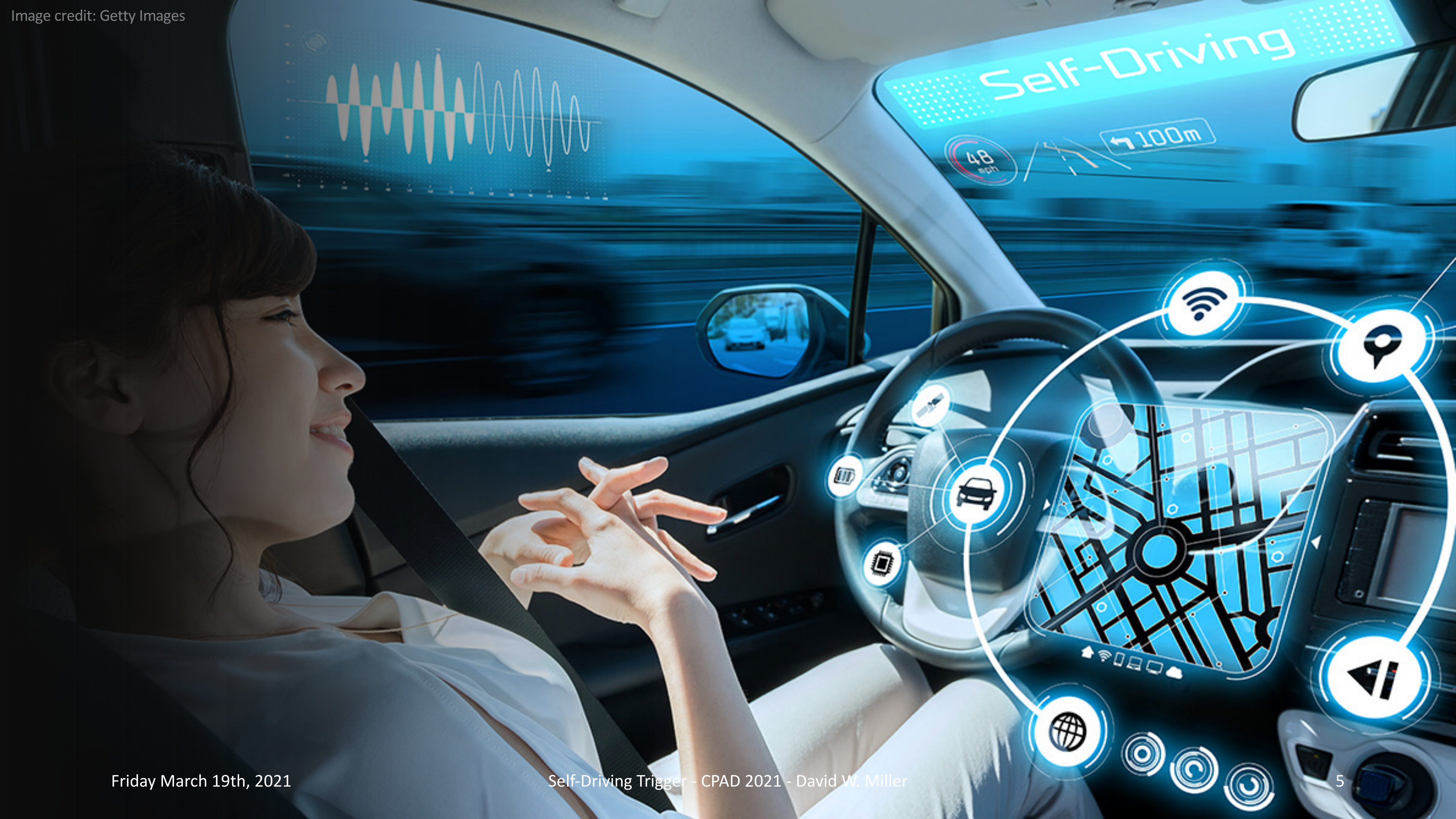




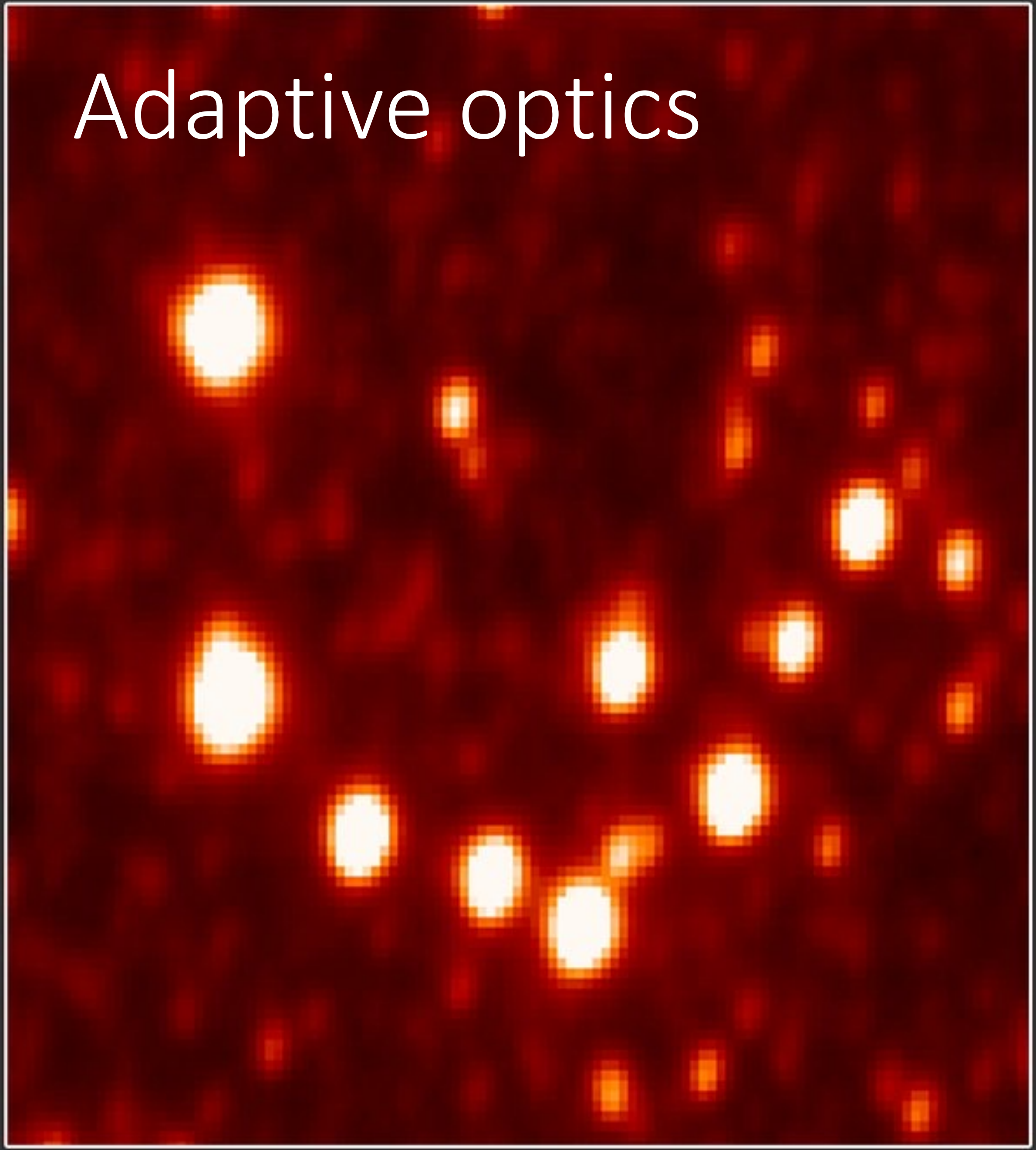


What if the data processing and reduction pipeline could continuously learn to determine what data to save on its own??

