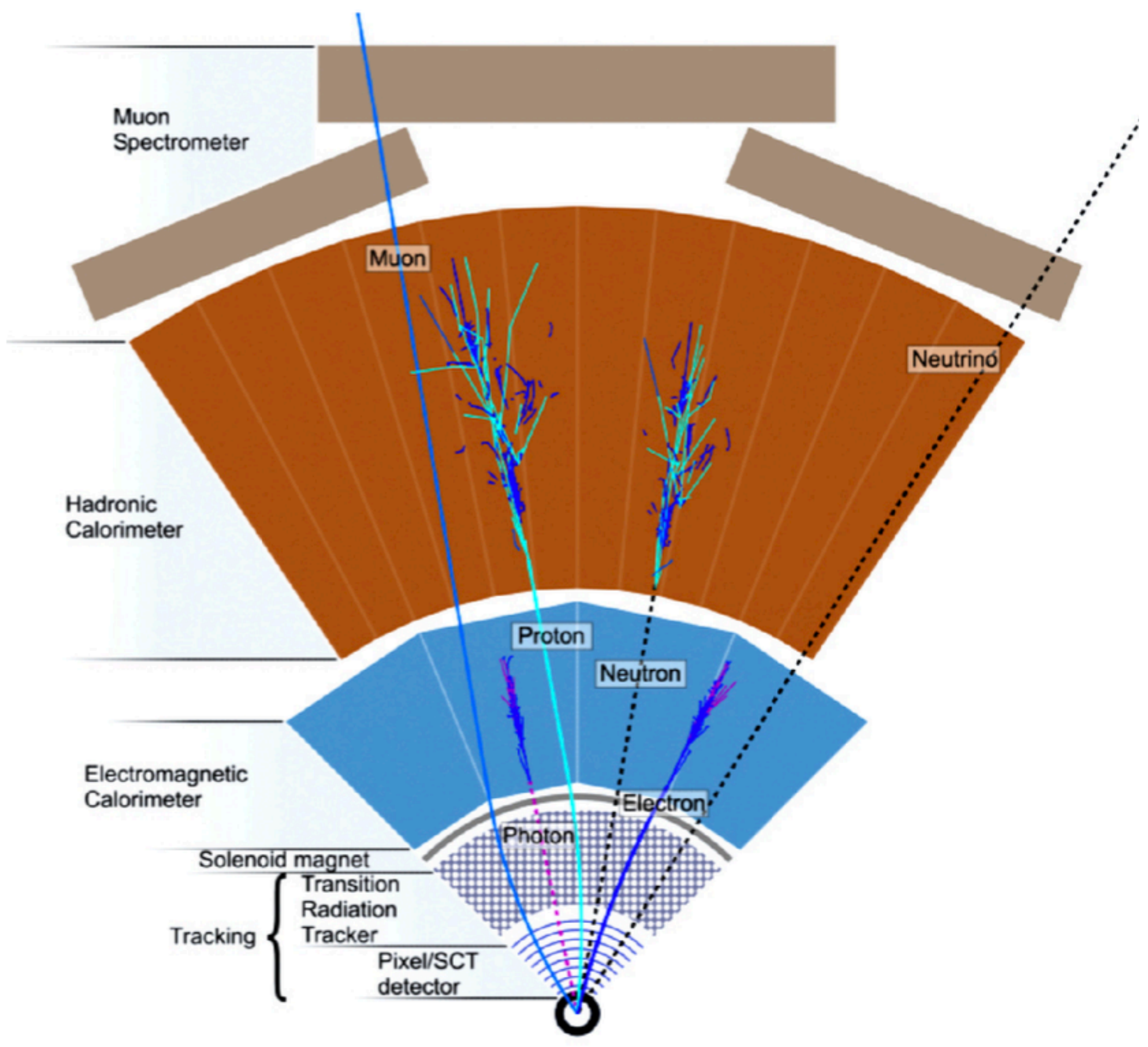
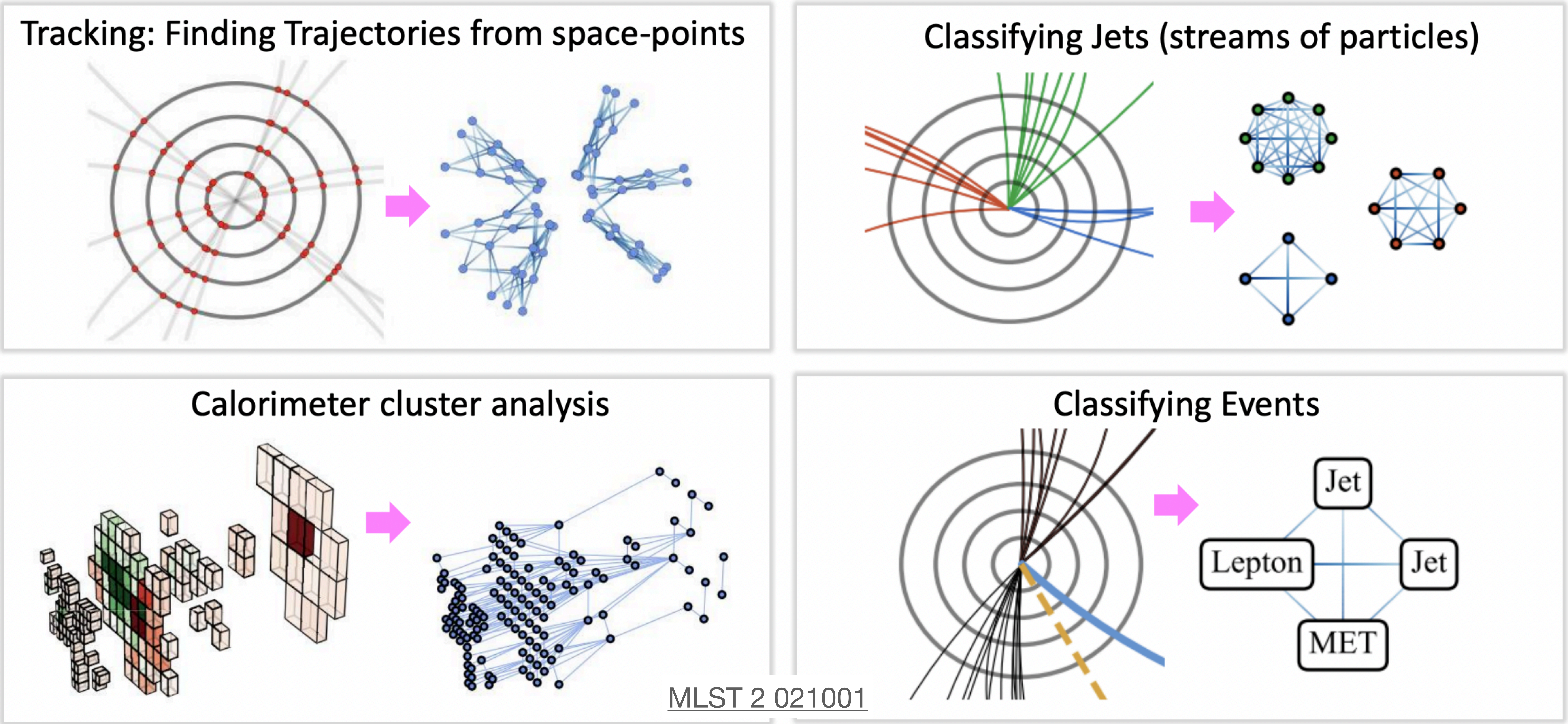
Overview of fast ML for detectors and control

Dylan Rankin [UPenn]
ML4FE Workshop
University of Hawaii
May 20th, 2025

Introduction

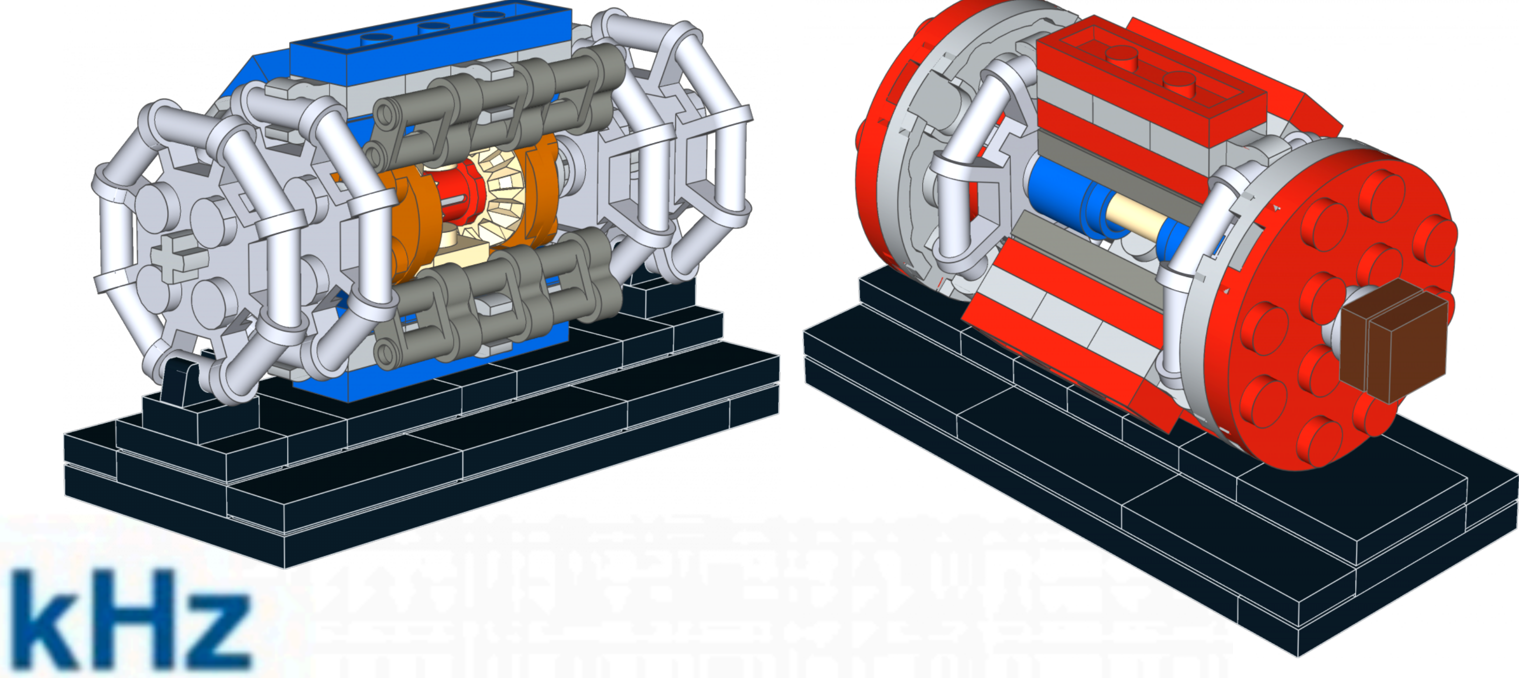
- ML is becoming more and more popular across science
- Better algorithms → improved sensitivity to new physics and measurements
- If we want to really make the most of these improvements, have to bring ML to our detectors (front ends, triggers, readout, data acquisition, …)

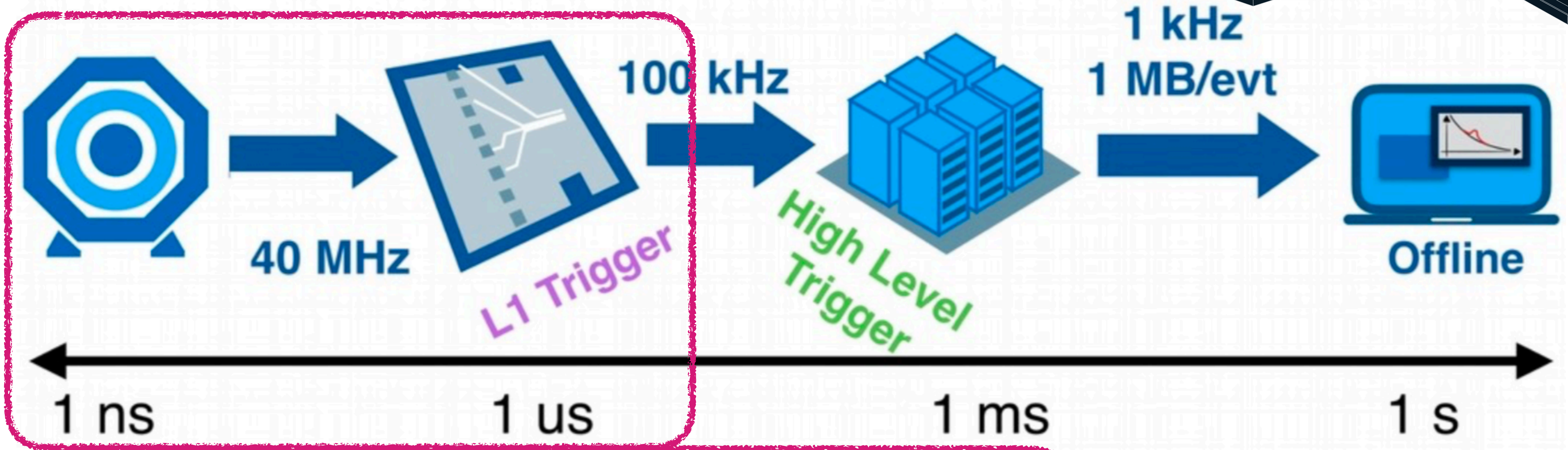




MLST 2 021001

LHC Data Processing / Readout

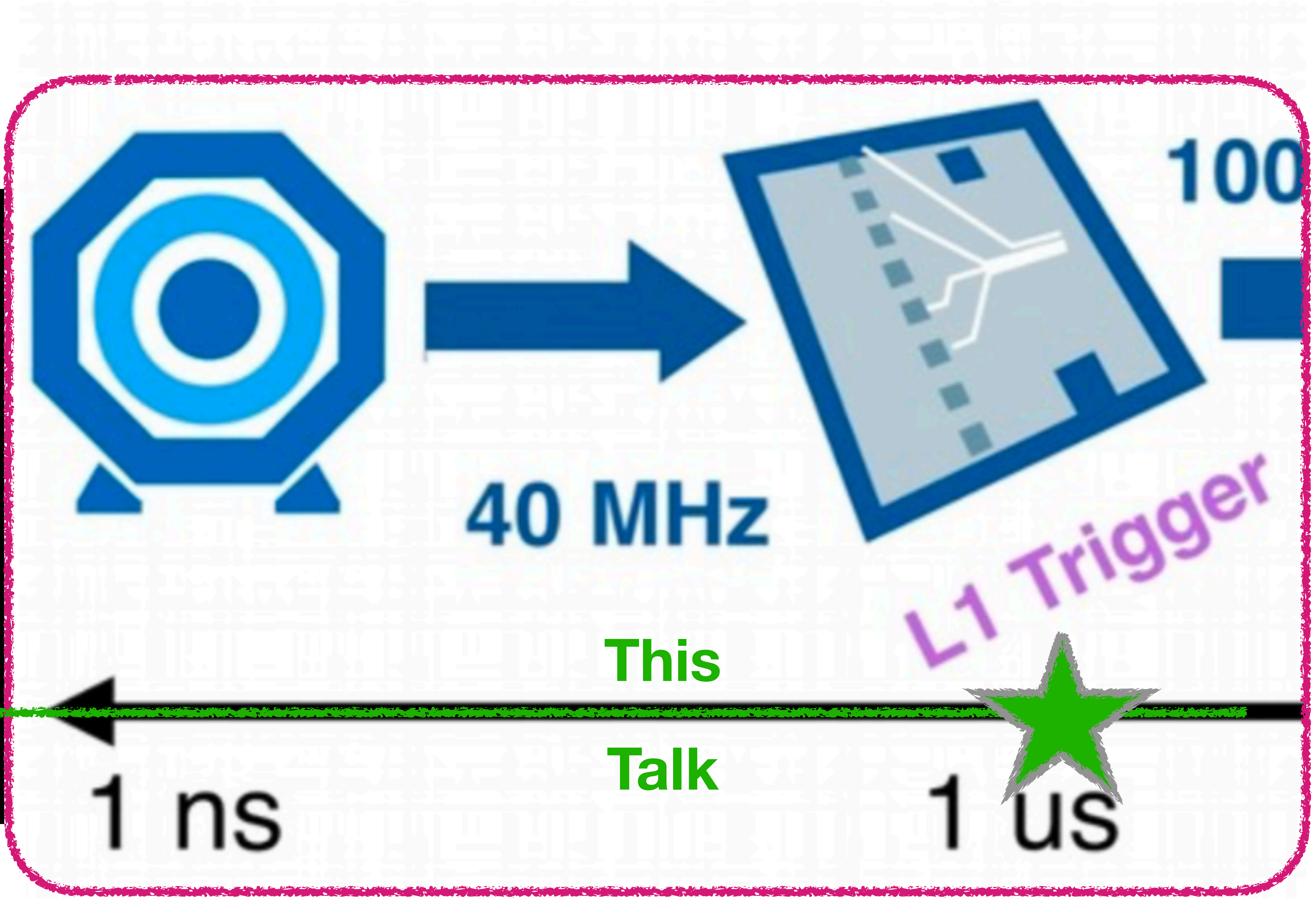




- **Level-1 Trigger** - O(μs) latency
- **High Level Trigger** - O(100 ms) latency
- **Offline** → 1 s latencies

If we don't identify interesting events in trigger we lose them forever!

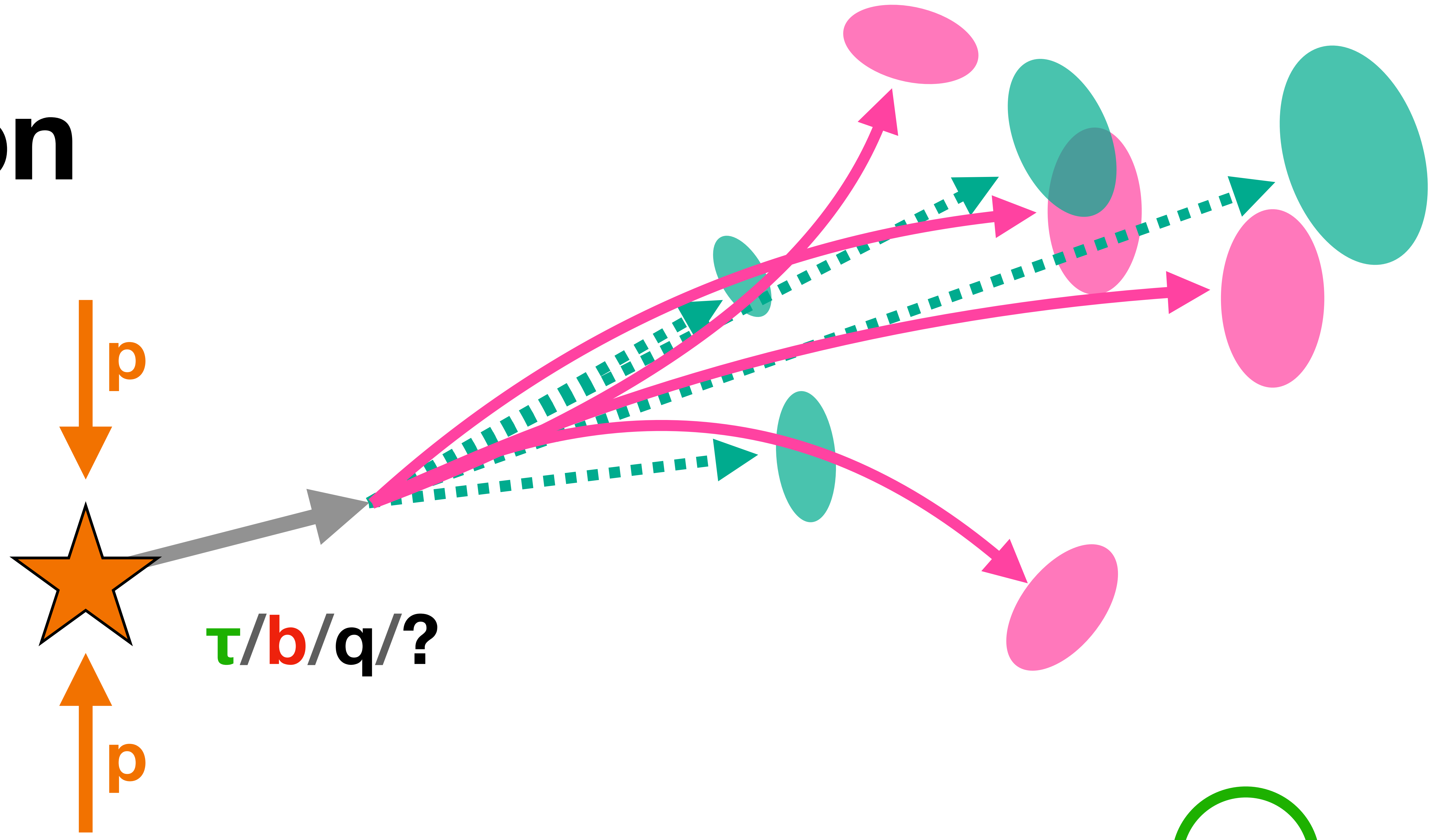
Outline

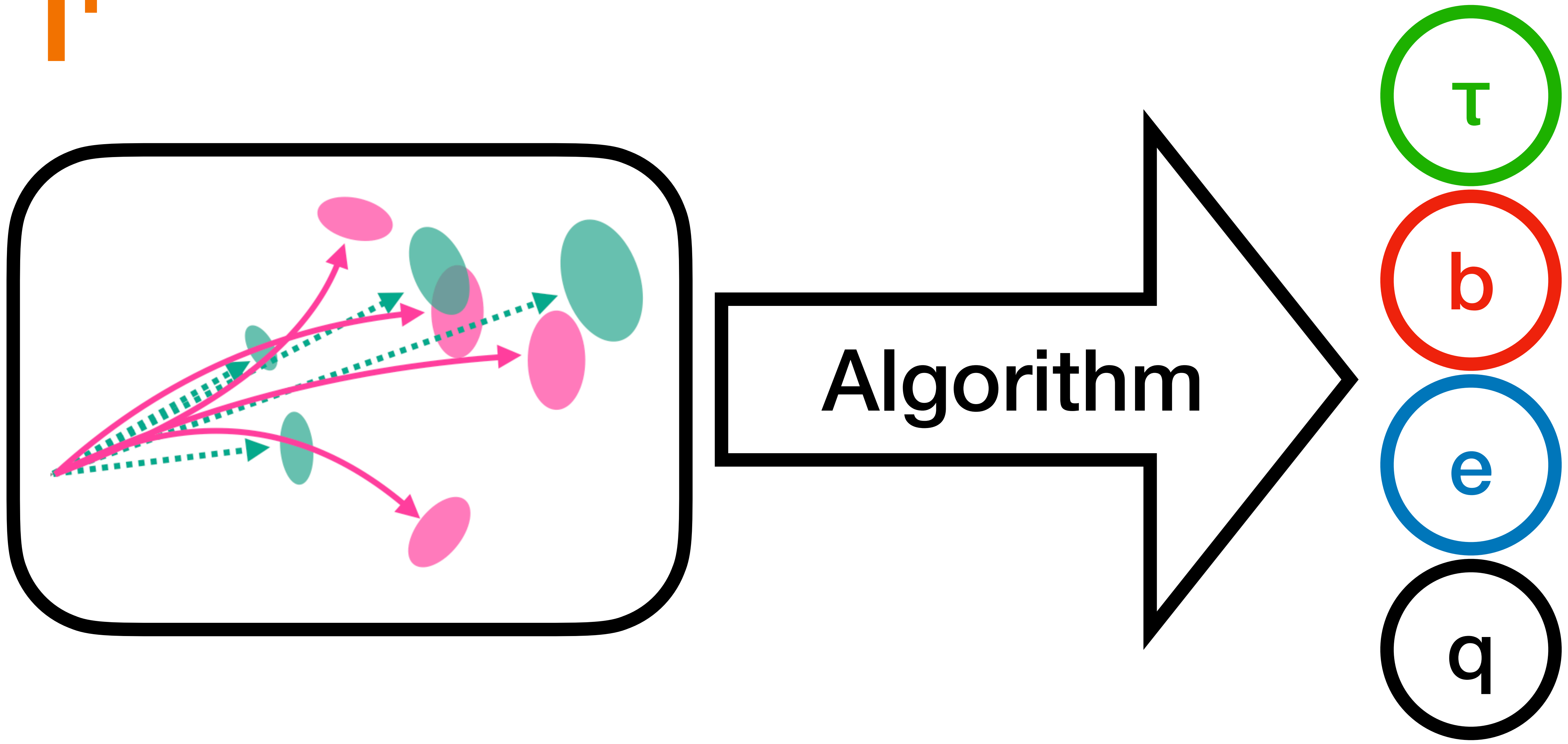


Caveat: I work on ATLAS/LHC, so this is an openly LHC-biased talk. But my goal is to make the lessons accessible!

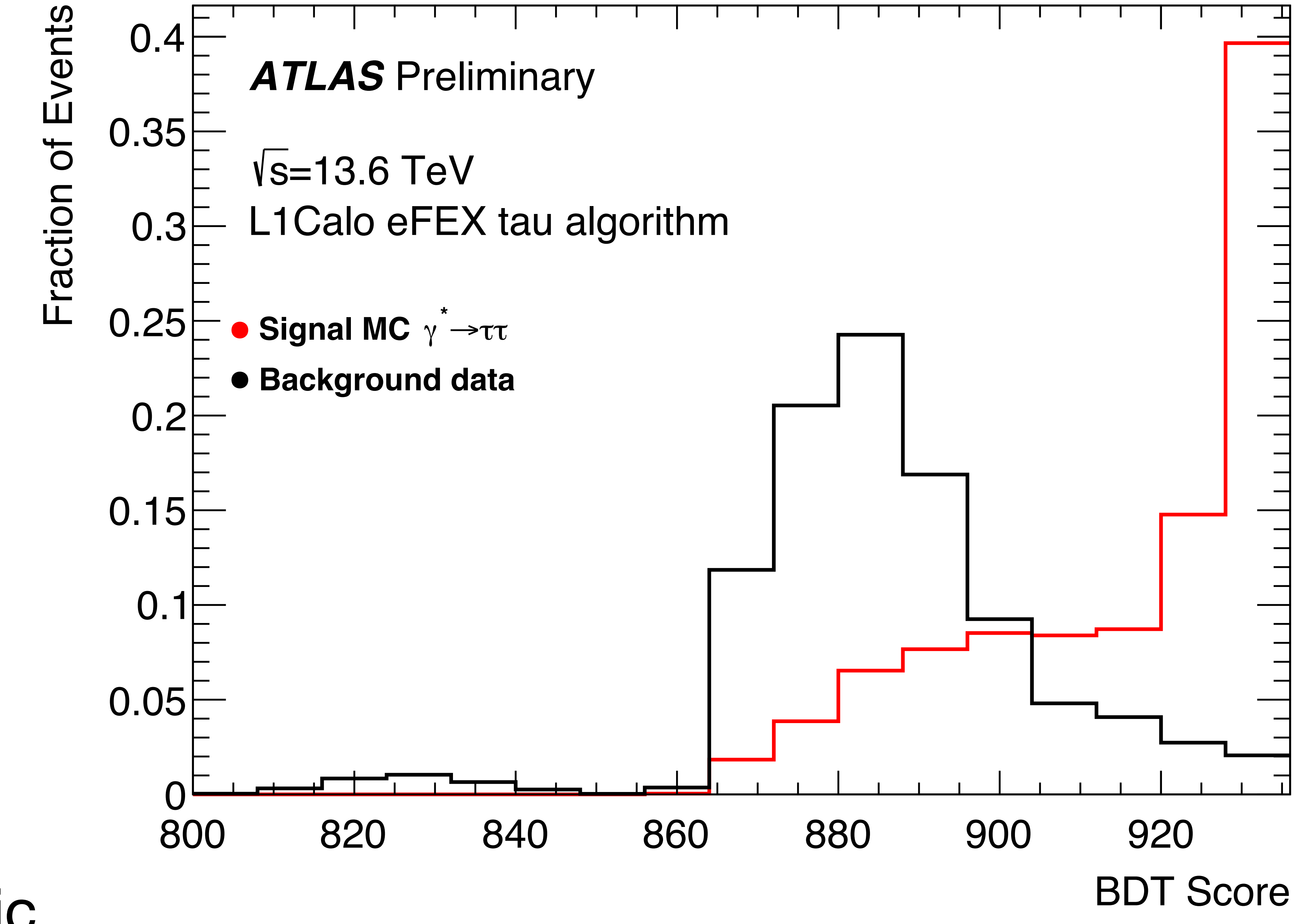
Particle Identification

- LHC triggers must differentiate different collections of particles / detector signals from overwhelming backgrounds
  - Background: light quarks, gluons, noise, combinatorics
  - Signals: τ lepton, bottom quark, electron, …
- ML is very well suited to these tasks



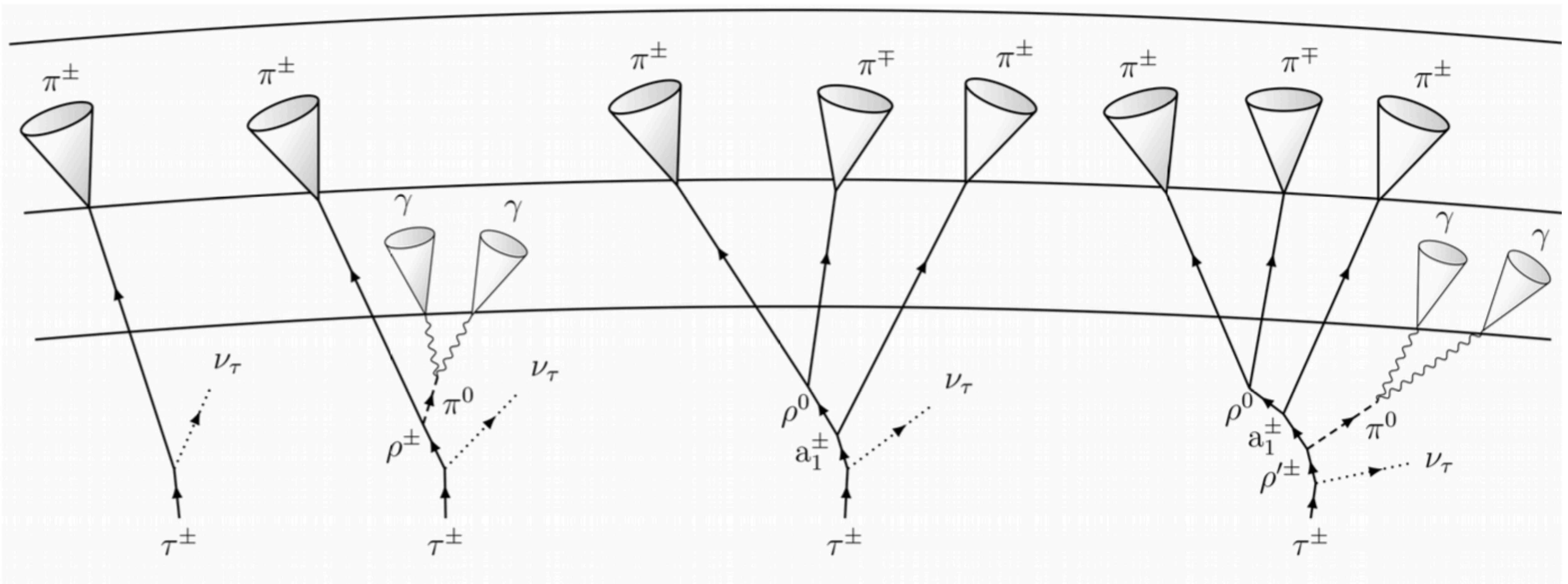


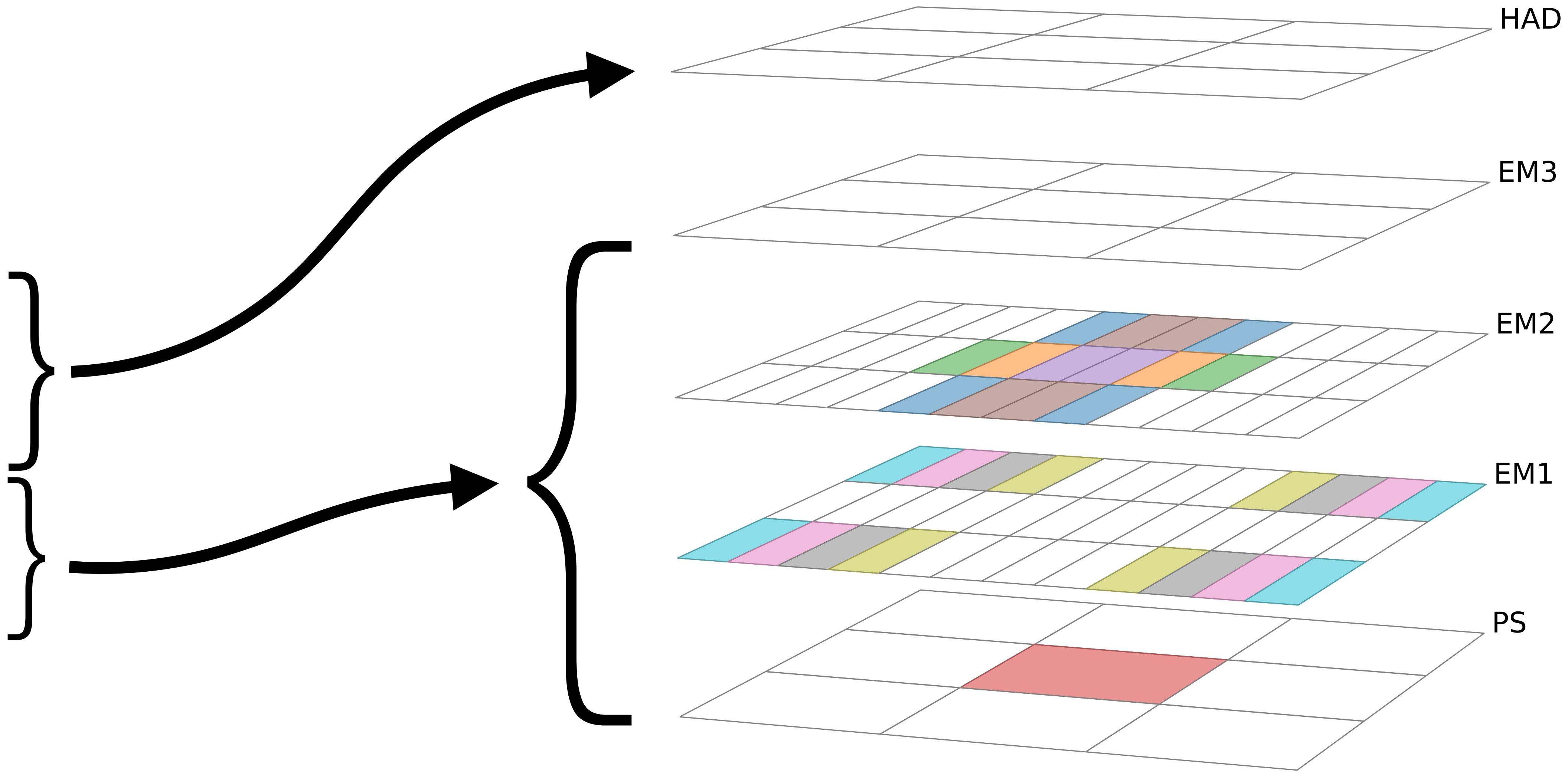
Hadronic τ BDT



- Tau leptons decay to hadrons ~65% of time (τ_h)
  - Difficult to distinguish from hadronic jets
  - Need to combine information from multiple different subdetectors
- Critical for many signals, eg. HH→bbττ
- BDT developed for identification of hadronic taus from energy in specific regions of calorimeters (+ total energy)
- Translated to firmware with conifer ( )

1-prong (+ π⁰) *3-prong (+ π⁰)*

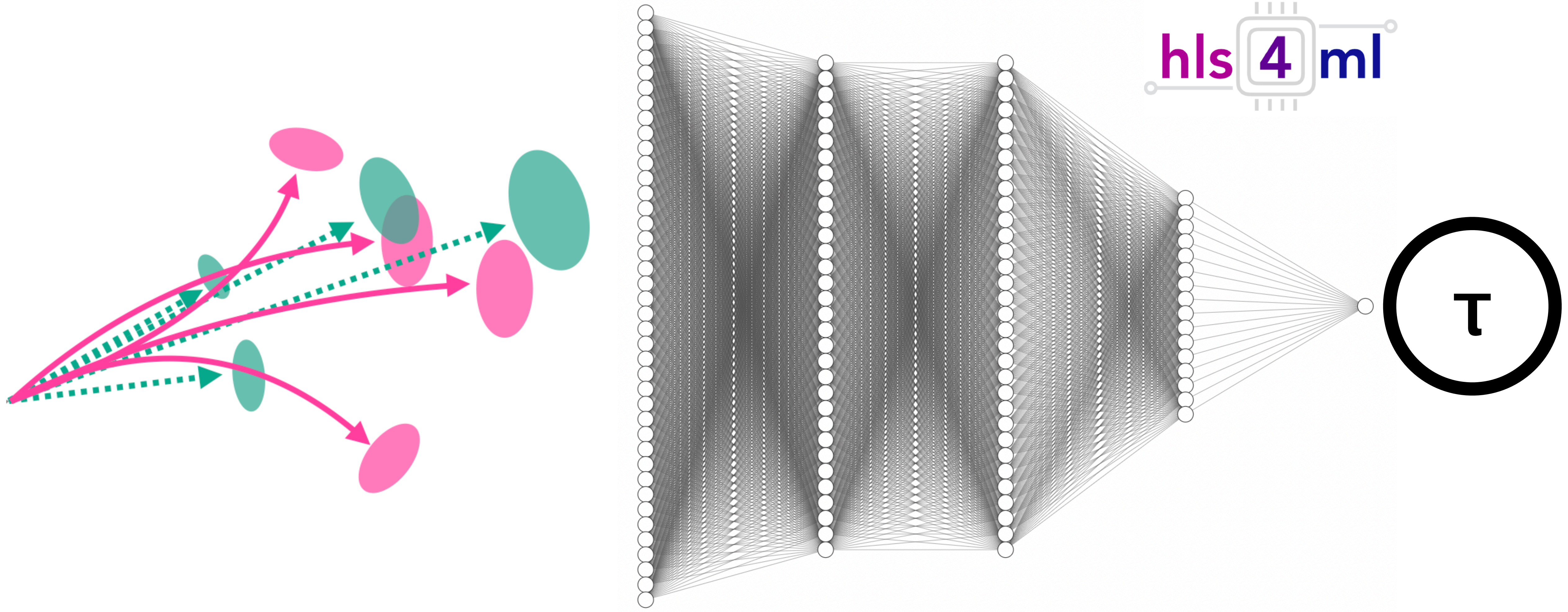


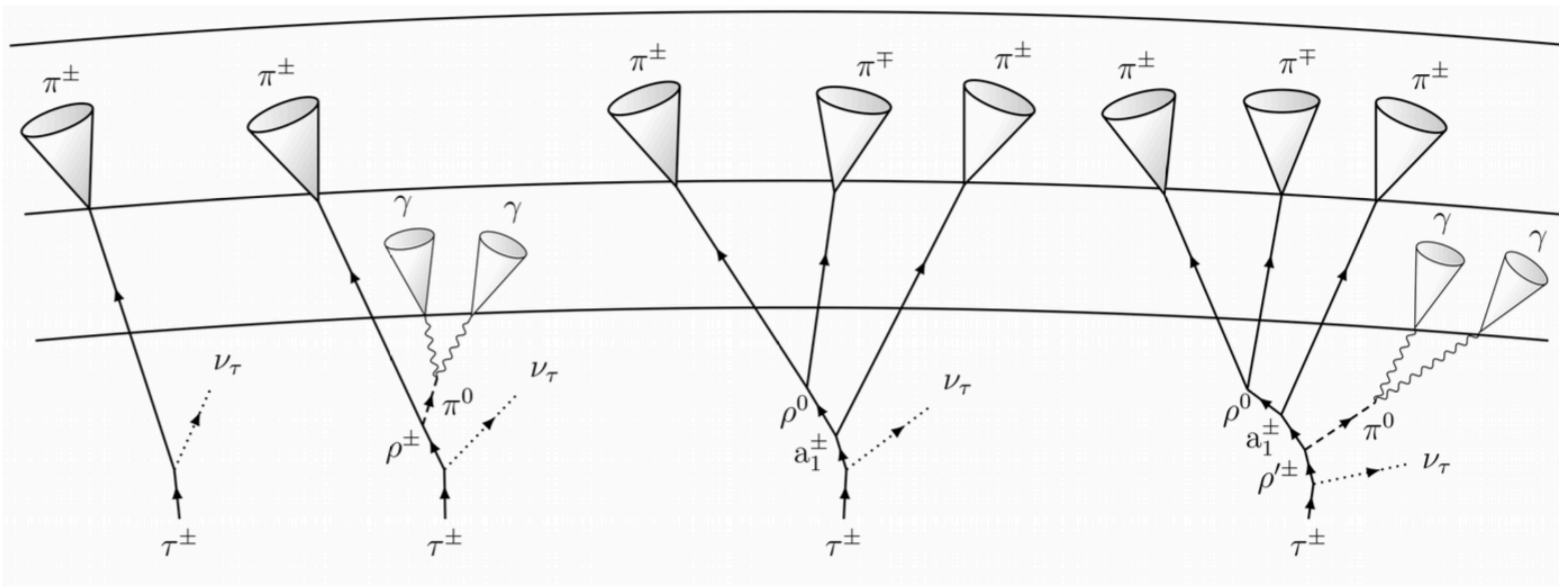


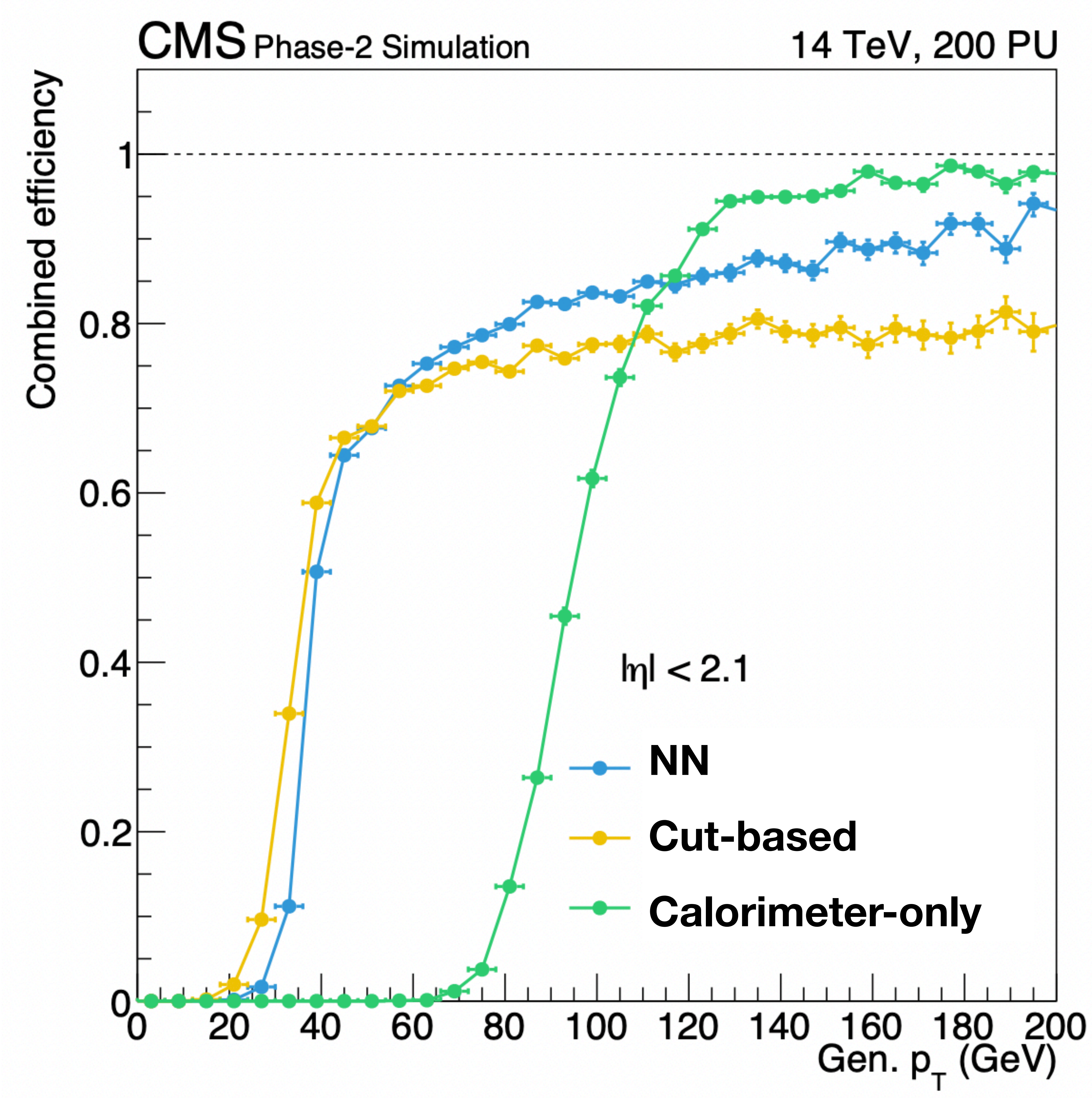
ATLAS L1Calo Trigger Public Results

Hadronic τ NN

- **NN algorithm** using 10 particles around a seed capable of accepting more τ leptons than traditional **cut-based method**
- Network is 3 layer dense model, uses information about particle p_T, η, ϕ, and type
- Outputs decision in 38 ns (9 clocks @ 240 MHz)