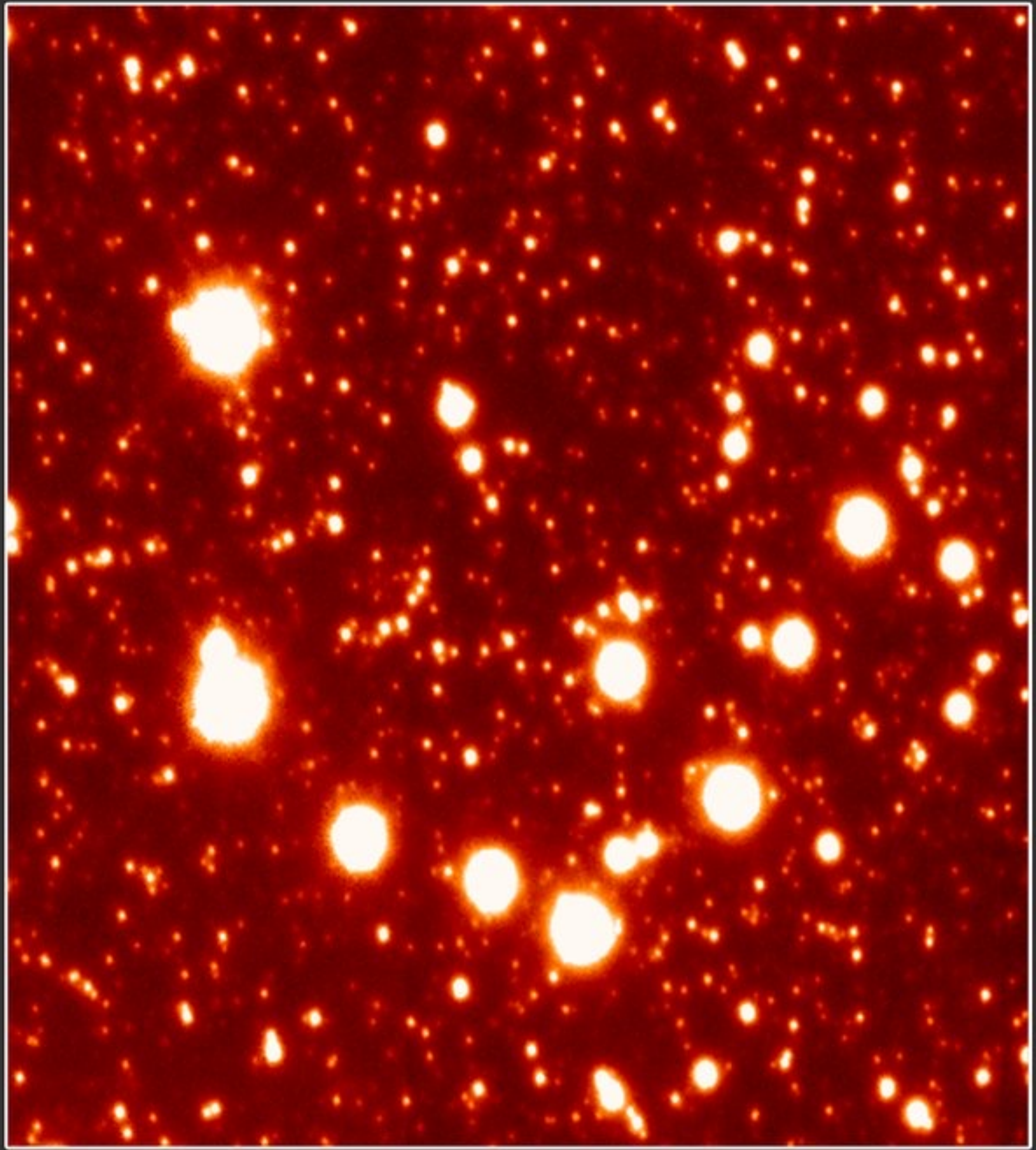
Image credit: Getty Images



Adaptive optics



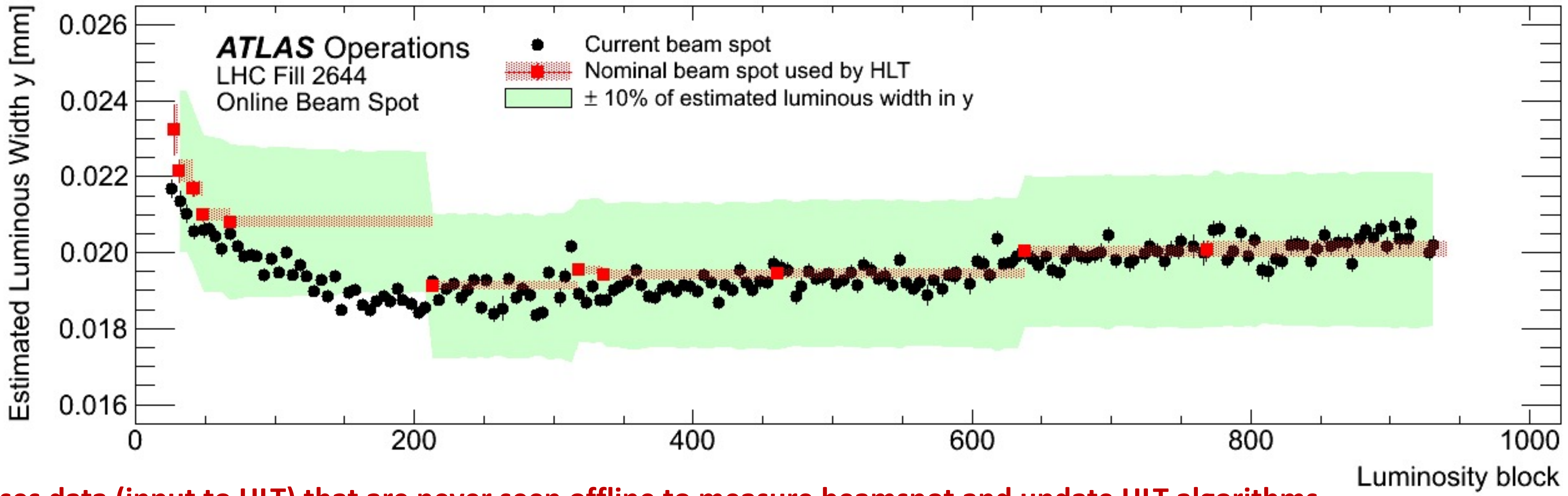
Without Adaptive Optics



With Adaptive Optics with MAD

ATLAS online beamspot measurement

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

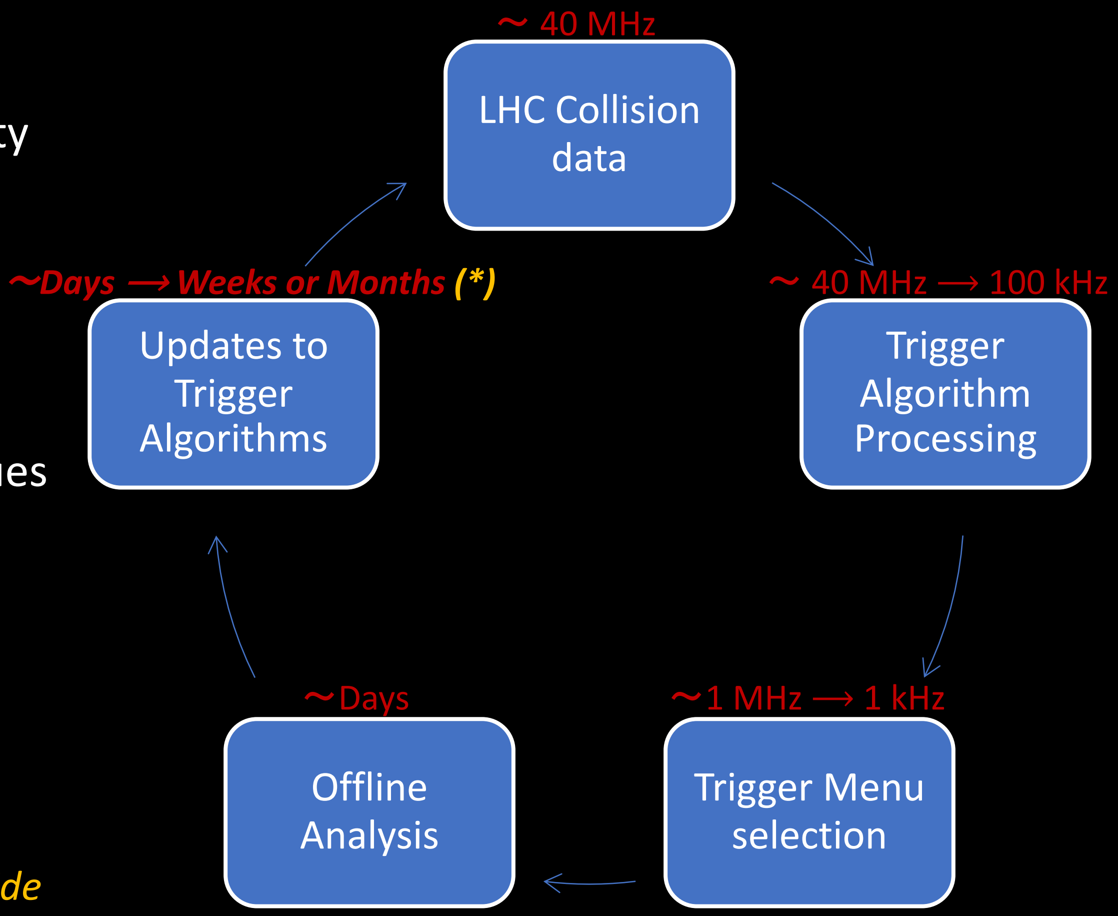


Uses data (input to HLT) that are never seen offline to measure beamspot and update HLT algorithms

Very rough sketch of current approaches

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

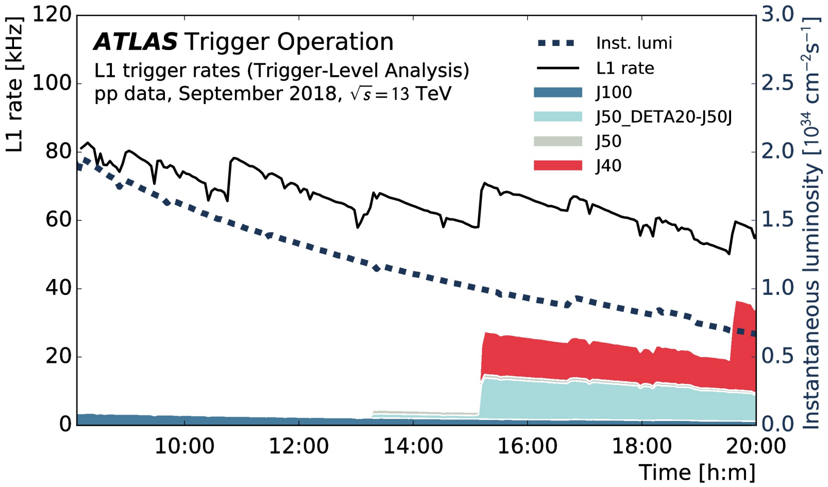
- Well-understood trigger efficiencies and behaviors
- Modeling in simulation
- Logistics and bookkeeping issues in menu design and analysis
- Known and unknown biases in selection algorithms

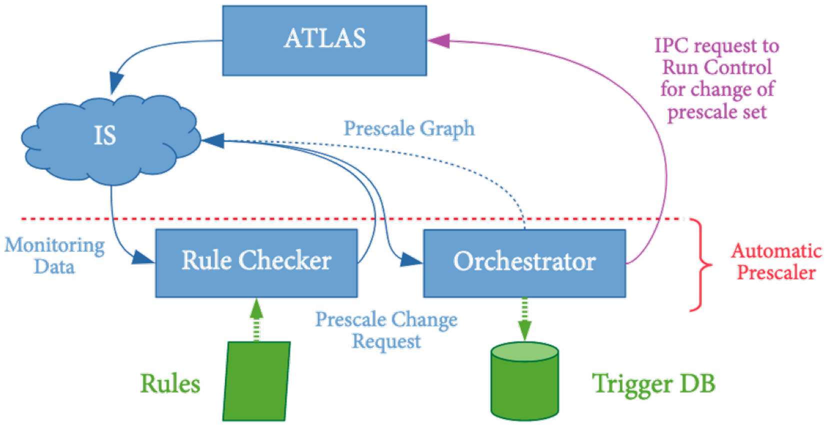


Before we would even consider allowing for continuous updates *(or intermittent but autonomous)* we would insist on knowing:

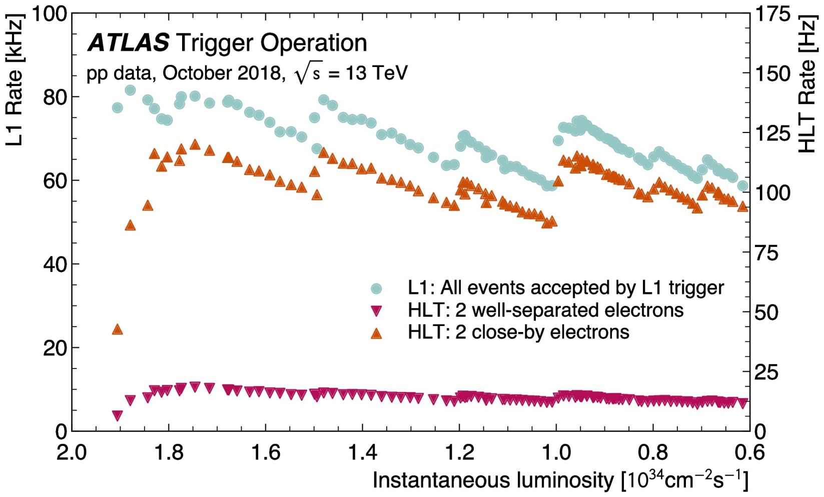
- What has been learned such that an update is merited?
  - ***Interpretability***
- What are the impacts of those updates?
  - ***Cost and benefit***

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





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 $\longrightarrow \mathrm{Trigger\,Rate} = \frac{1}{\mathrm{Prescale}}\sigma\mathcal{L}$