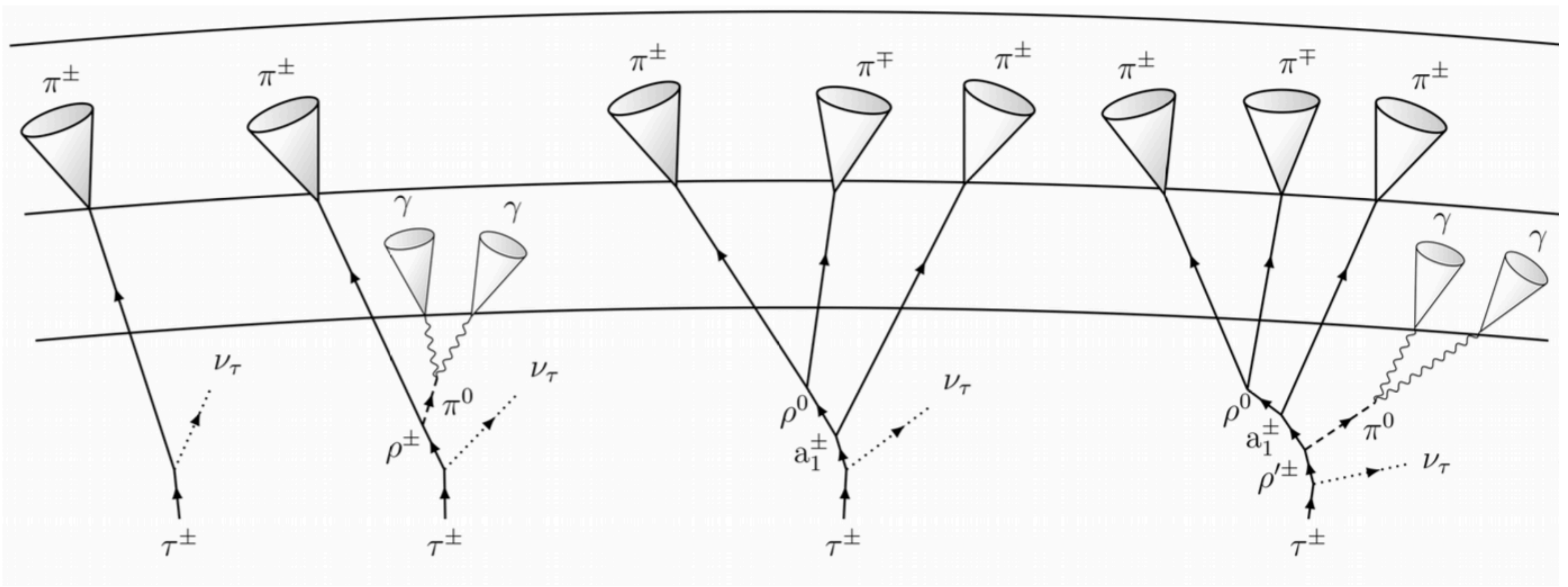


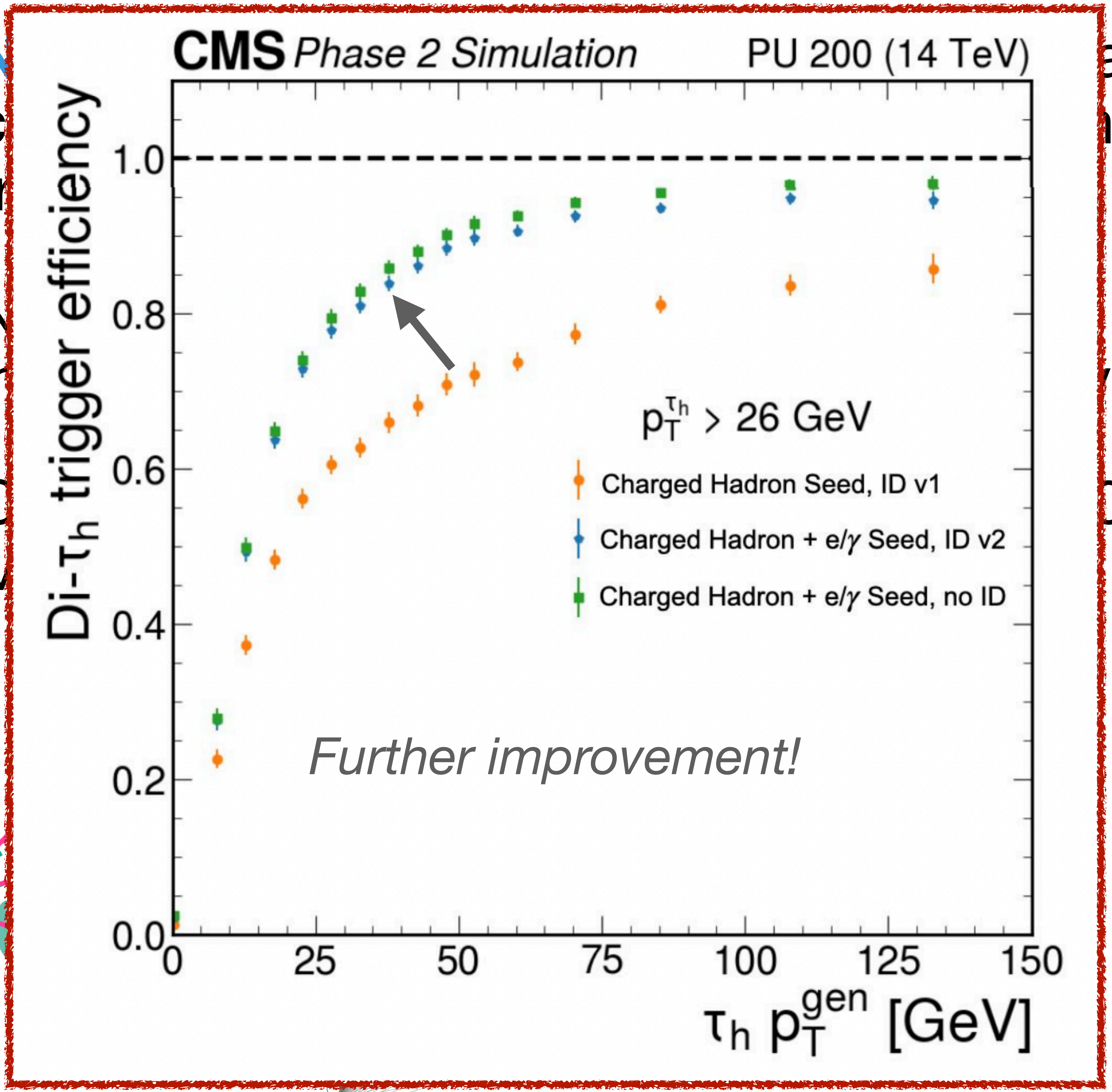


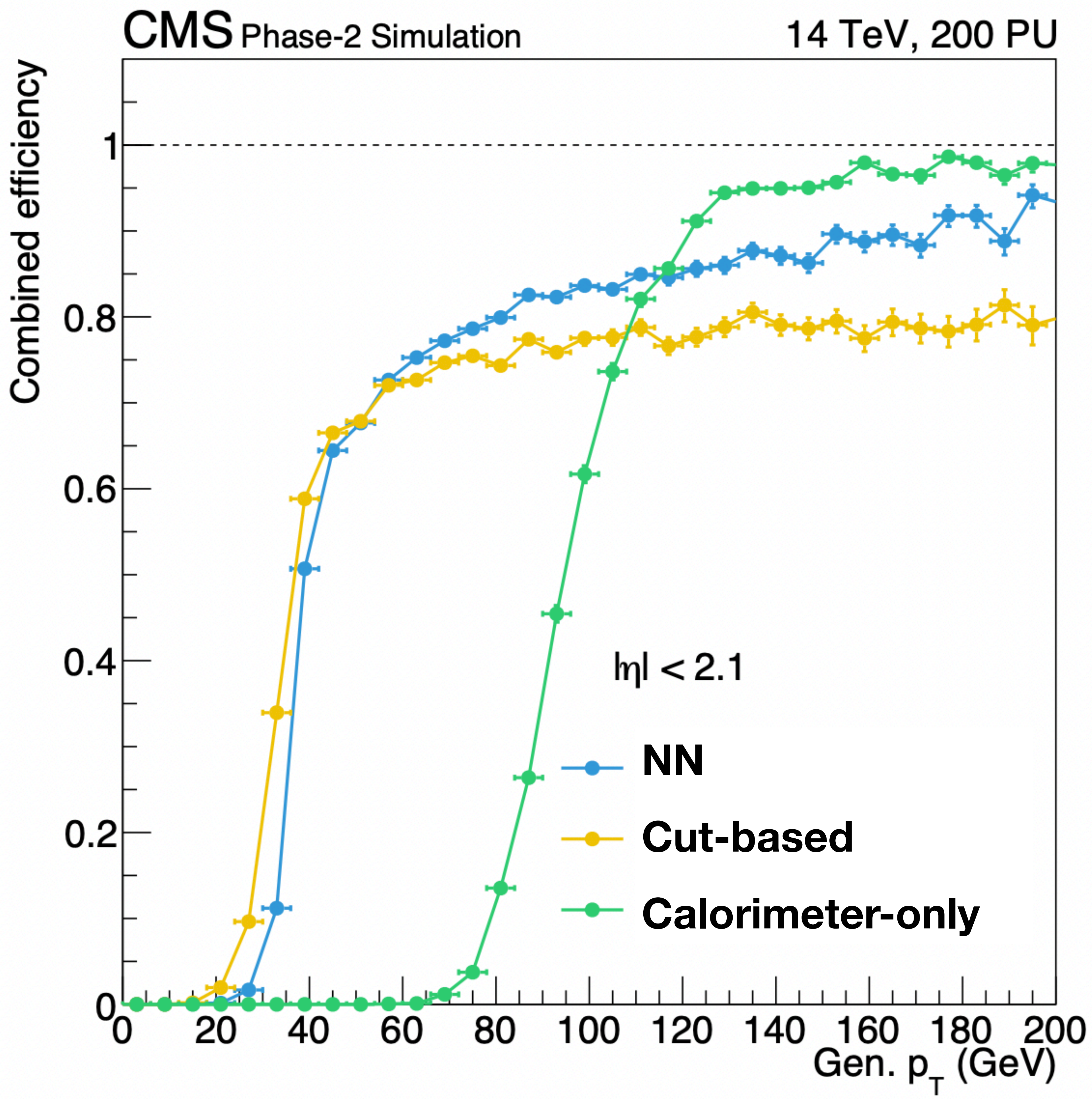
CMS TDR-021

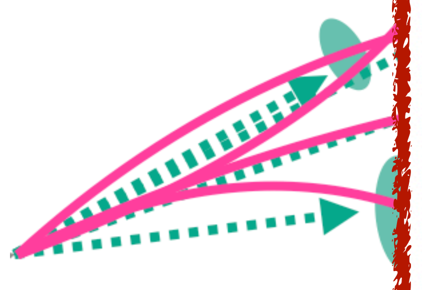
Hadronic τ NN (v2)

- N a seed
  c n
  tr
- N
  ir ype
- C 0
  M







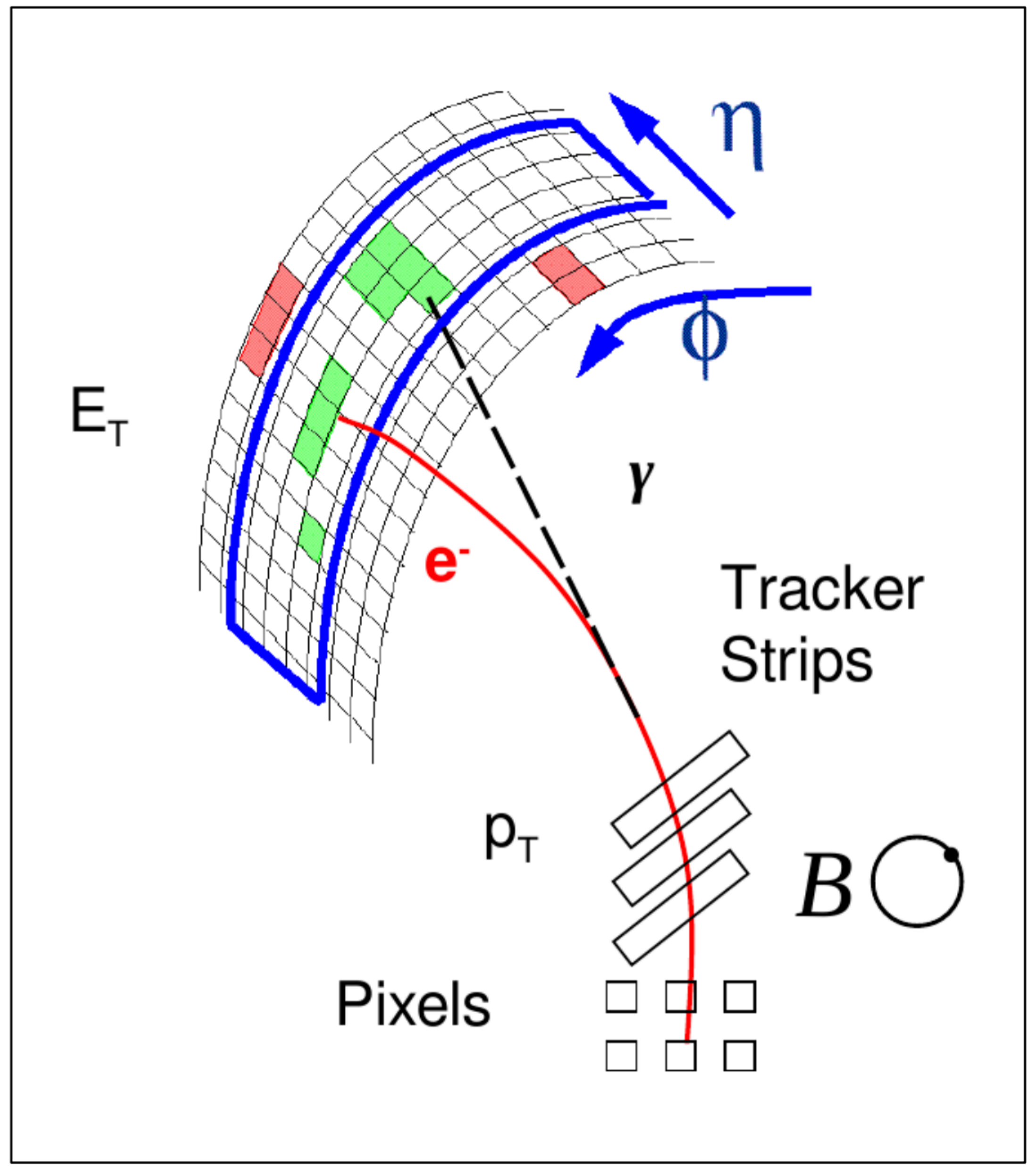


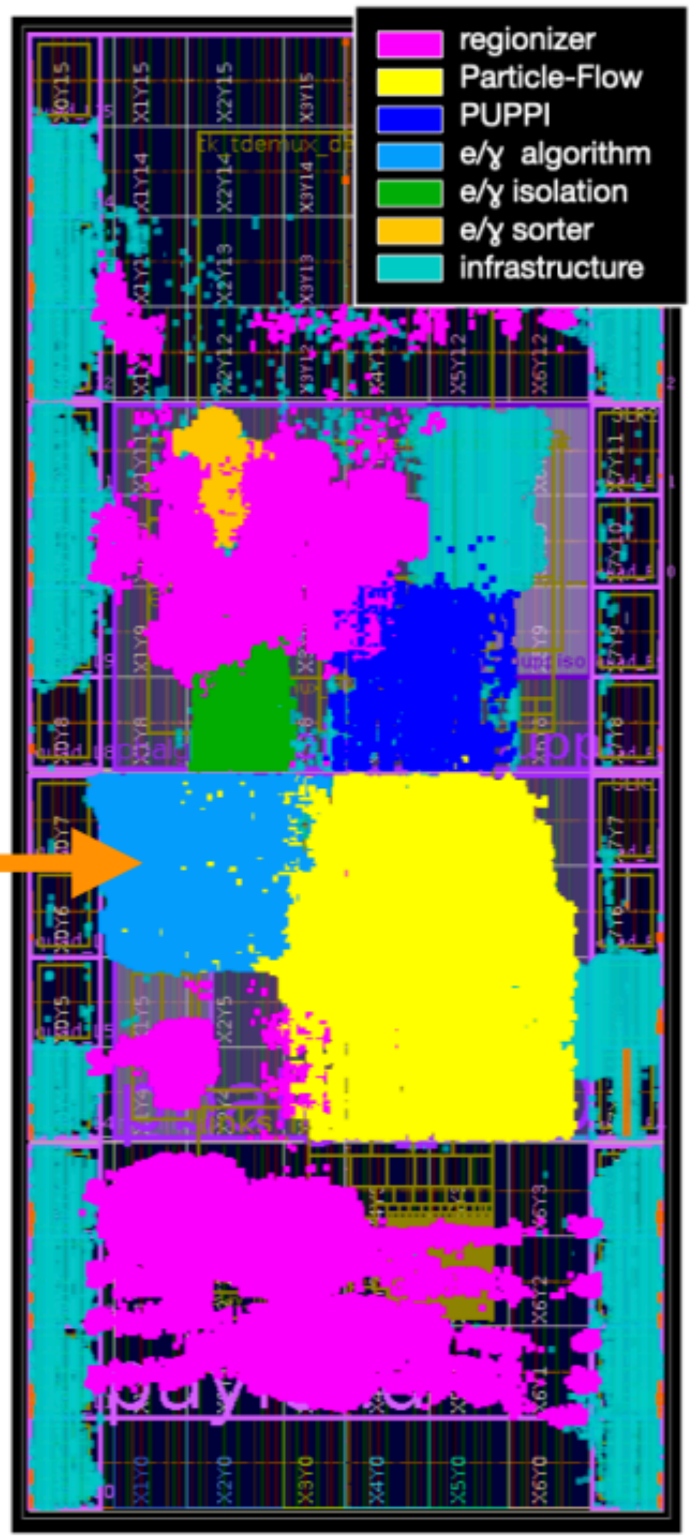
CMS DP-2024/018

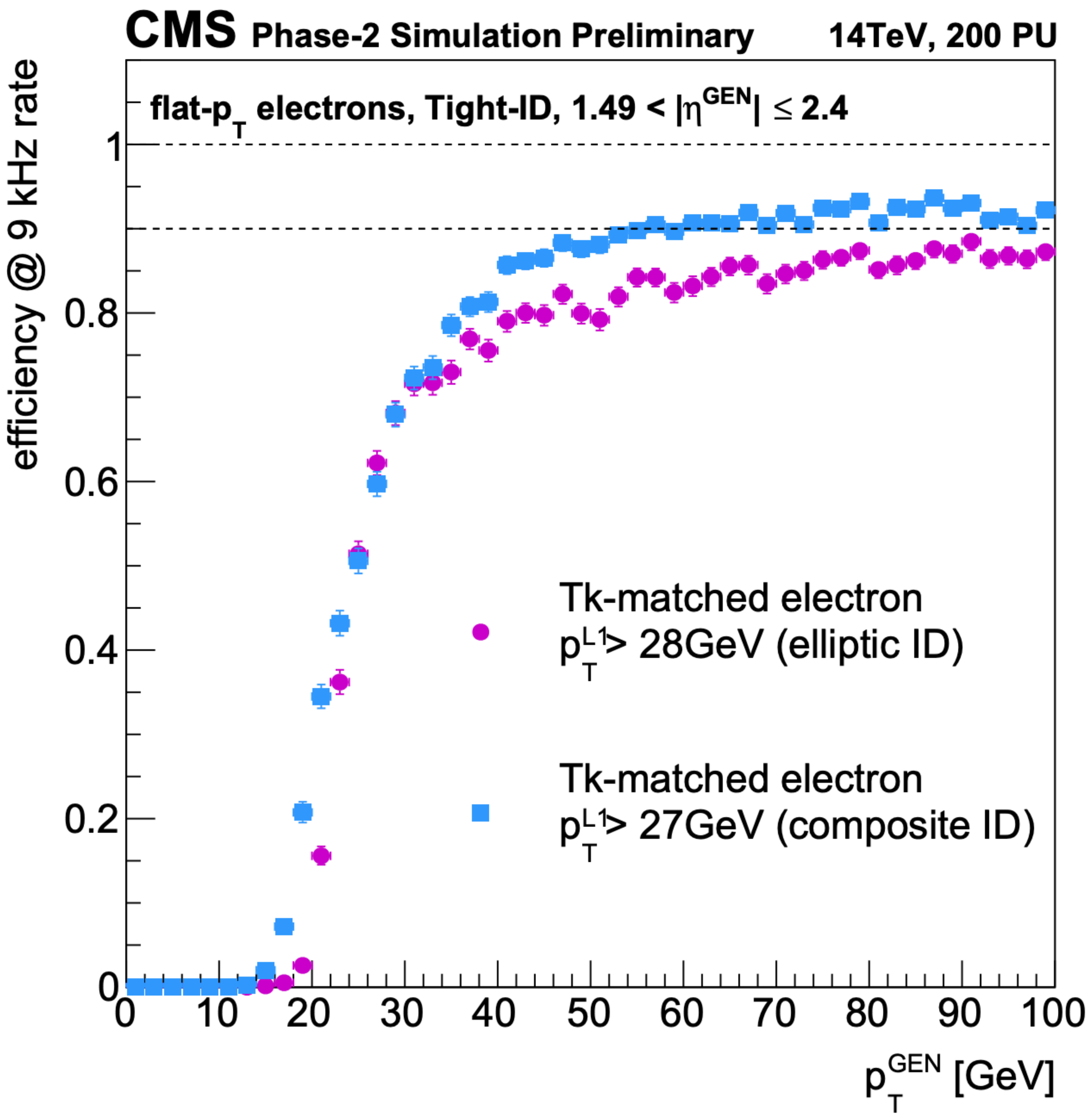
CMS TDR-021

Electron BDT

- Electrons are also complex signatures
  - Signals span multiple sub detectors (tracker & calorimeter)
  - Undergo bremsstrahlung ($e \to e + \gamma$)
- Electron ID is well-suited to ML
  - Handles correlations between different inputs
  - **5-10% improvement in plateau efficiency**
- Important for many different physics signatures

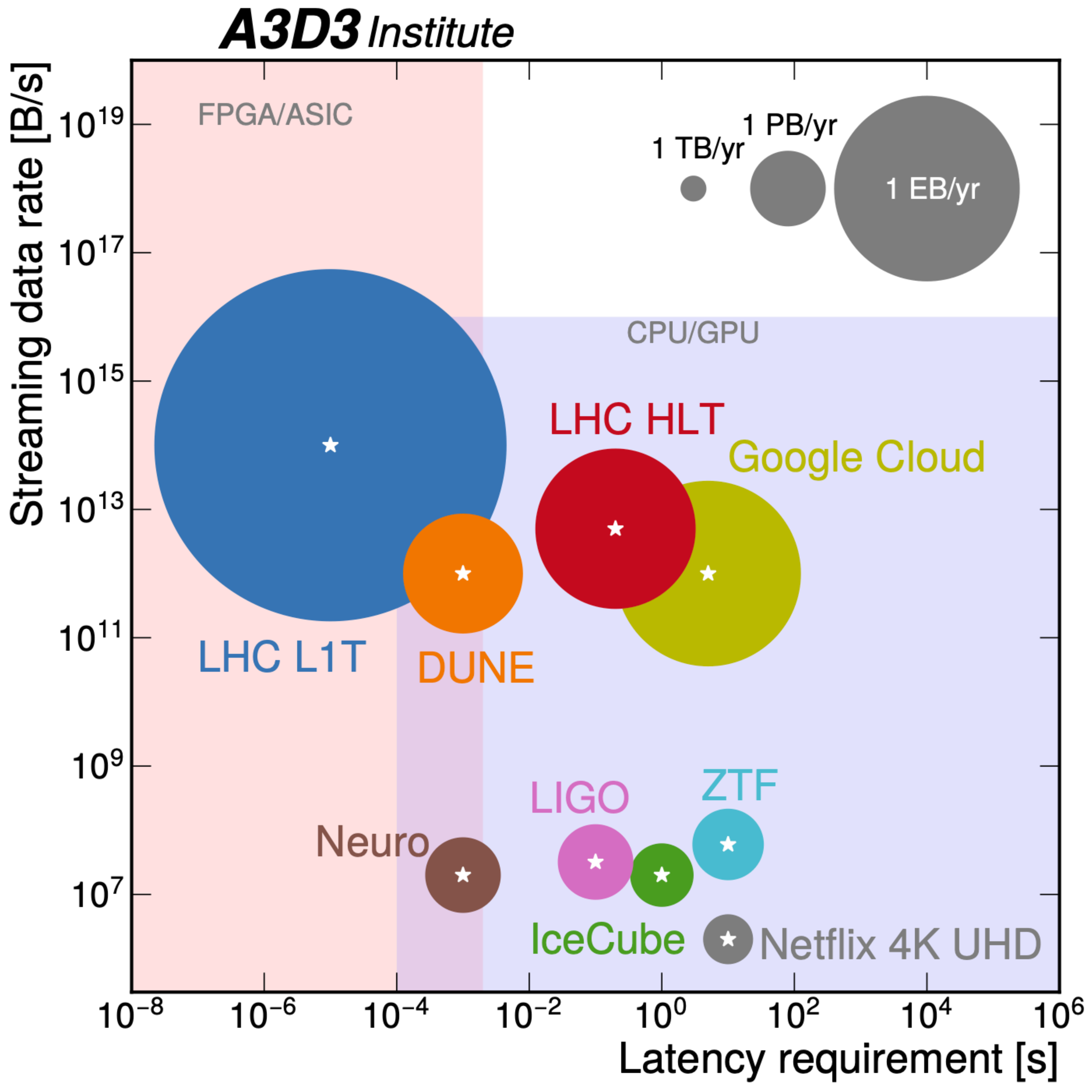




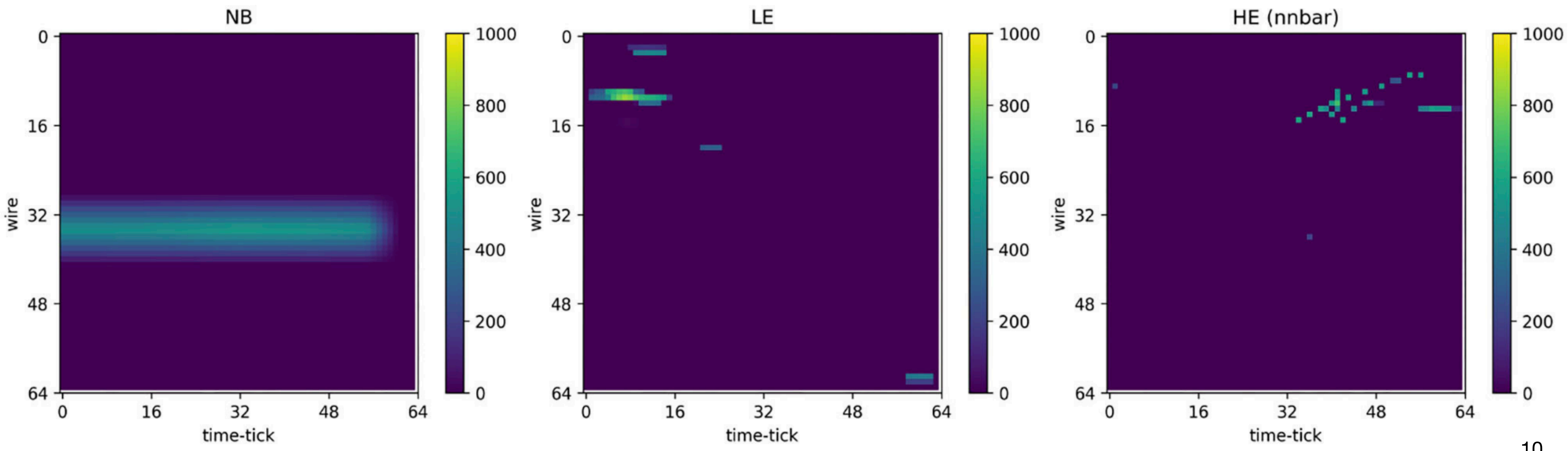


CMS DP-2023/047

LArTPC Neutrinos

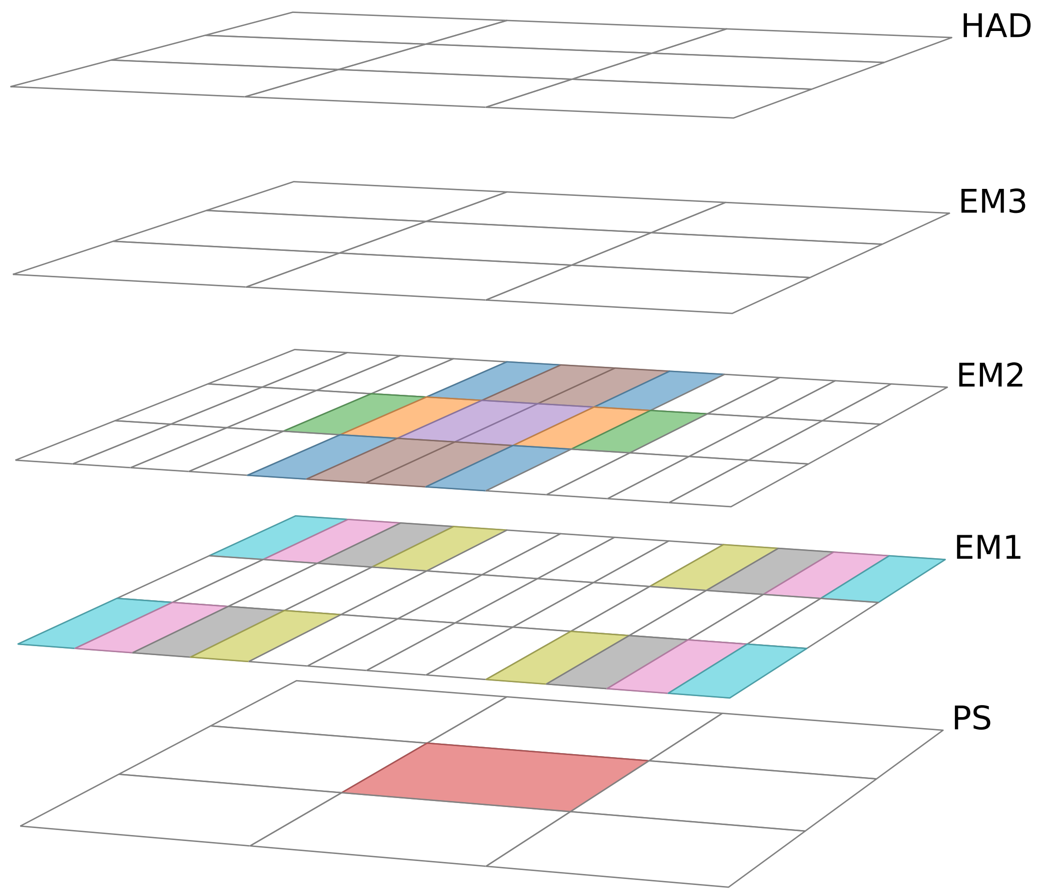


- DUNE will bring LHC-scale data rates to neutrino physics
- Fast identification of particles (particularly in dense environments) potentially important for maximizing experimental capabilities (eg. fast superova neutrino detection)
- Requirements:
  - Reject noise (NB) with >99.99% efficiency
  - Classify low-energy supernova neutrino (LE) with 90% efficiency
  - Process incoming image within 32 μs
- 2DCNN [A. Malige, FastML 2024] capable of meeting performance benchmarks, latencies between 3-5 μs
  - QKeras employed for QAT, tested on Alveo U250 & U55C

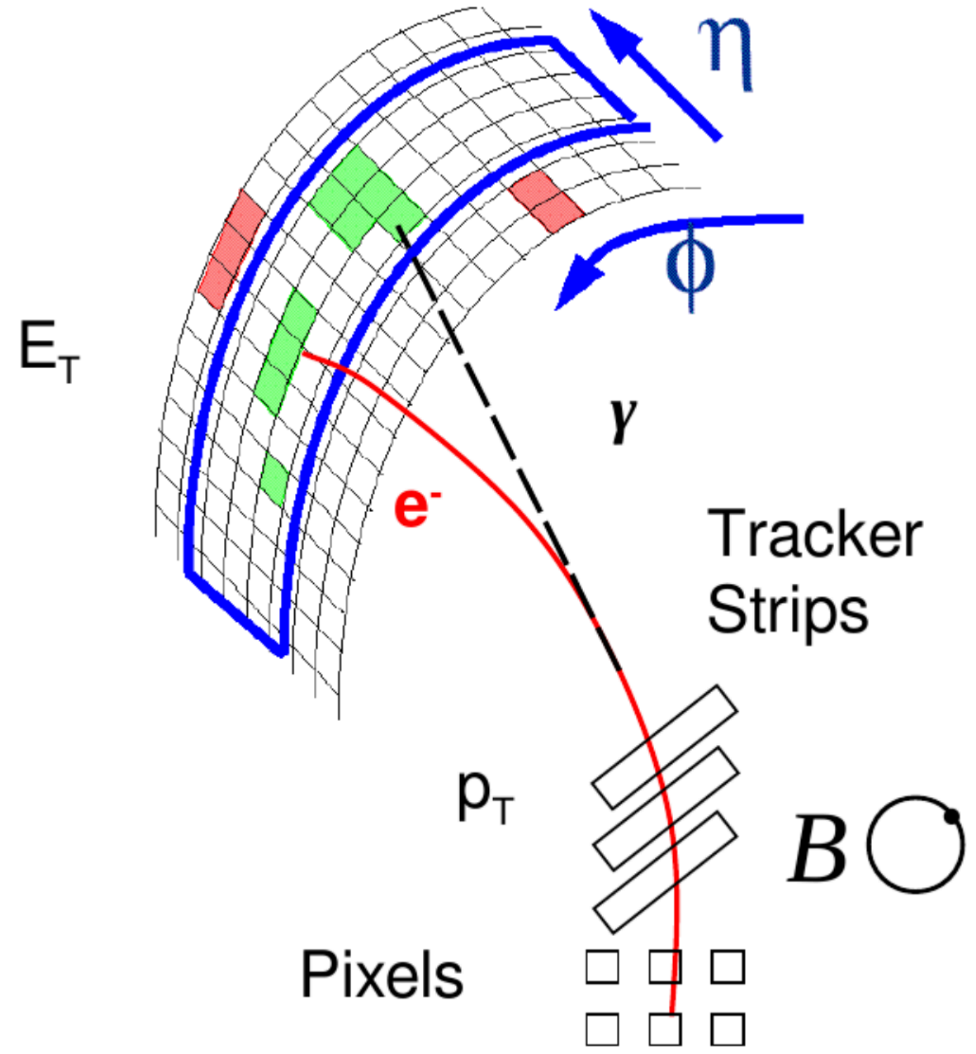


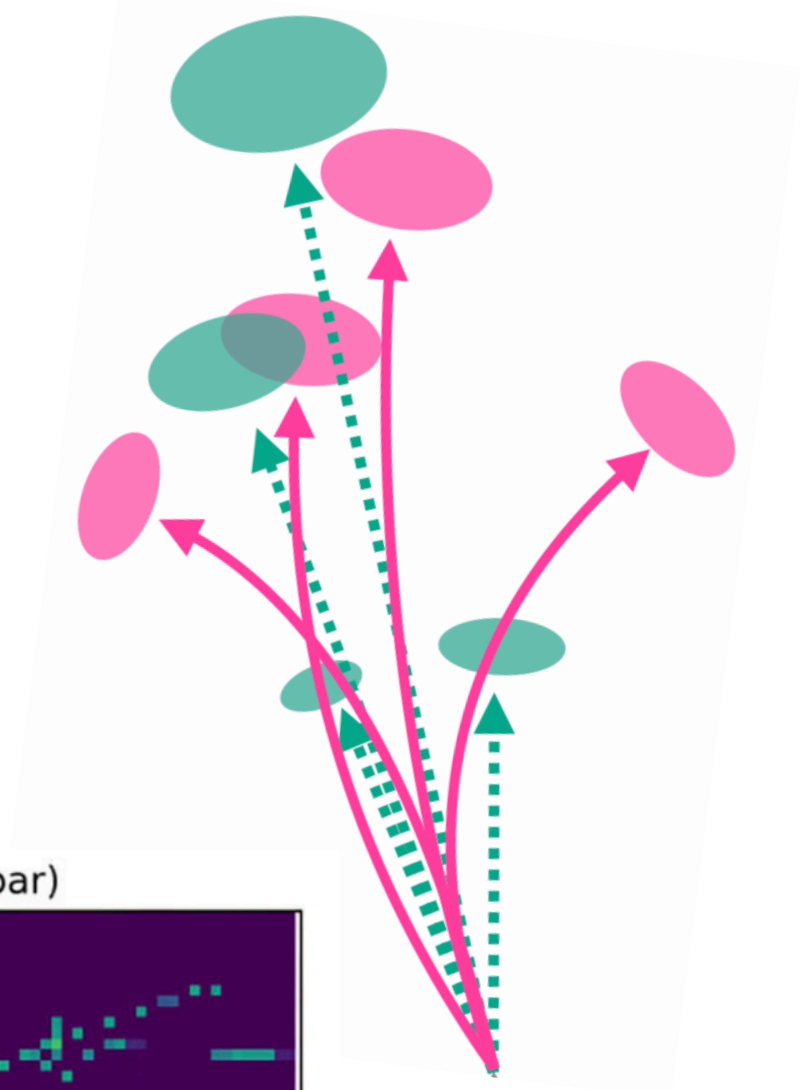
2201.05638

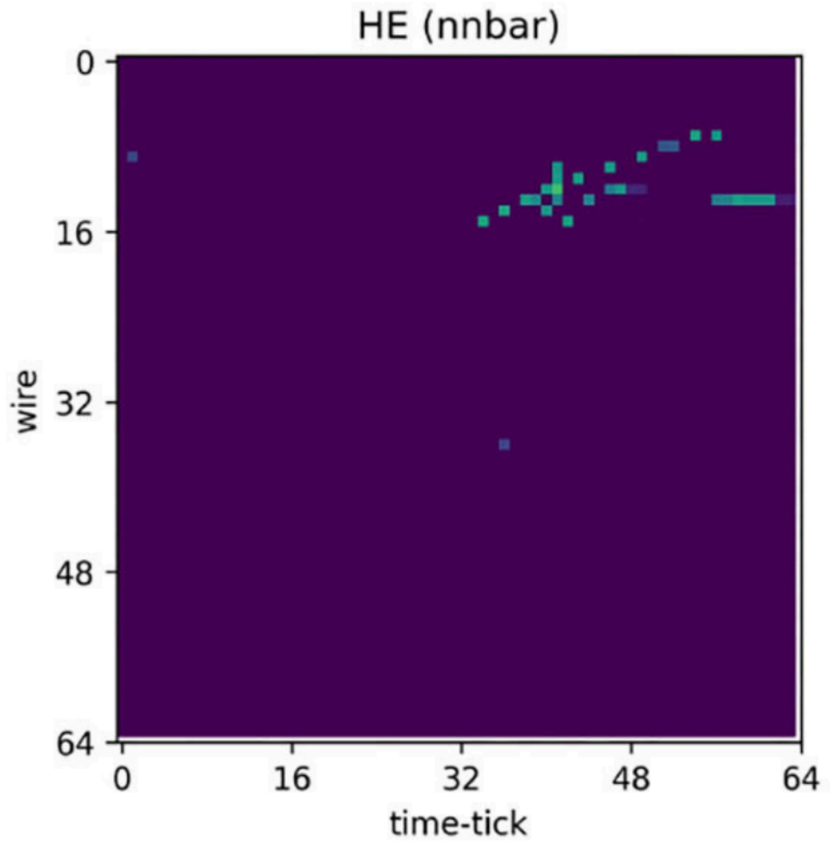
Particle Identification Lessons



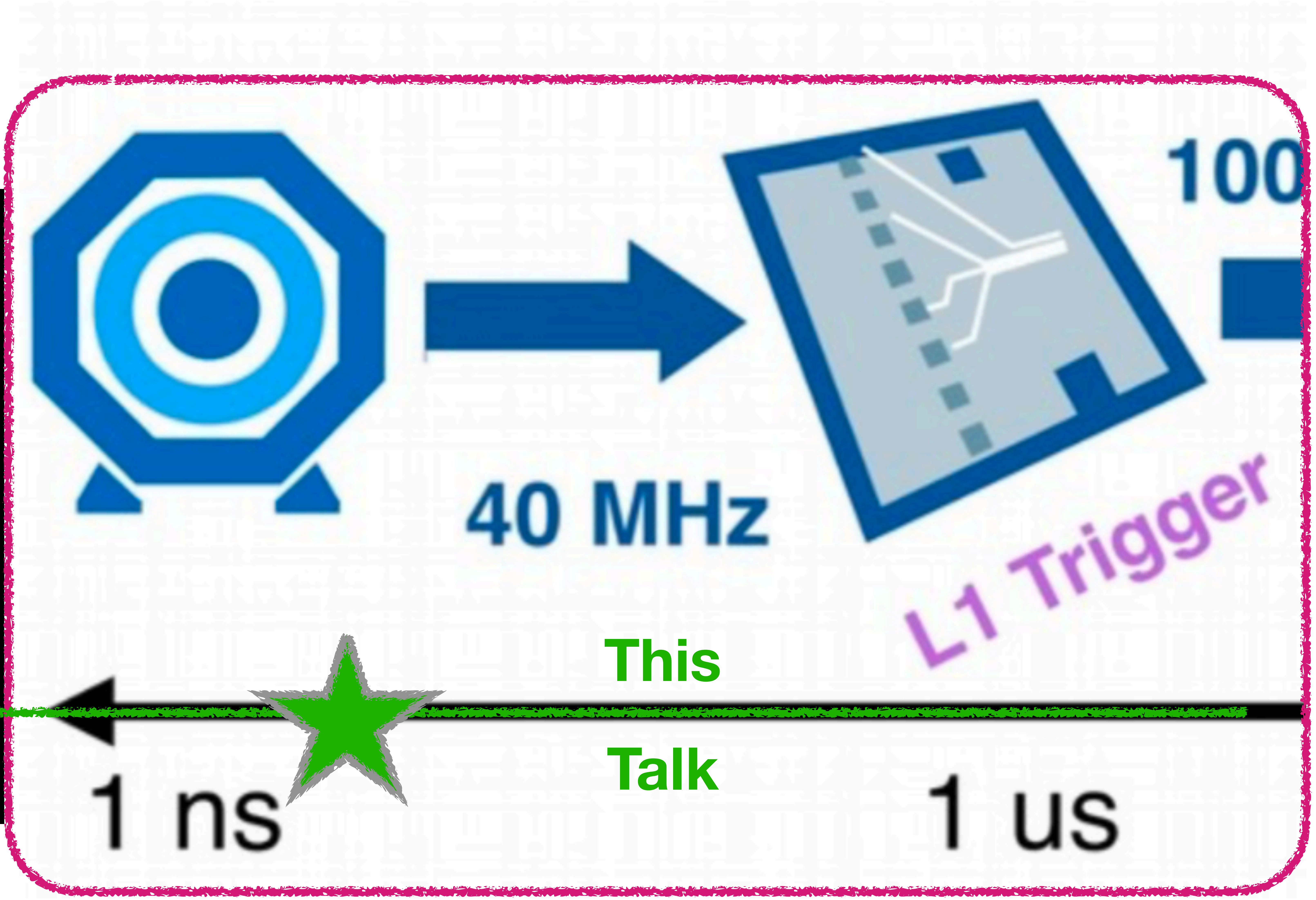
- No one size fits all solution
  - True across applications
- Finding “best” solutions requires complete picture of task
  - Eg. Calorimeter-based taus different from particle-based taus, different from electrons, b-jets (see backup or Javier’s talk) …
  - *Codesign* critical for optimizing performance