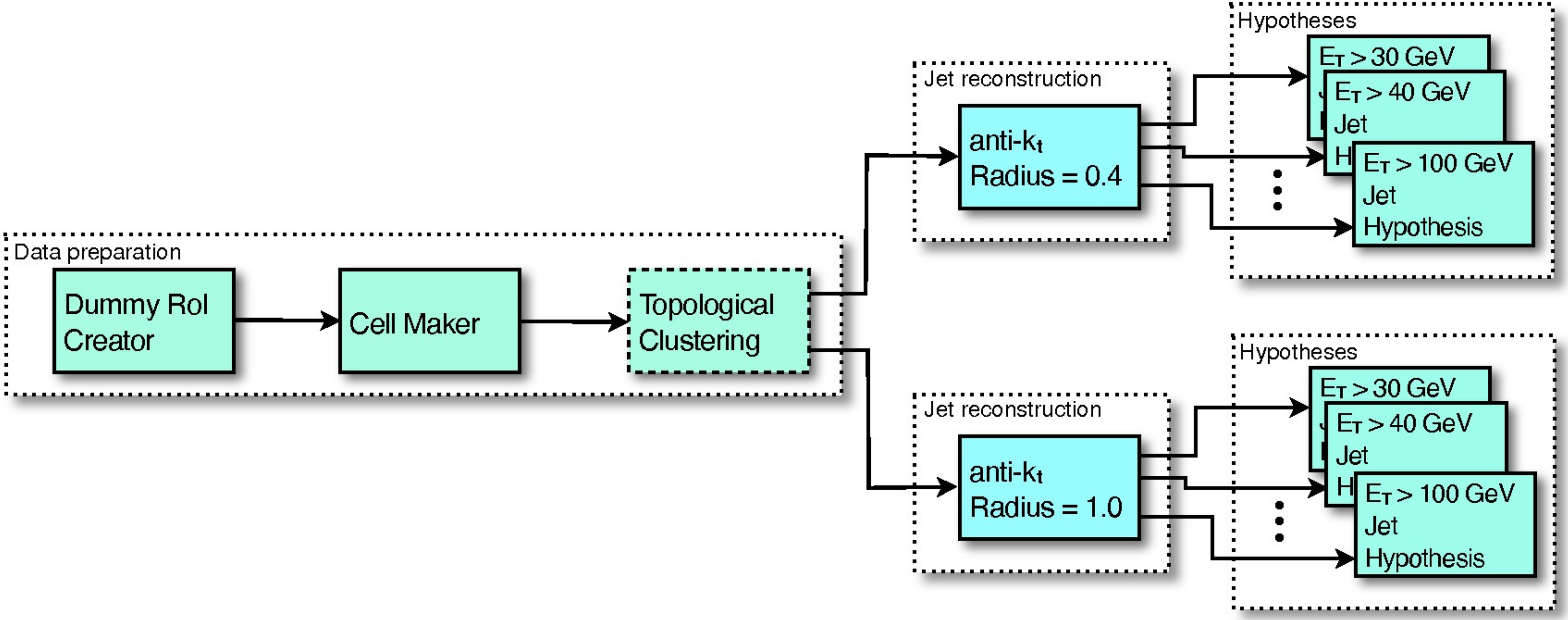
In 2018 alone:

All events: 142 BILLION events (～11 PB)

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

Number of Triggers: ～1200 triggers

Typical (HLT) trigger algorithm workflow



Envisioning a self-driving trigger system

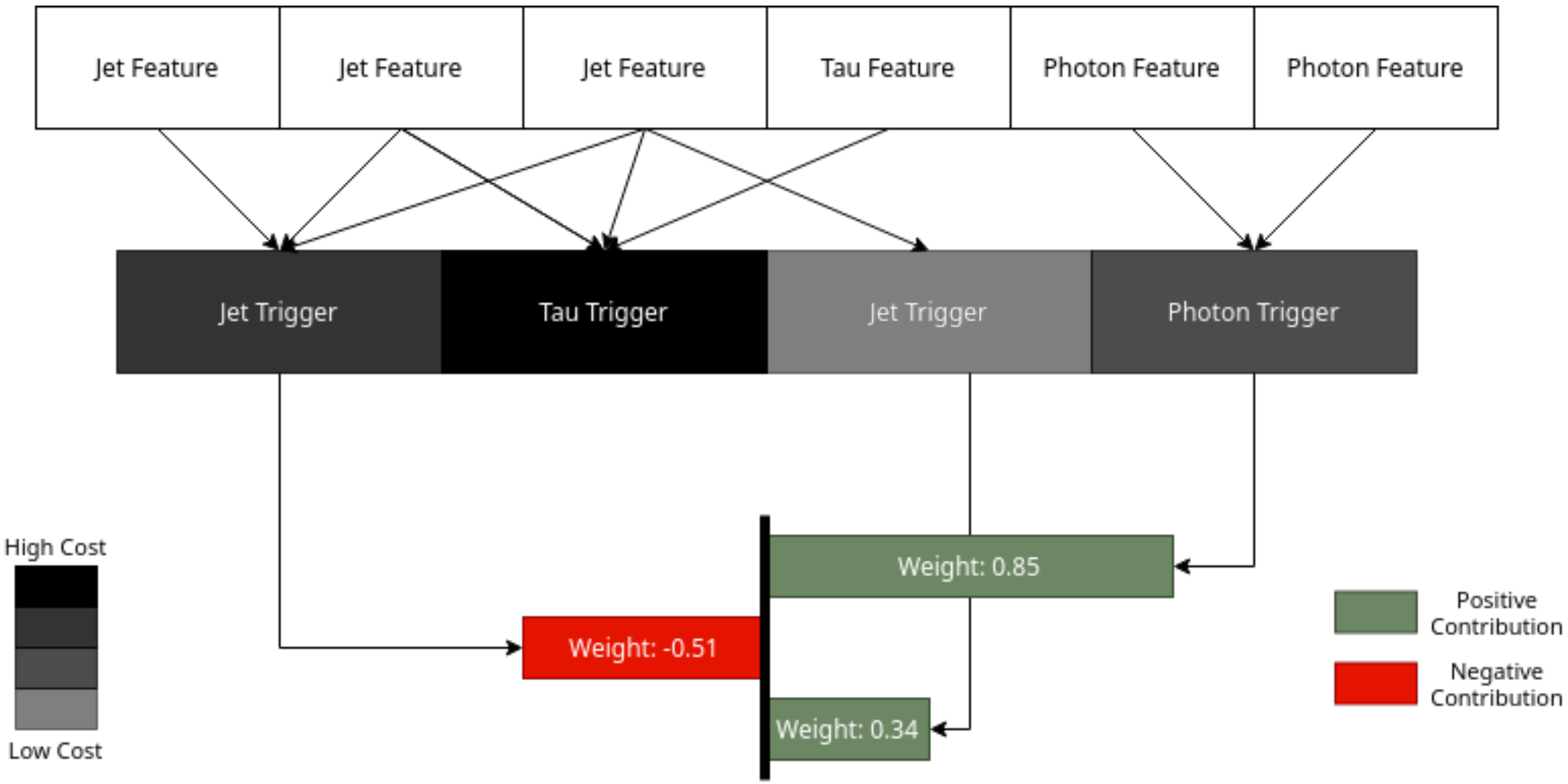
What has been learned such that an update is merited?