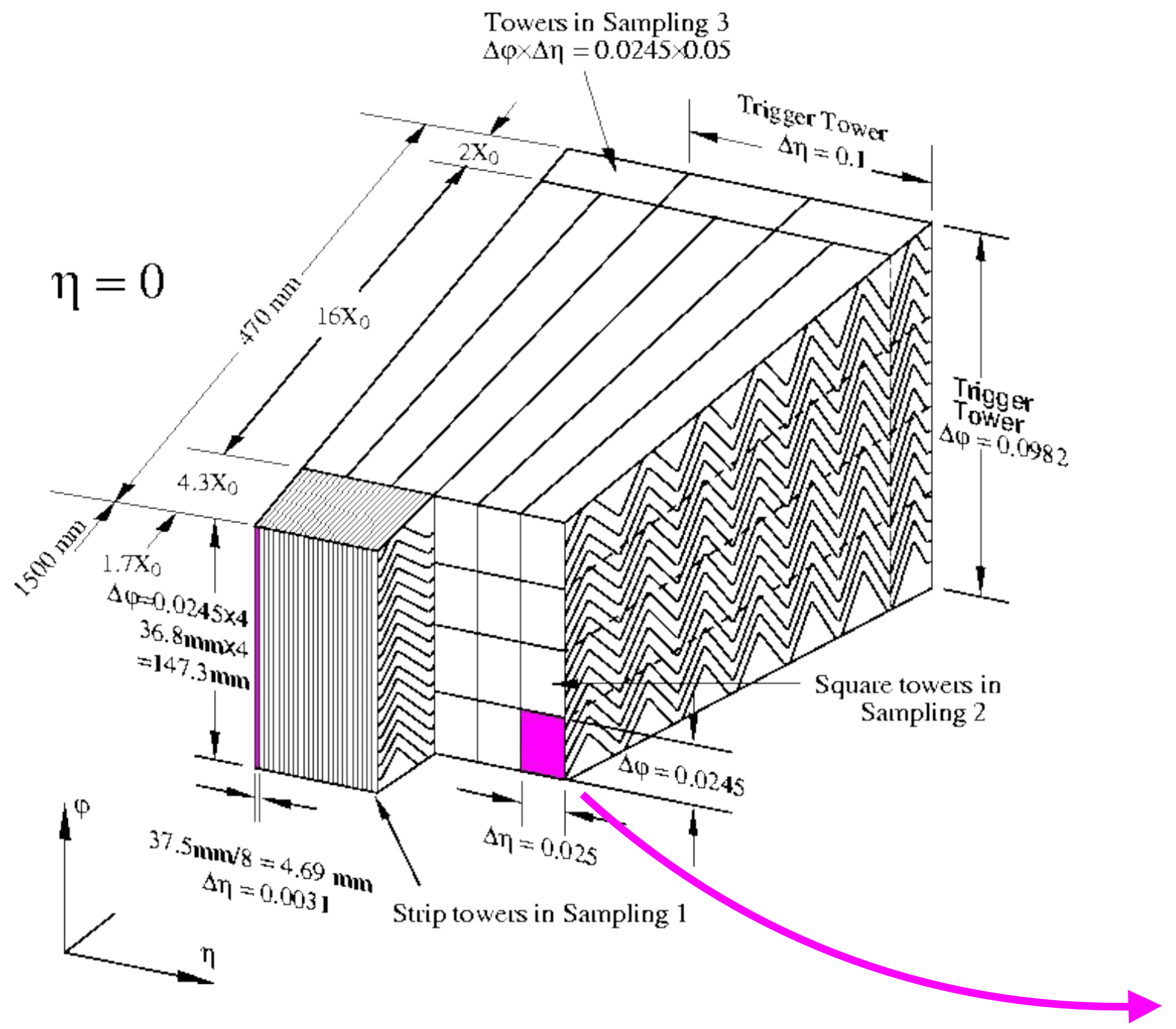


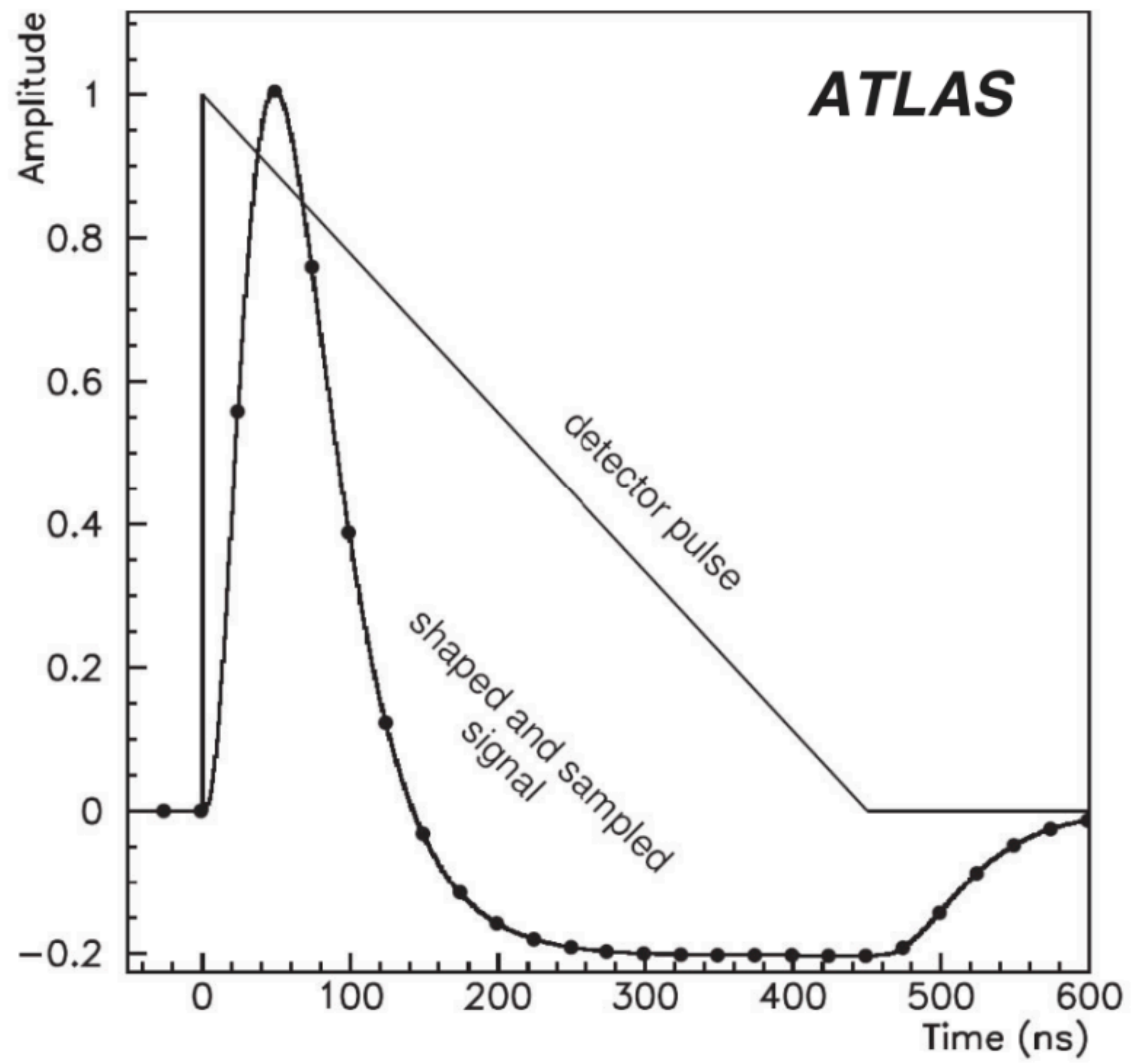


Outline

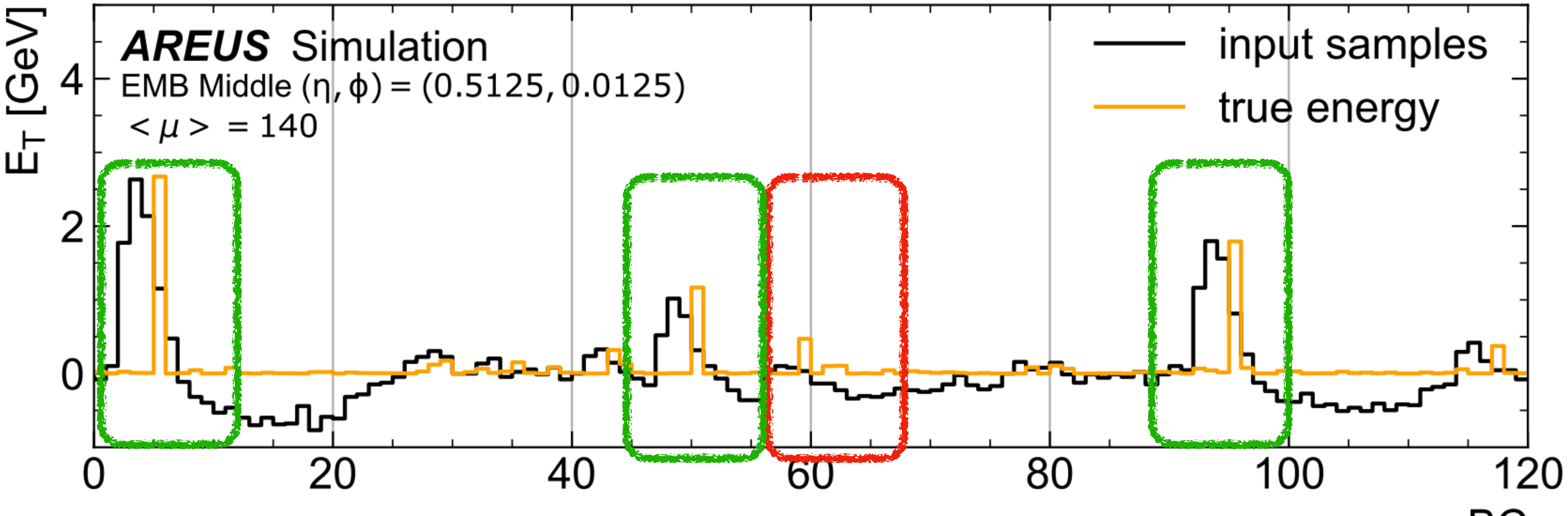


LAr Peak Finding

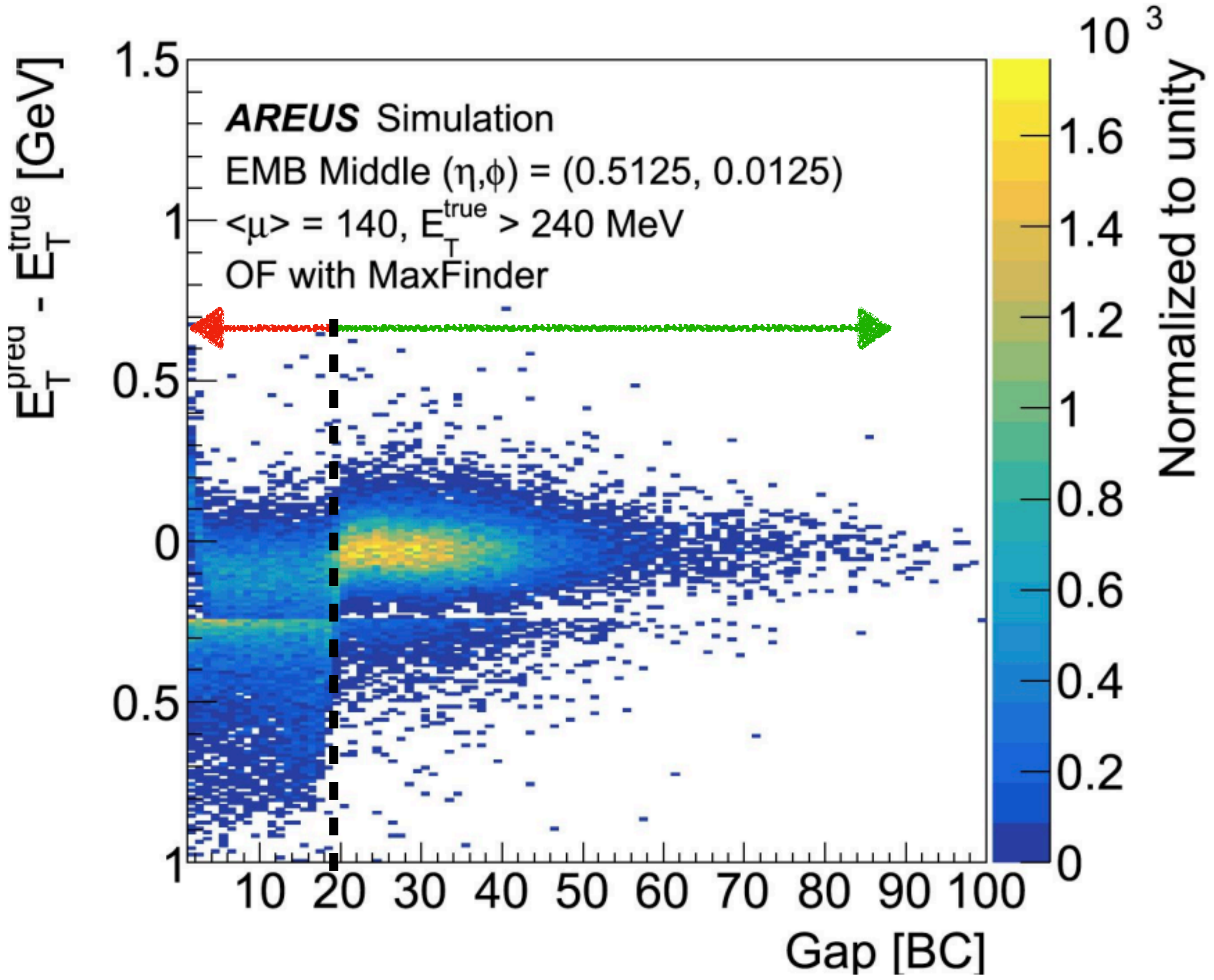




- ATLAS LAr calorimeter needs to measure time and energy of pulses
  - Overlapping pulses difficult for simple, fast algorithms to handle (150 ns = 6 BXs)
- **CNN** and LSTM architectures both able to significantly improve performance
  - Well-suited for data structure, able to account for non-linear correlations

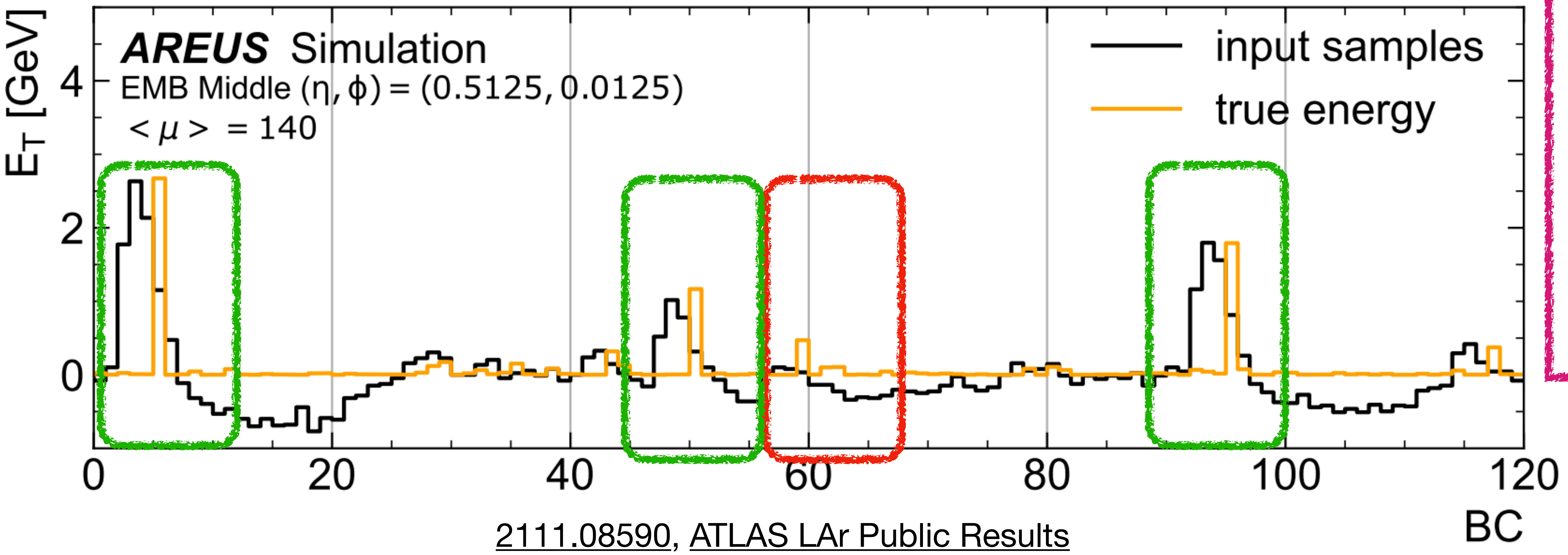


2111.08590, ATLAS LAr Public Results

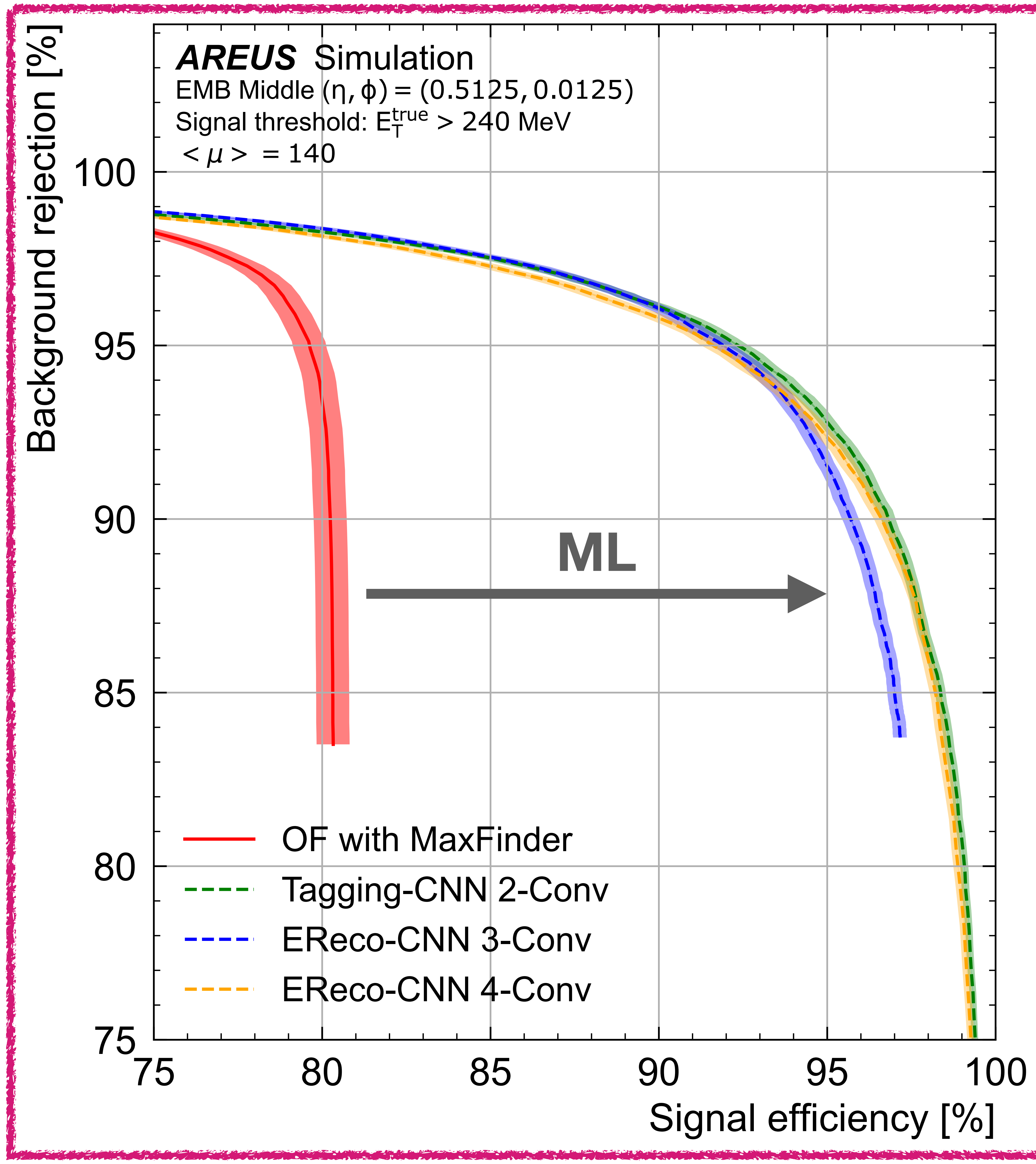


LAr Peak Finding

- ATLAS LAr calorimeter needs to measure time and energy of pulses
  - Overlapping pulses difficult for simple, fast algorithms to handle (150 ns = 6 BXs)
- **CNN** and LSTM architectures both able to significantly improve performance
  - Well-suited for data structure, able to account for non-linear correlations

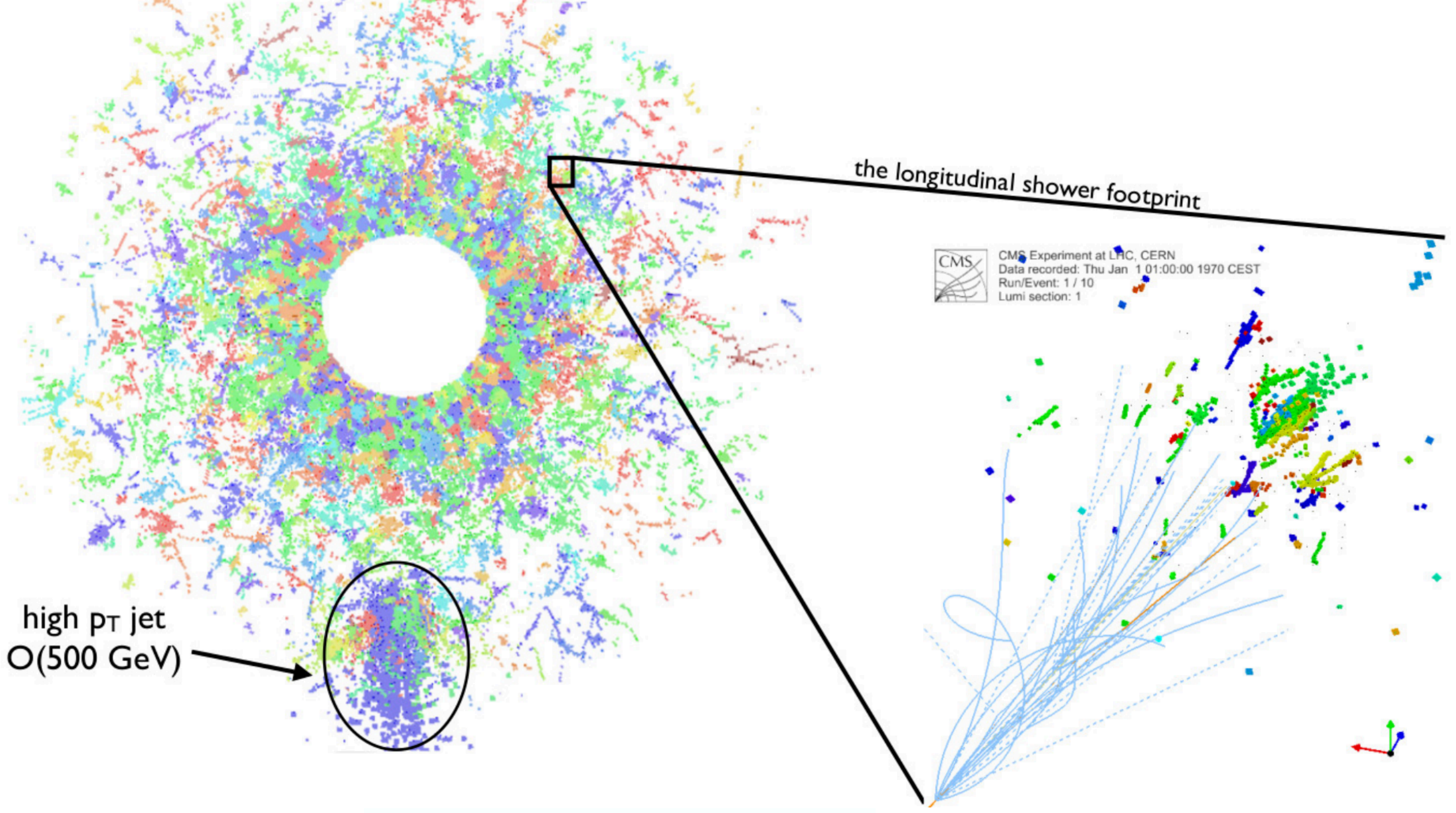


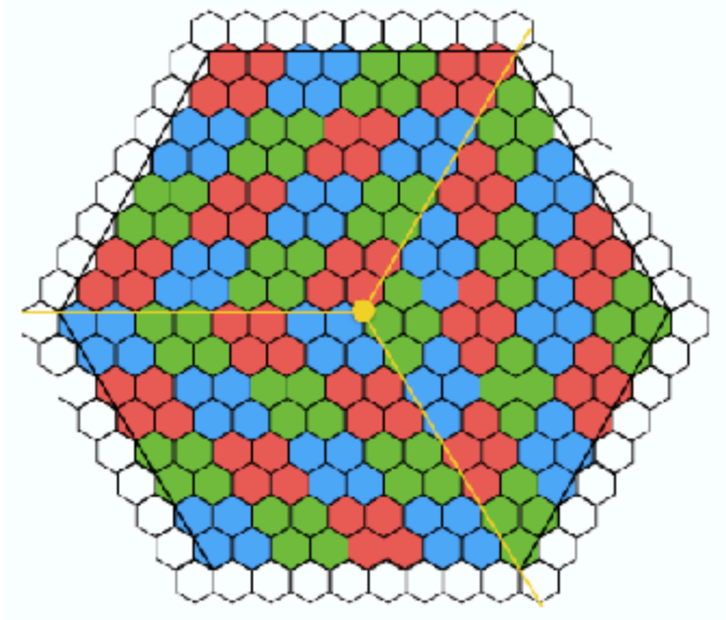
2111.08590, ATLAS LAr Public Results

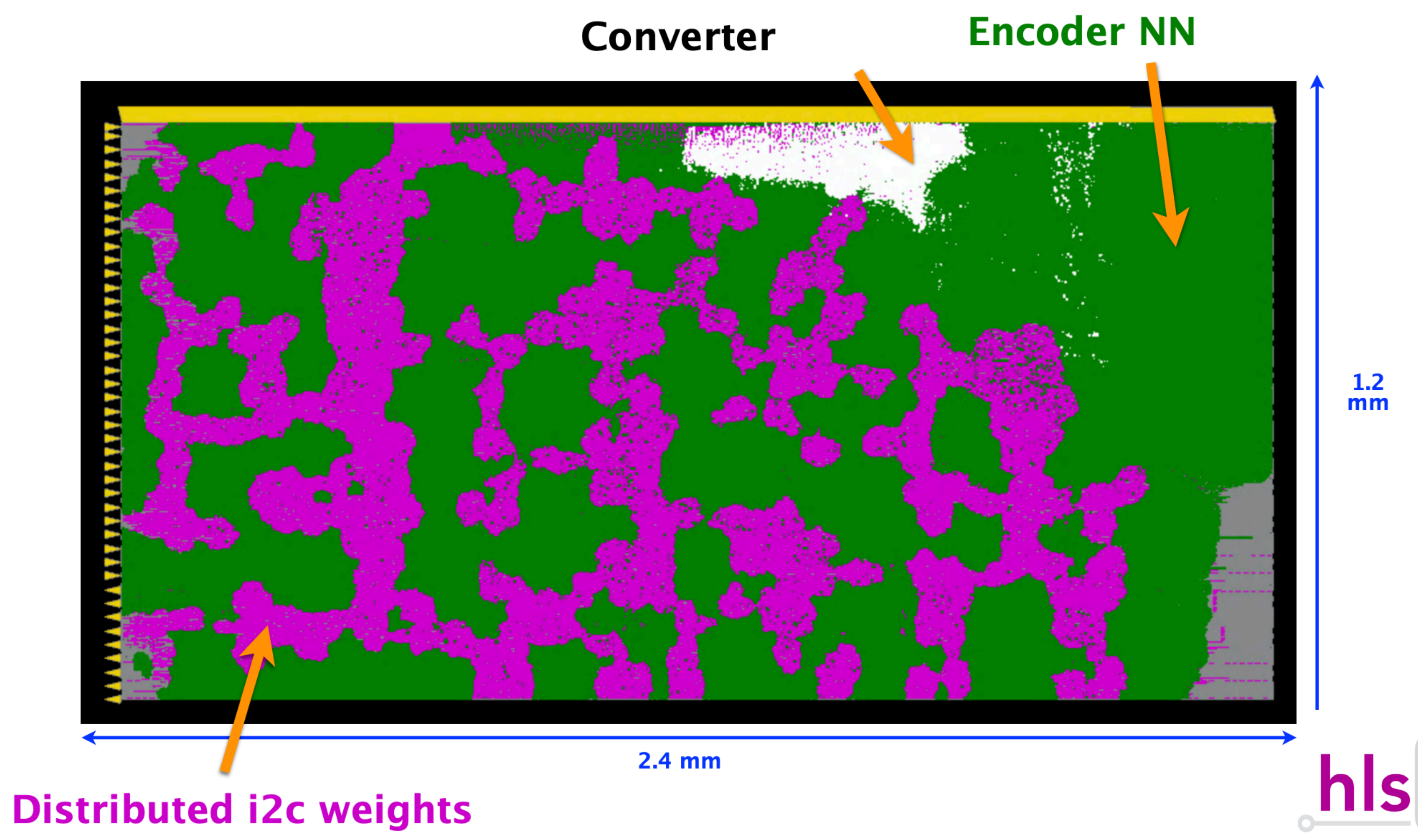


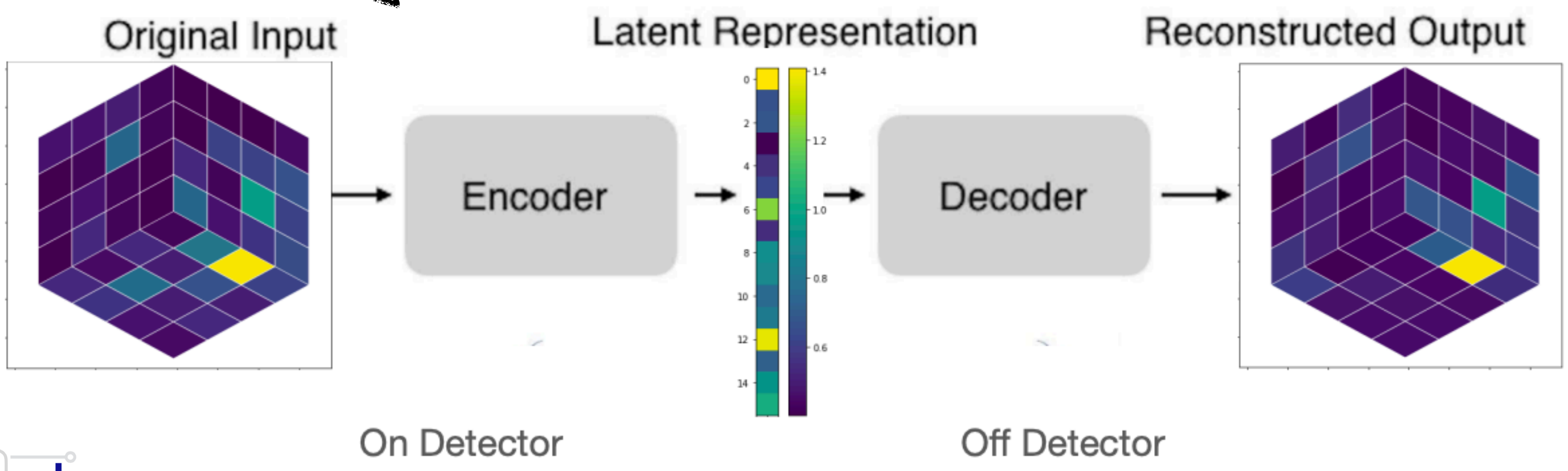
Data Compression

- What if there's simply too much data to get off the detector in the first place?!
  - CMS High Granularity Calorimeter will have 6.5 million readout channels, 50 layers → need some compression
- AEs are lossy compression algorithms (only transmit latent space)
- Model must be run in high radiation environment (ECON-T ASIC, logic triplicated) [2105.01683]





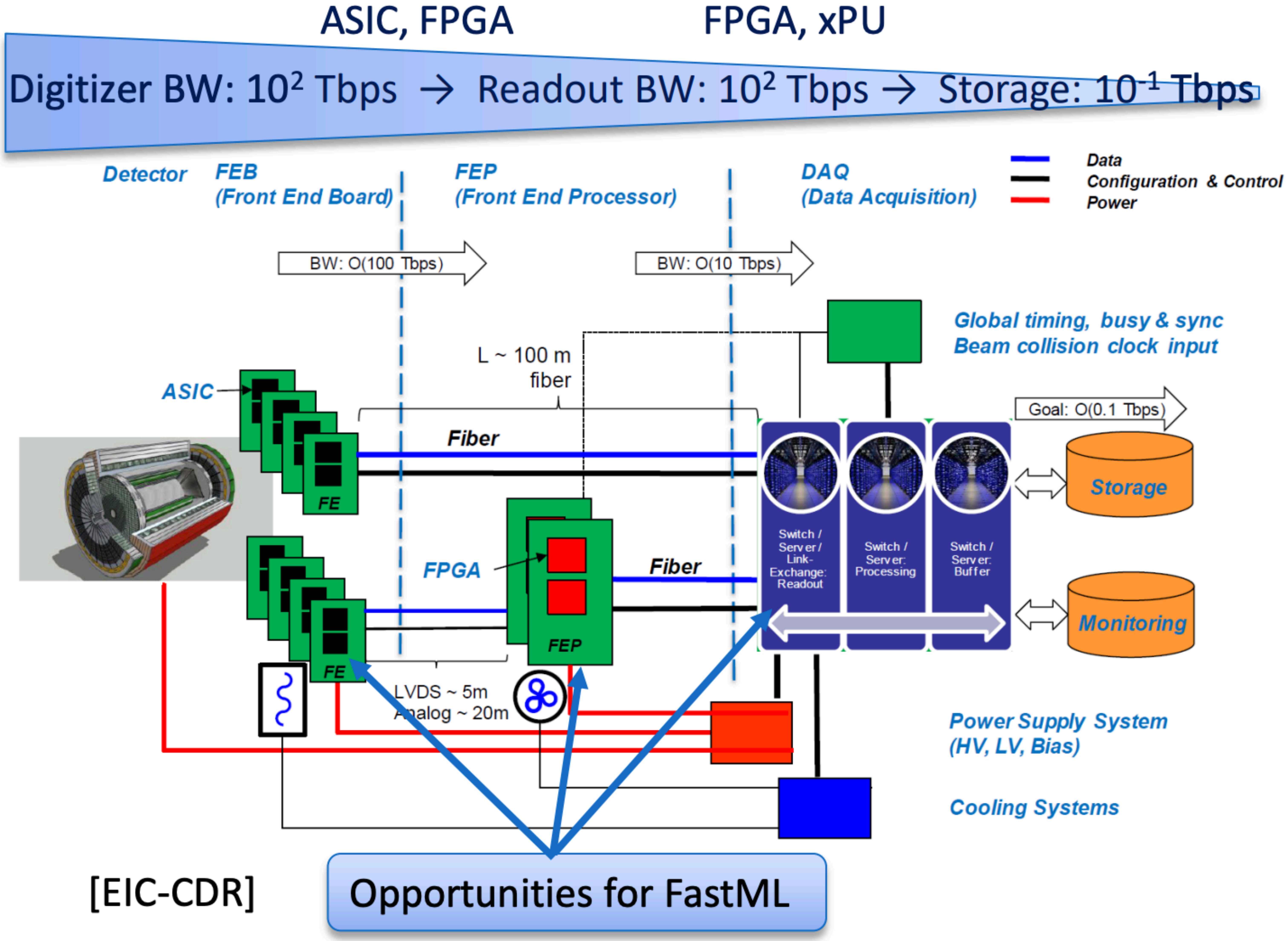




hls 4 ml

Streaming Readout

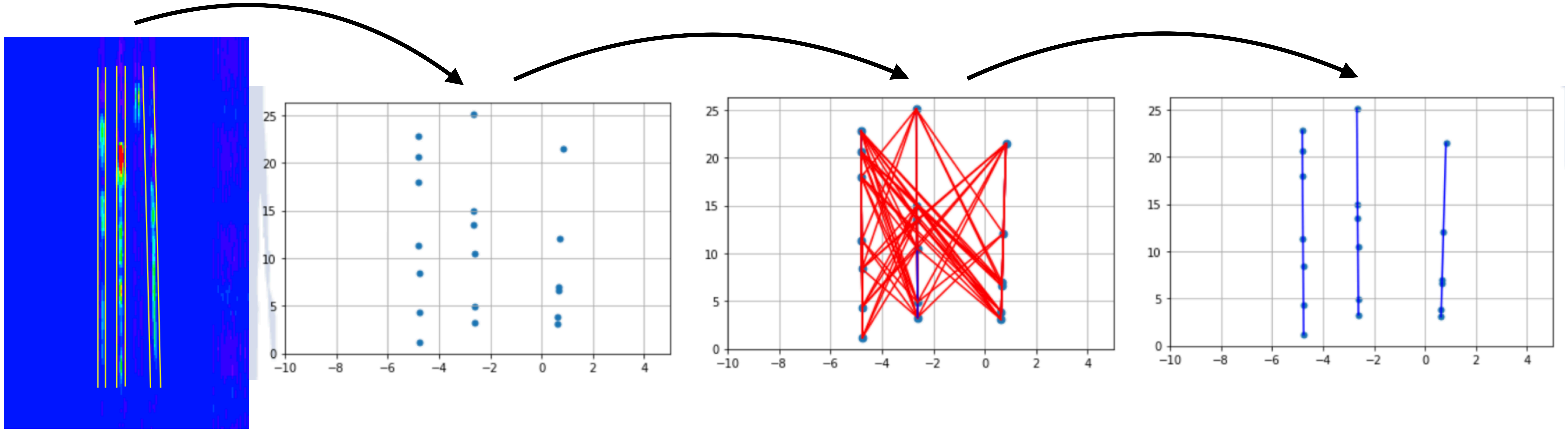
- Nuclear physics experiments beginning to achieve hundreds of Gb/s data rates (sPHENIX @ RHIC)
    - Future experiments will push past Tb/s (EIC)
- In order to reduce trigger bias and keep wide range of event topologies, streaming readout will be employed



[EIC-CDR]

Streaming Readout Example

- Tracking necessary for Gas Electron Multiplier Transition Radiation Detector (GEM TRD)
  - Critical for e/pi discrimination
- Ongoing development targeting VU9P FPGA
  - Capable of serving 21 hits and 42 edges (3-5 tracks)
  - GNN already implemented using 70% of DSPs (16 bits for weights/biases), latency of ~3 μs (200 MHz clock)
- Streaming readout makes it necessary to do all parts of reconstruction on-chip!



RNN/LSTM for track fitting

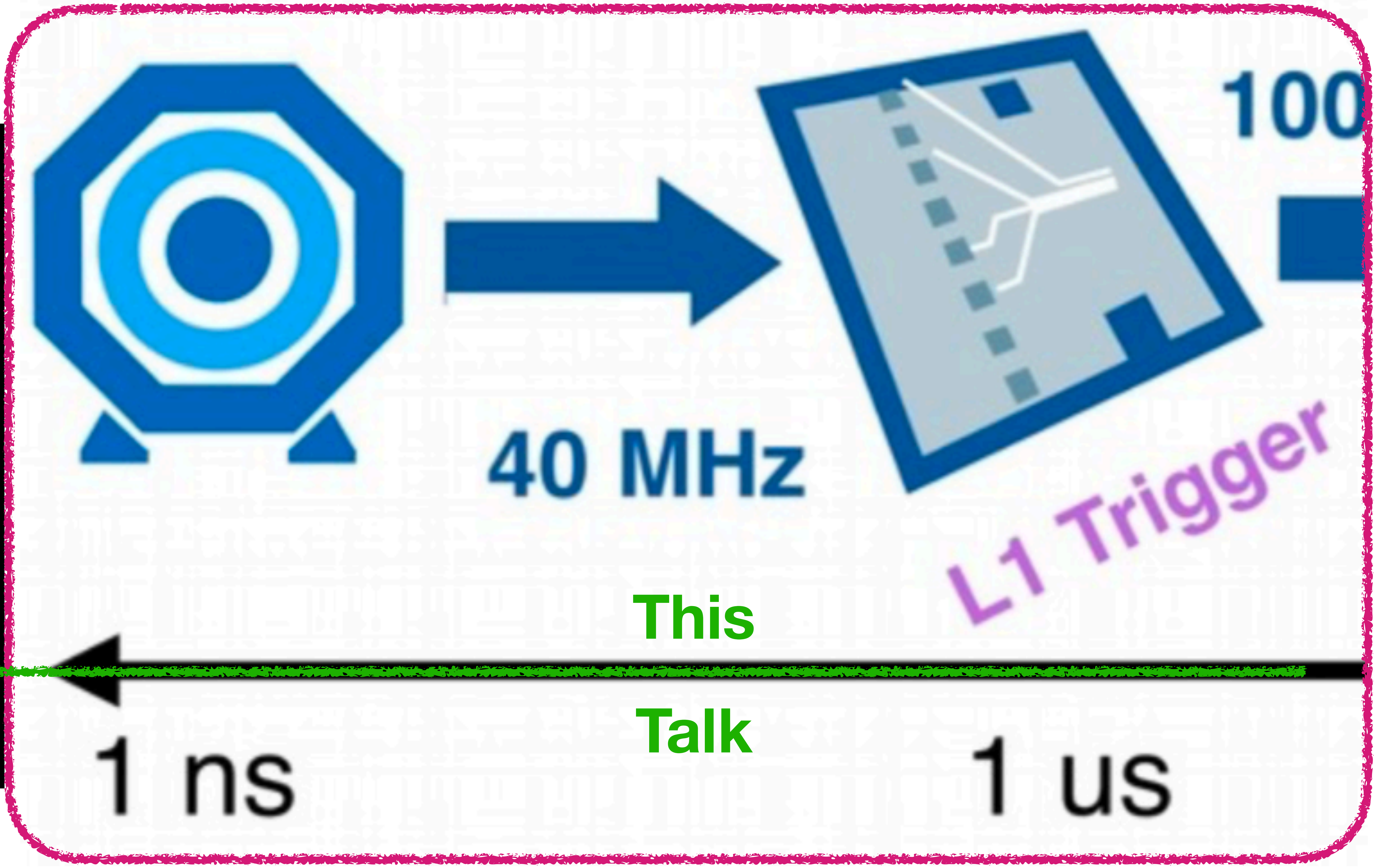
MLP for particle ID

[1] https://indico.cern.ch/event/1387540/timetable/?view=standard#44-real-time-ml-fpga-filter-fo

Readout

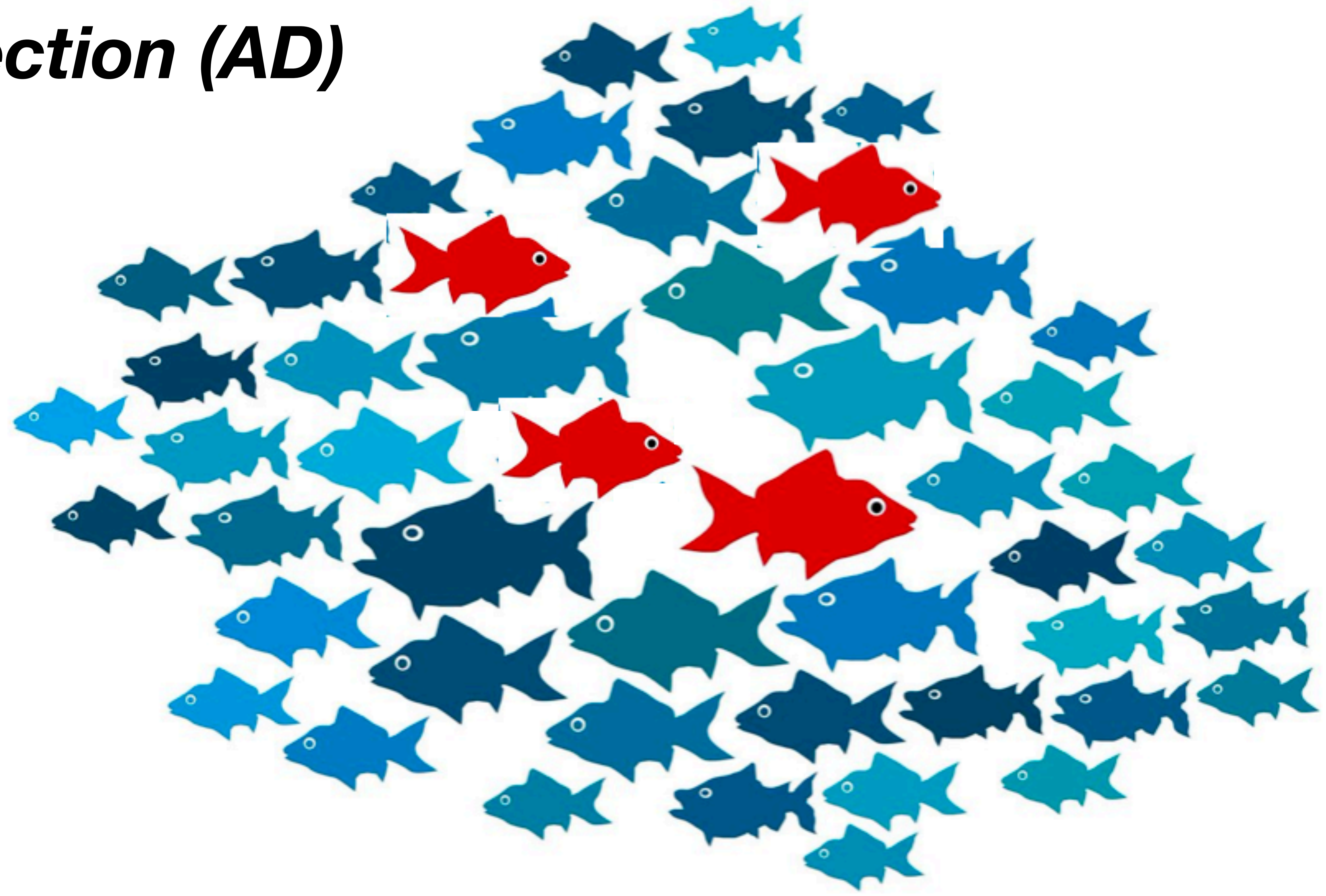
- Difficult to push ML into readout!
  - System constraints stricter
- In some cases, hardware development has already happened
  - Fewer chances for codesign
- Systems often need to be more robust to changing conditions, unforeseen circumstances
  - Critical part of deployment