- Interpret the output of the algorithm
- “Why” was the event triggered?
- What trigger algorithm was “most important” to the trigger decision?

What are the impacts of those updates?

- Given a definition of the resource cost of a set of triggers, how can we optimize the algorithm execution and usage to minimize that resource usage?
- Cost might include bandwidth considerations, CPU time, data preparation, etc

[NeurIPS 2020 WS] Self-Driving Trigger Paper



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.

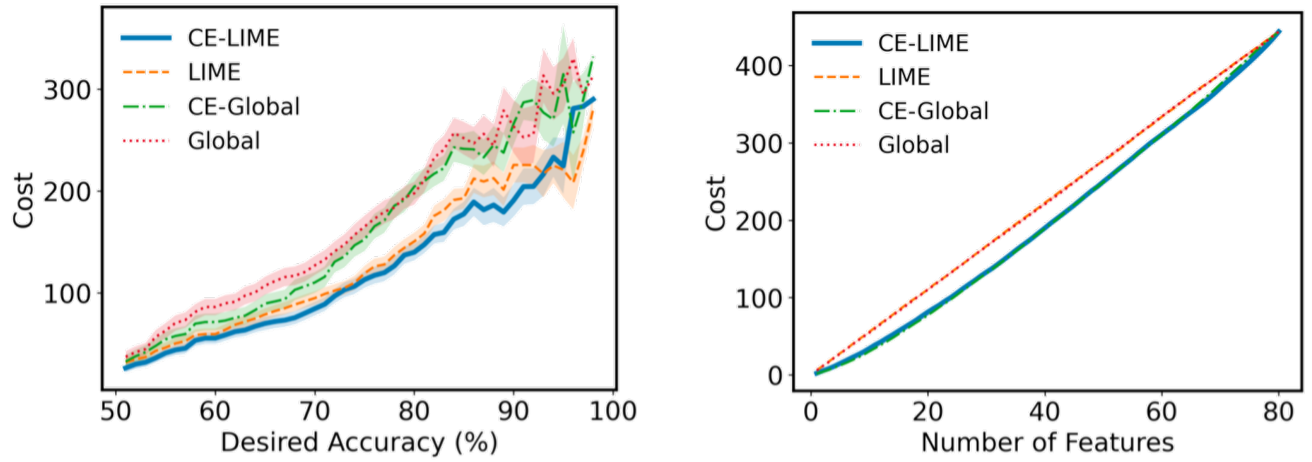
Demonstration of trigger (cost) optimization

[NeurIPS 2020 WS] Self-Driving Trigger Paper

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

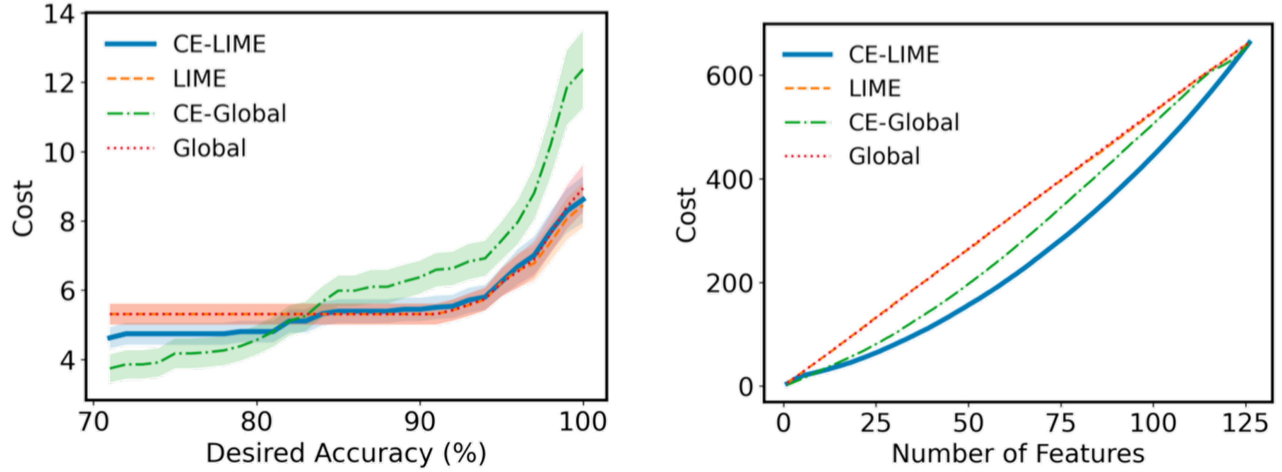
- **Toy dataset**
  - Randomly generate trigger items ("features") and associated resource costs ("cost")
  - Minimize the total cost while maintaining the physics result (accept or reject!)