Outline



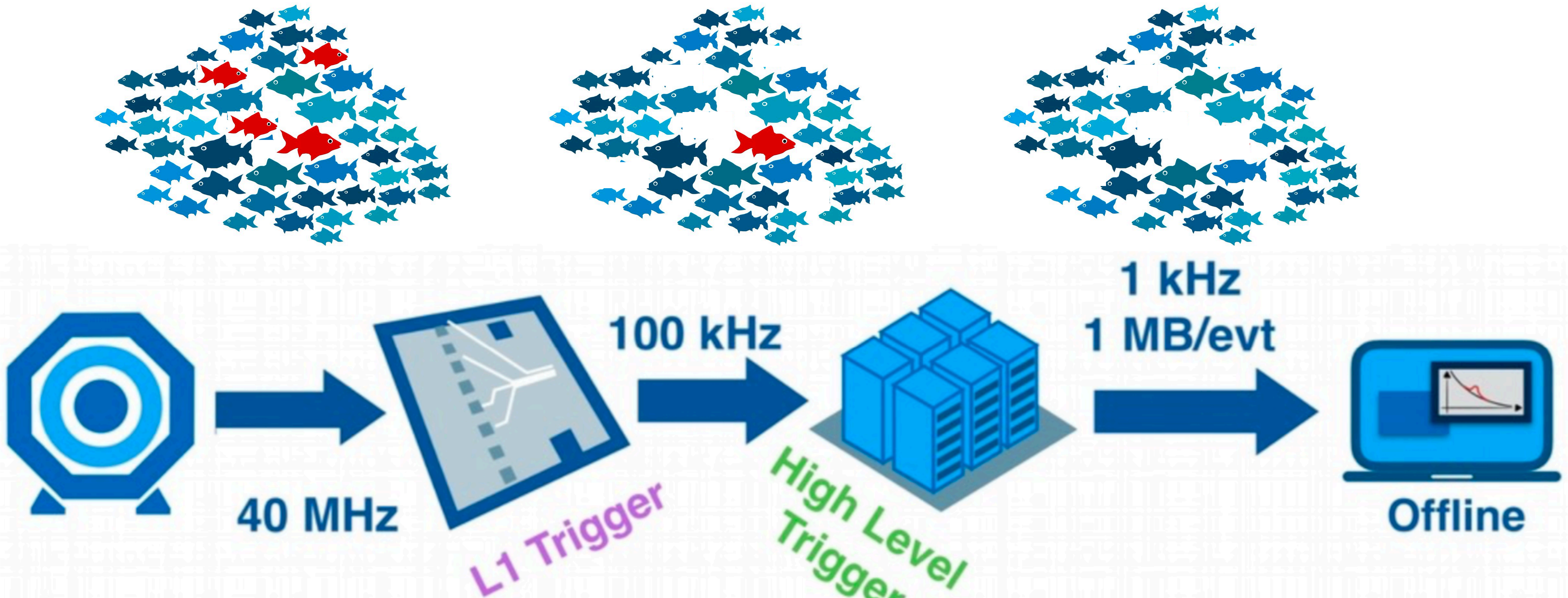
Anomaly Detection

- What if we don’t know exactly what we are looking for?
- ML offers unique solution to this challenge (no traditional alternative)
    - Broad field of ***anomaly detection (AD)***



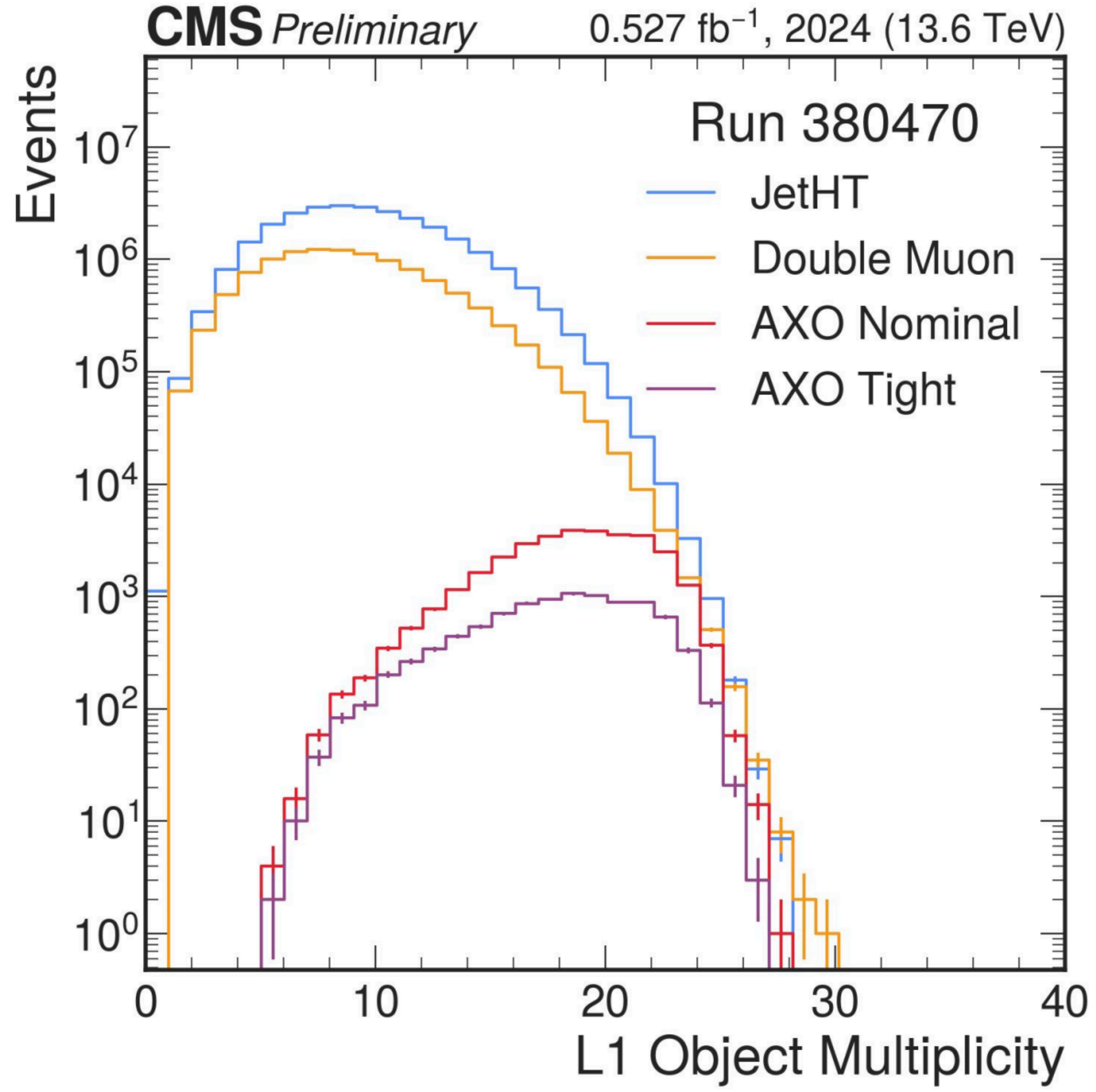
L1 Trigger AD

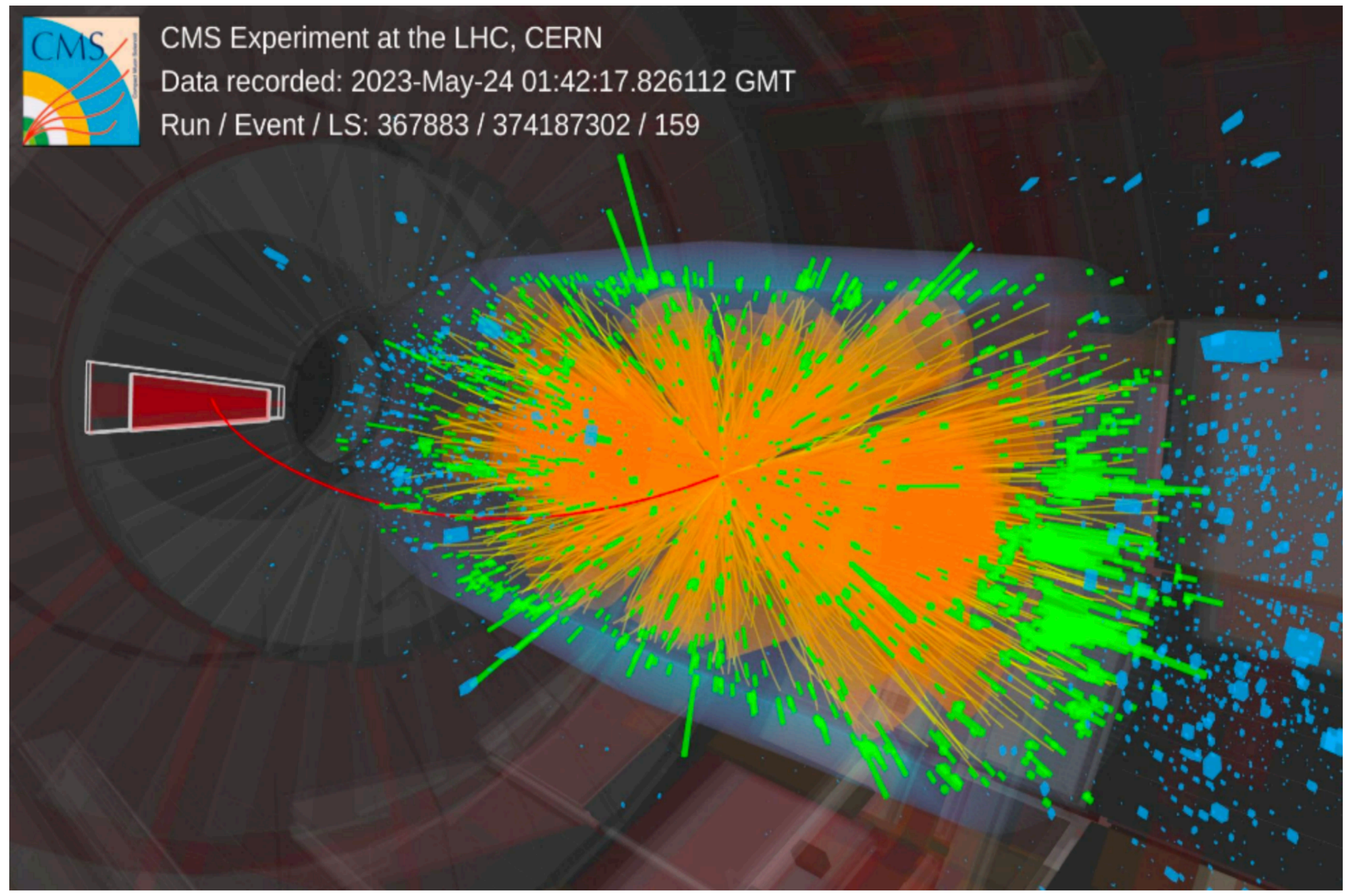
- Depending on anomaly, we could have none left in recorded data
- Low-latency ML is the only option! (eg. autoencoders)

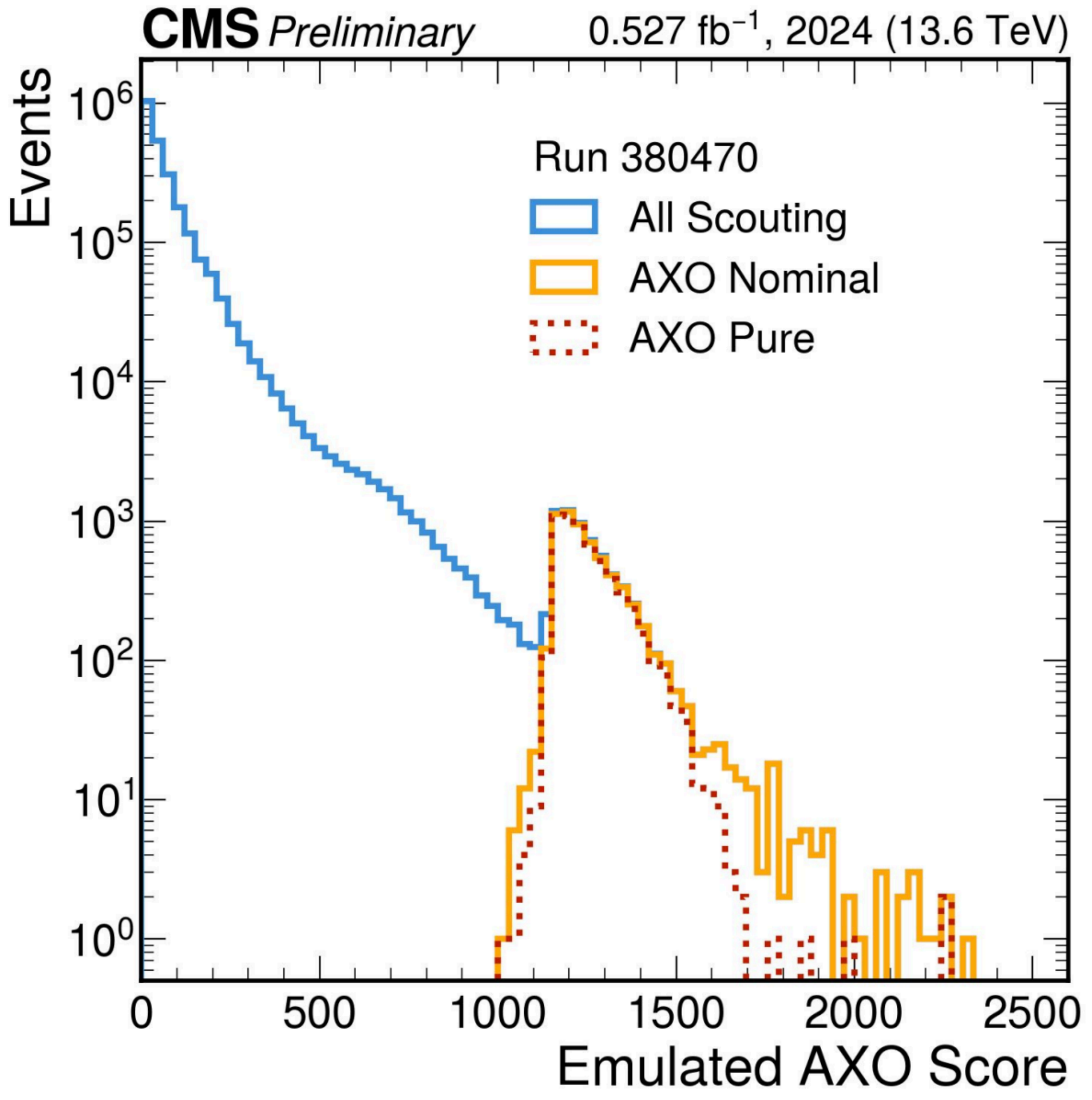


L1 Trigger AD

- CMS has already deployed multiple AD algorithms in trigger
  - AXOL1TL [CMS DP-2023/079, CMS DP-2024/059] & CICADA [CMS DP-2023/086]
- Currently collecting interesting events that would have been missed
  - Network preferentially identifies large multiplicity events, potentially large gains in new physics acceptance
- First AD-based trigger deployed in ATLAS as well, results to come soon!
  - Other ATLAS AD triggers in development as well

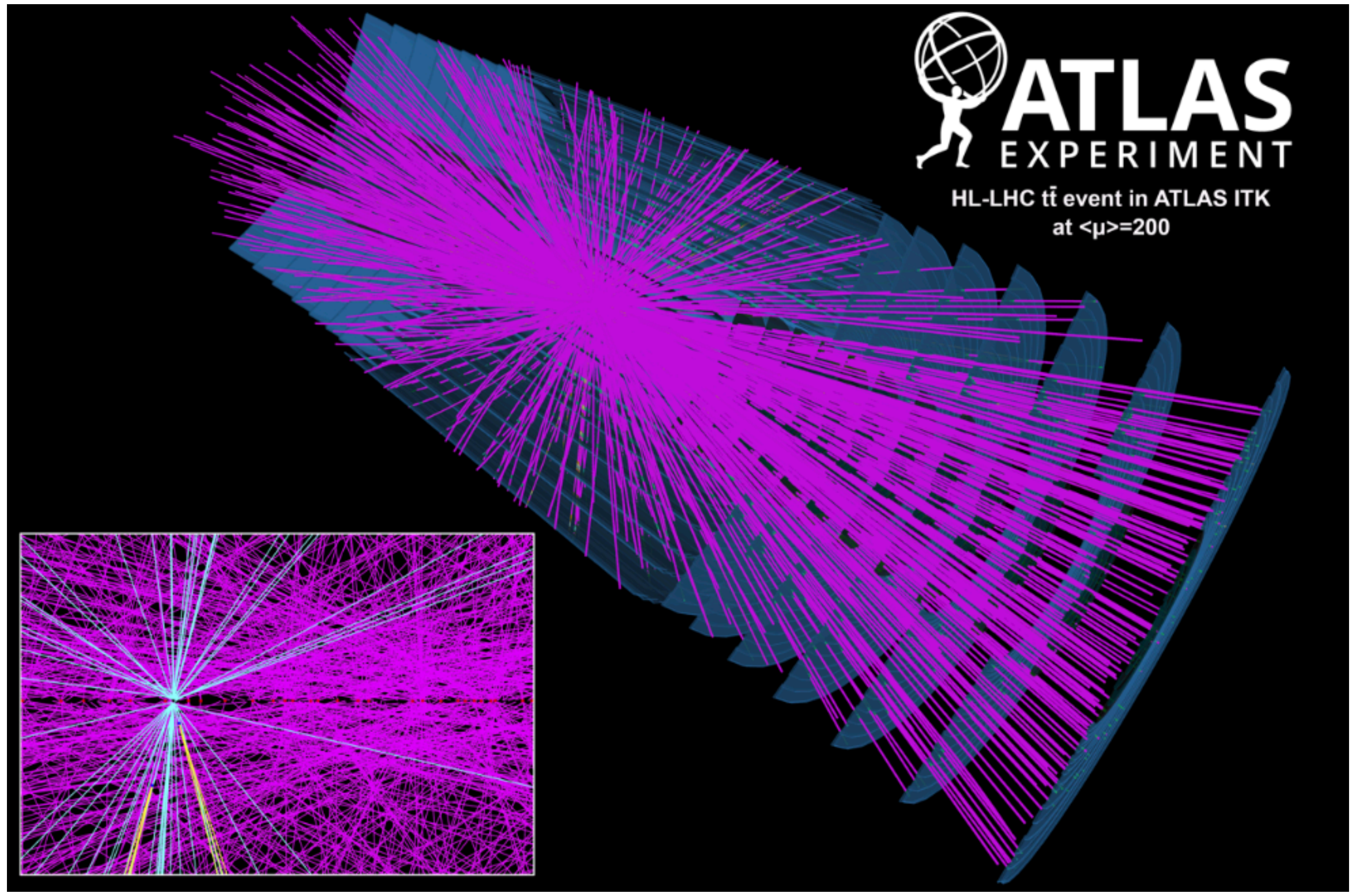


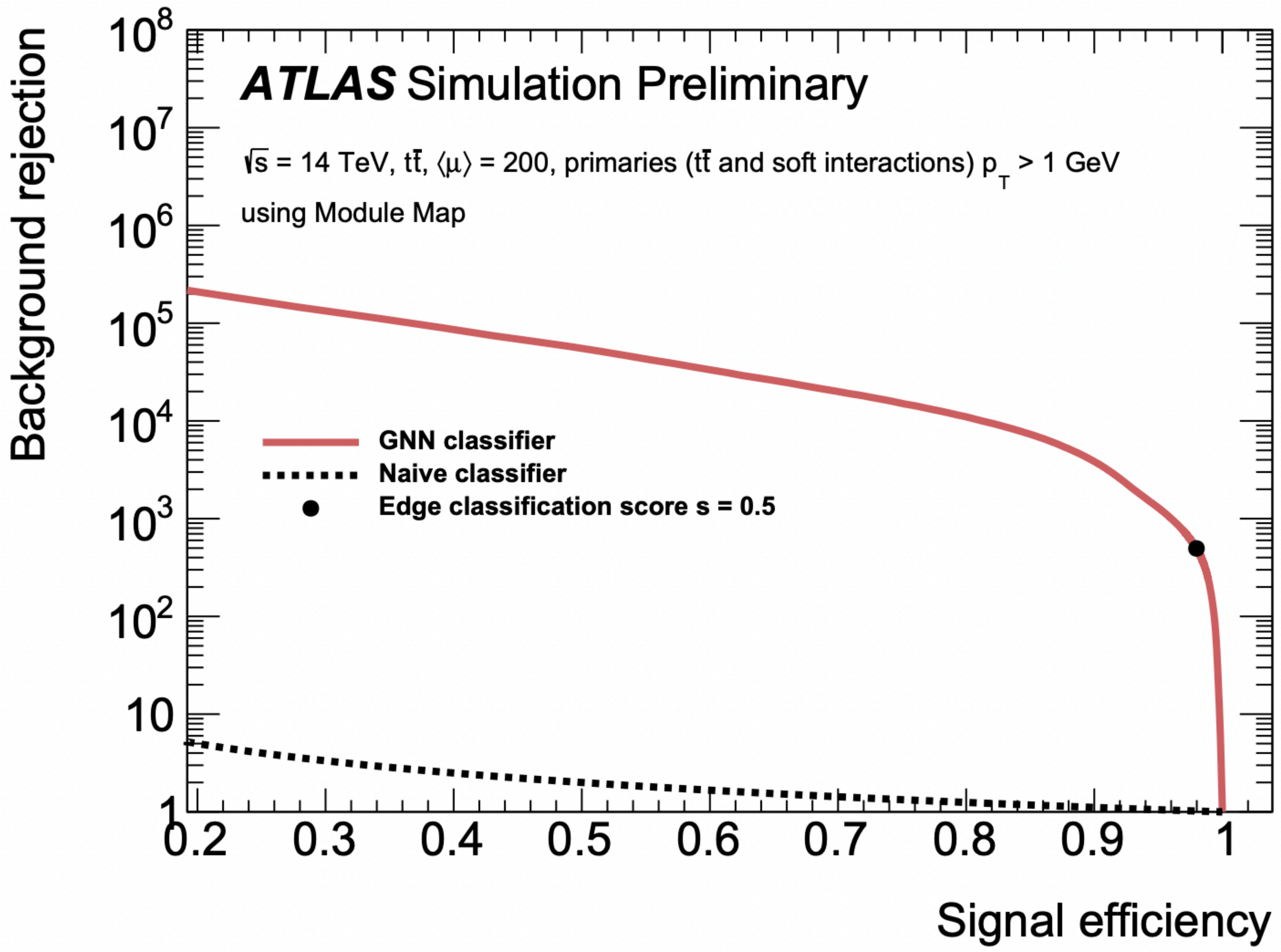


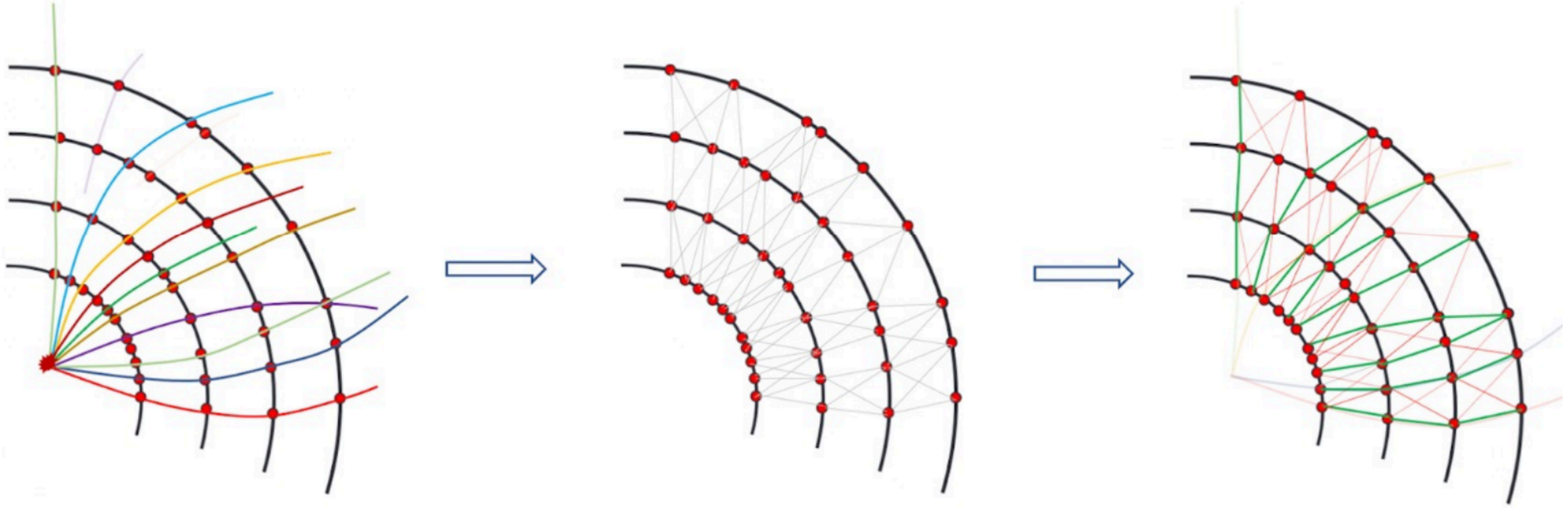


GNN Tracking

- Tracking is an incredibly hard problem, tracking in HLT even harder
  - Huge combinatorics, only going to get worse
- GNNs show promise for HL-LHC
  - ~2.7 x 10^5 nodes, ~1.3 x 10^6 edges