- **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

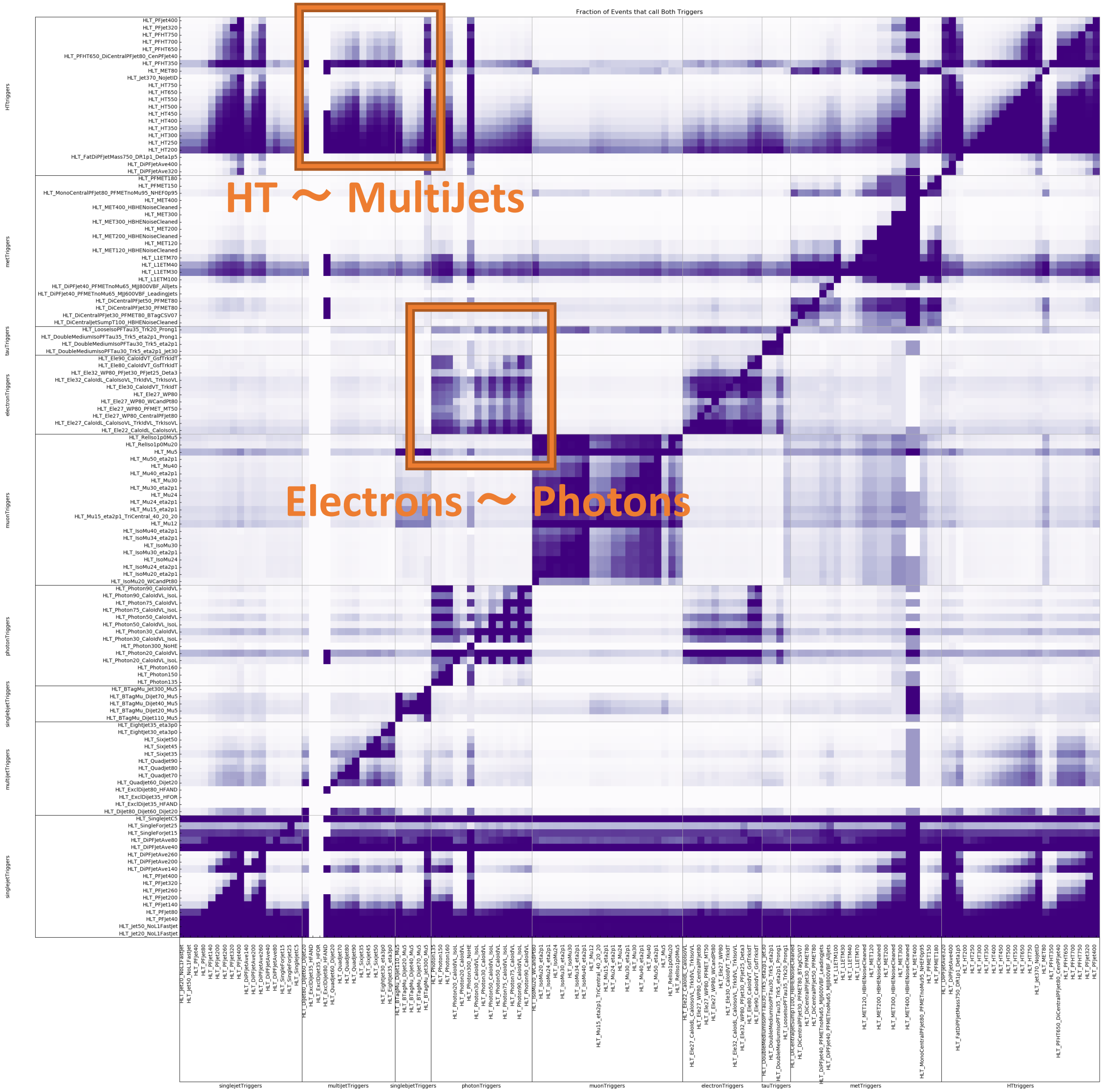


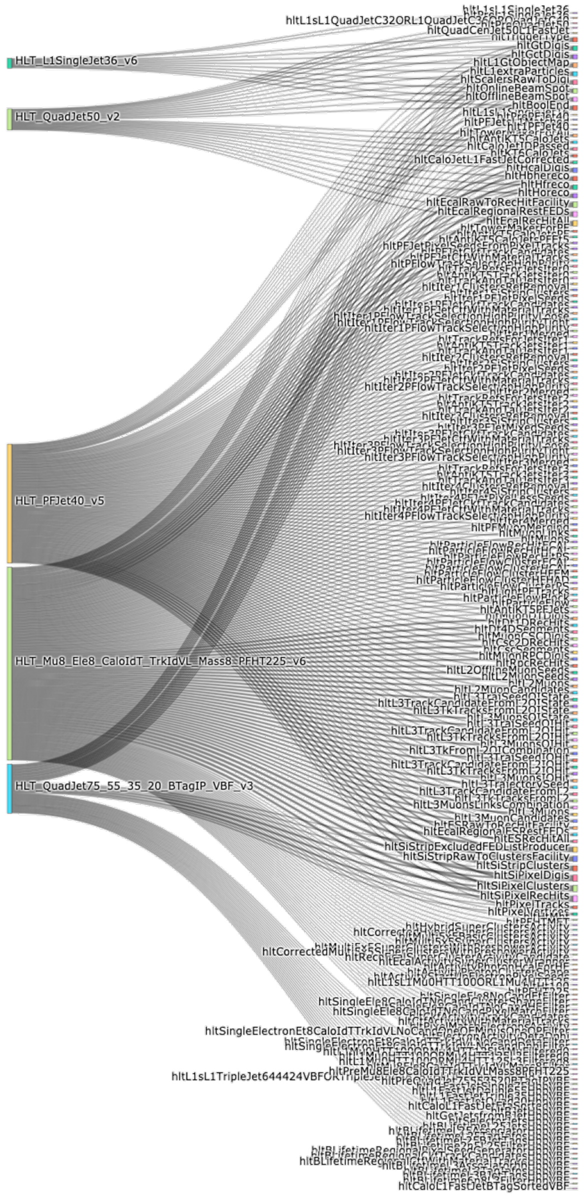
(c) Cost vs Performance, CMS Open Data

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

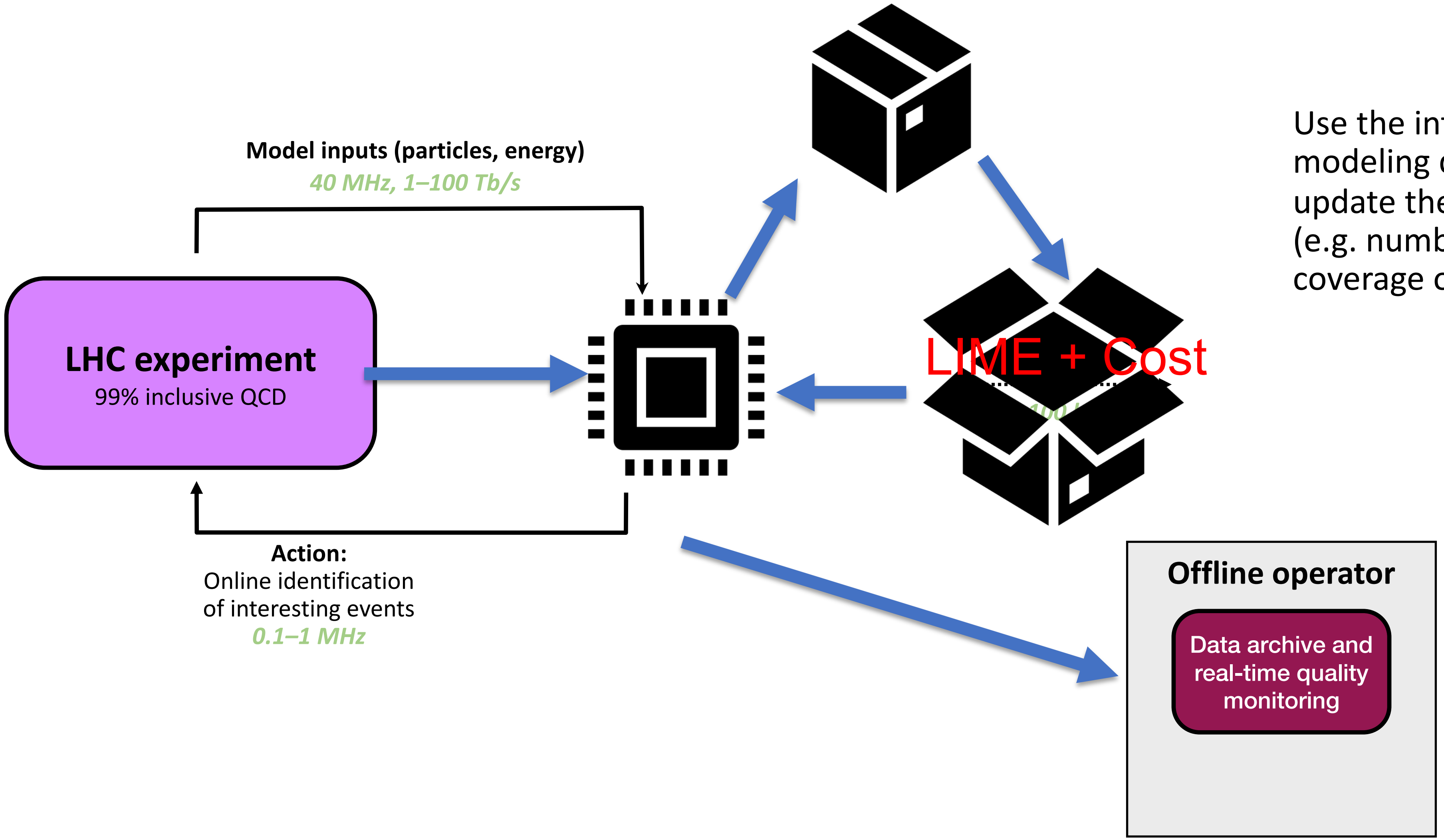
Modeling trigger “cost” using CMS open data

[NeurIPS 2020 WS] Self-Driving Trigger Paper





Future work: stream-based active learning



Use the interpretable and cost-effective modeling described in the previous slides to update the trigger selections and algorithms (e.g. number of jets) required to maintain coverage of key physics processes.

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