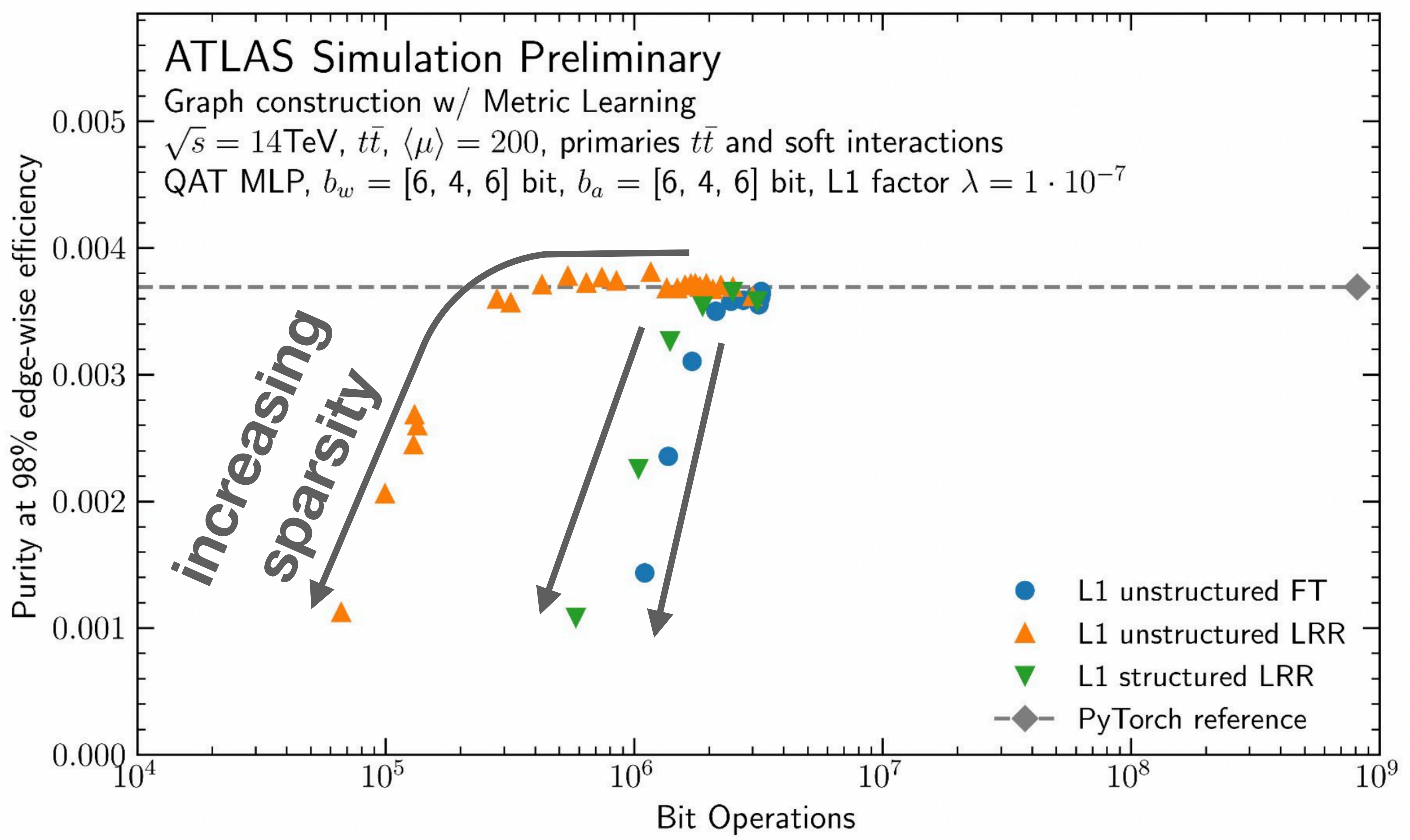




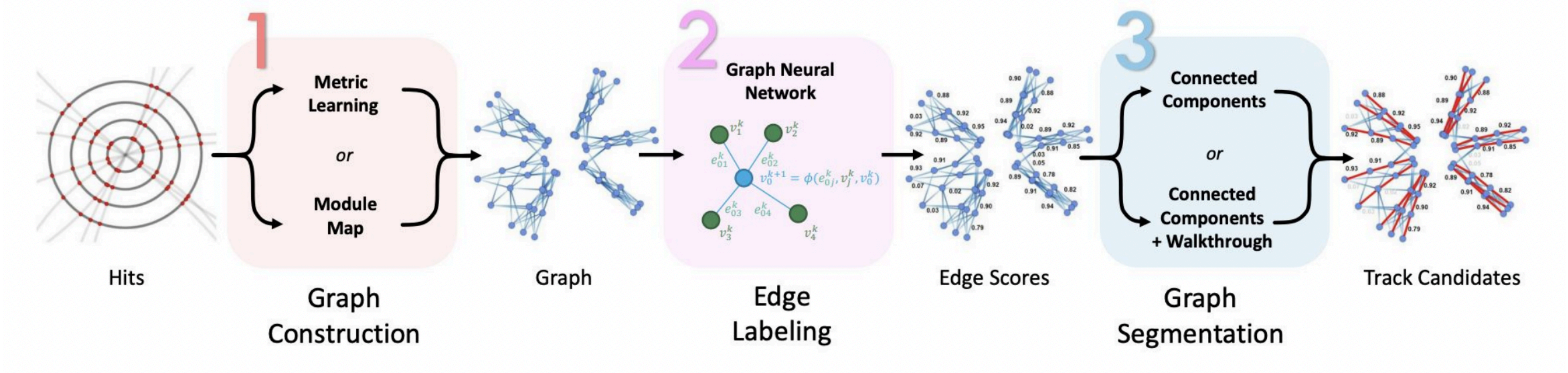
ATL-ITK-PROC-2022-006

GNN Tracking

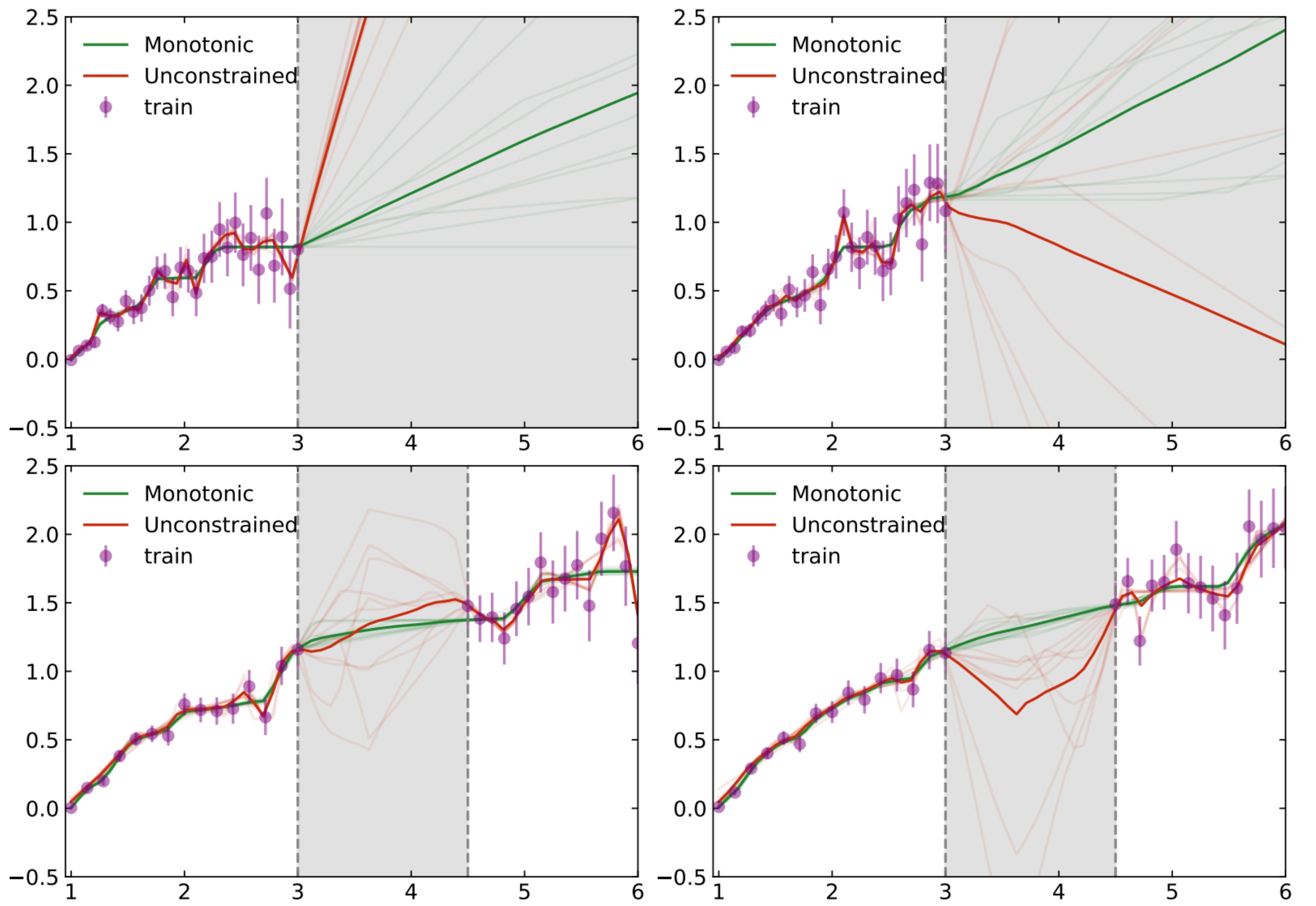
- Pipeline from raw hits to track candidates involves multiple steps
- Complicated workflow, large networks
- Pruning one potential option for reducing size, still need to run quickly in trigger
- As-a-service is a promising option



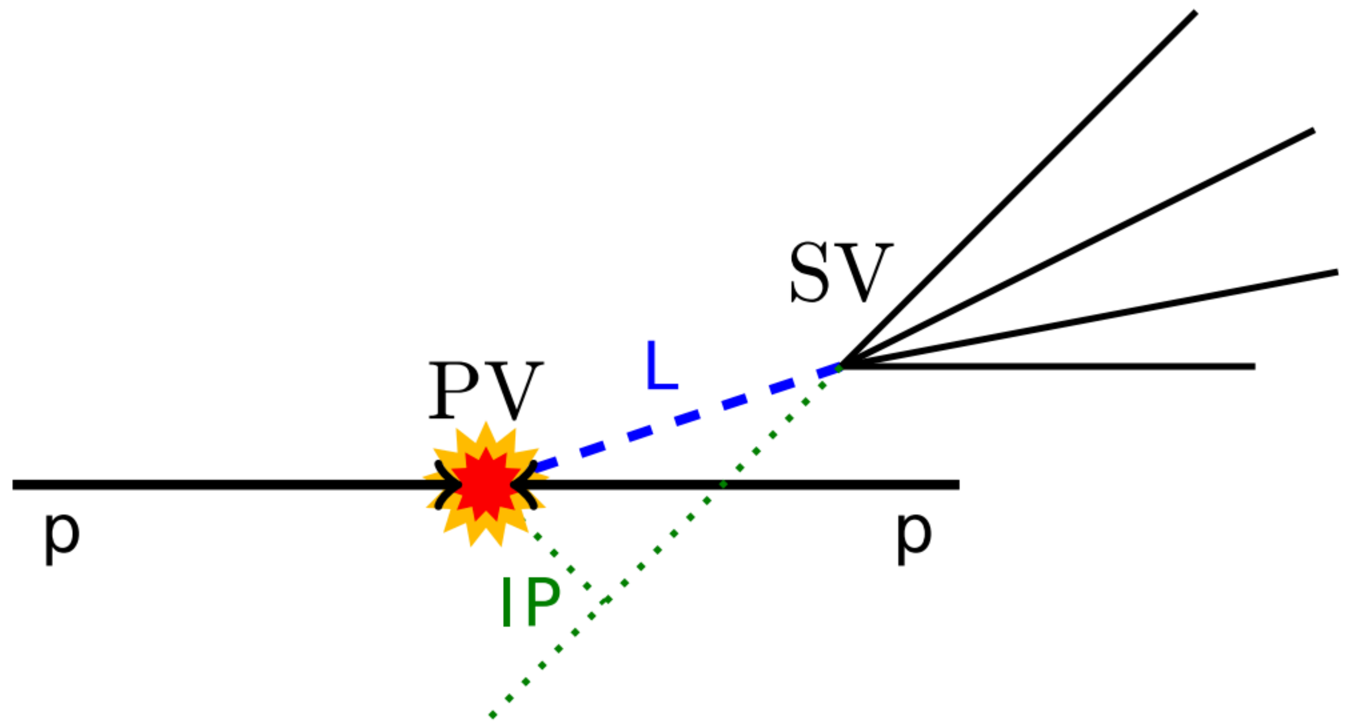
ATL-COM-DAQ-2024-004

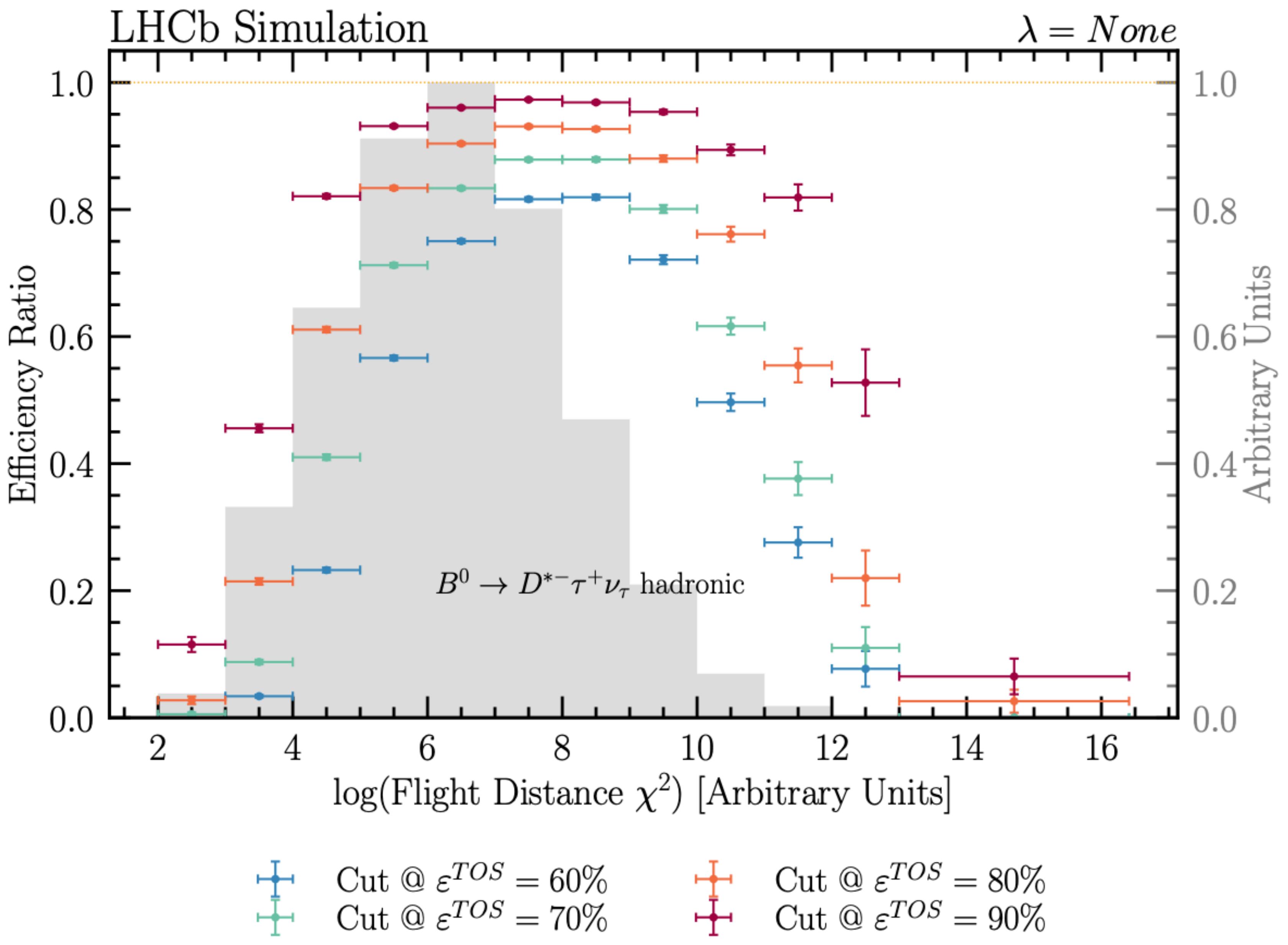


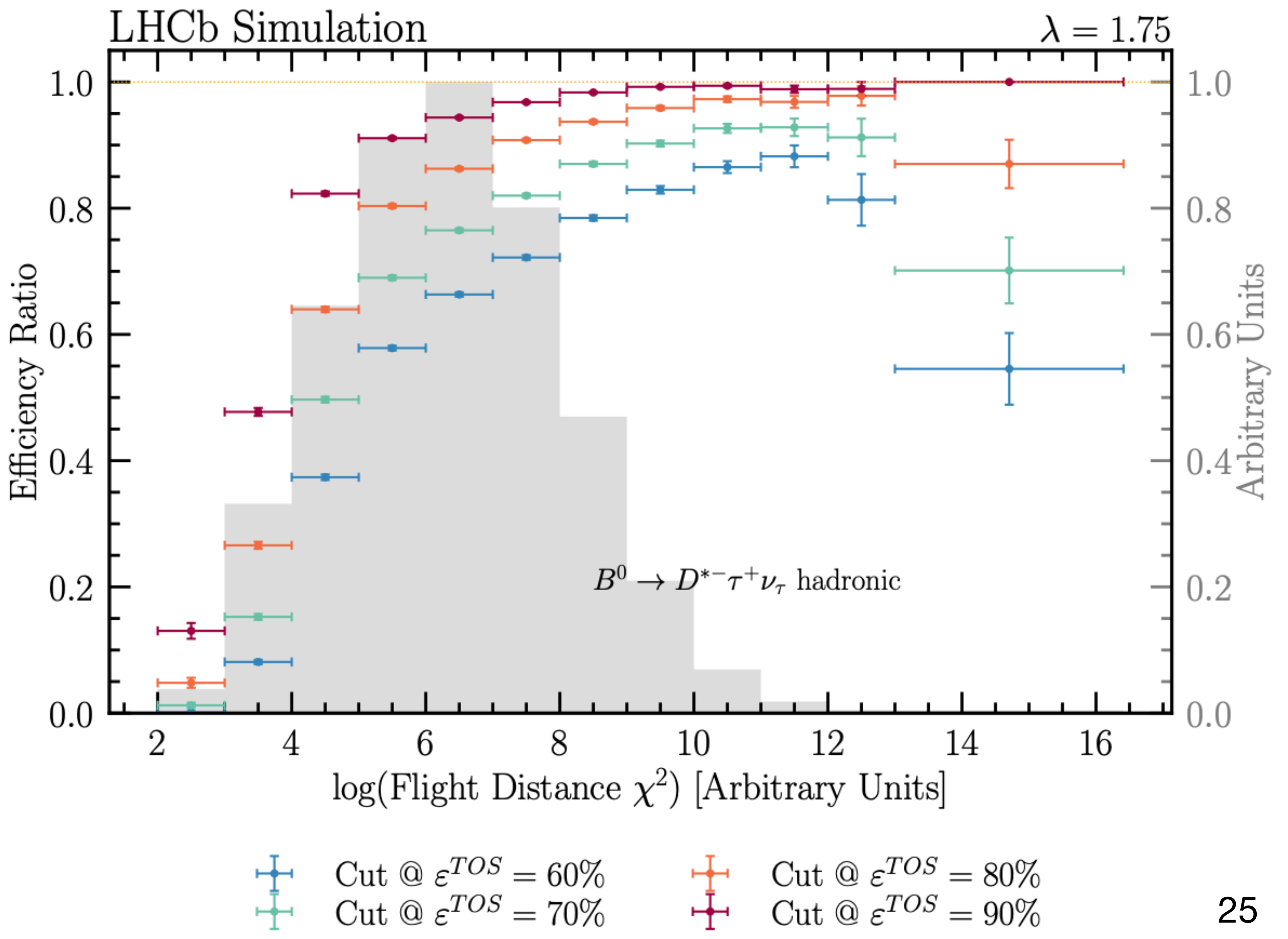
Lipschitz Monotonic NN



- On-detector ML is not just about speed
  - Robustness and understandability are also very important
- Networks can be made provably monotonic [2112.00038]
- LHCb has used this technique to design NNs for use in HLT
  - Eg. smooth dependence on flight distance for heavy flavor decays
- Improved stability



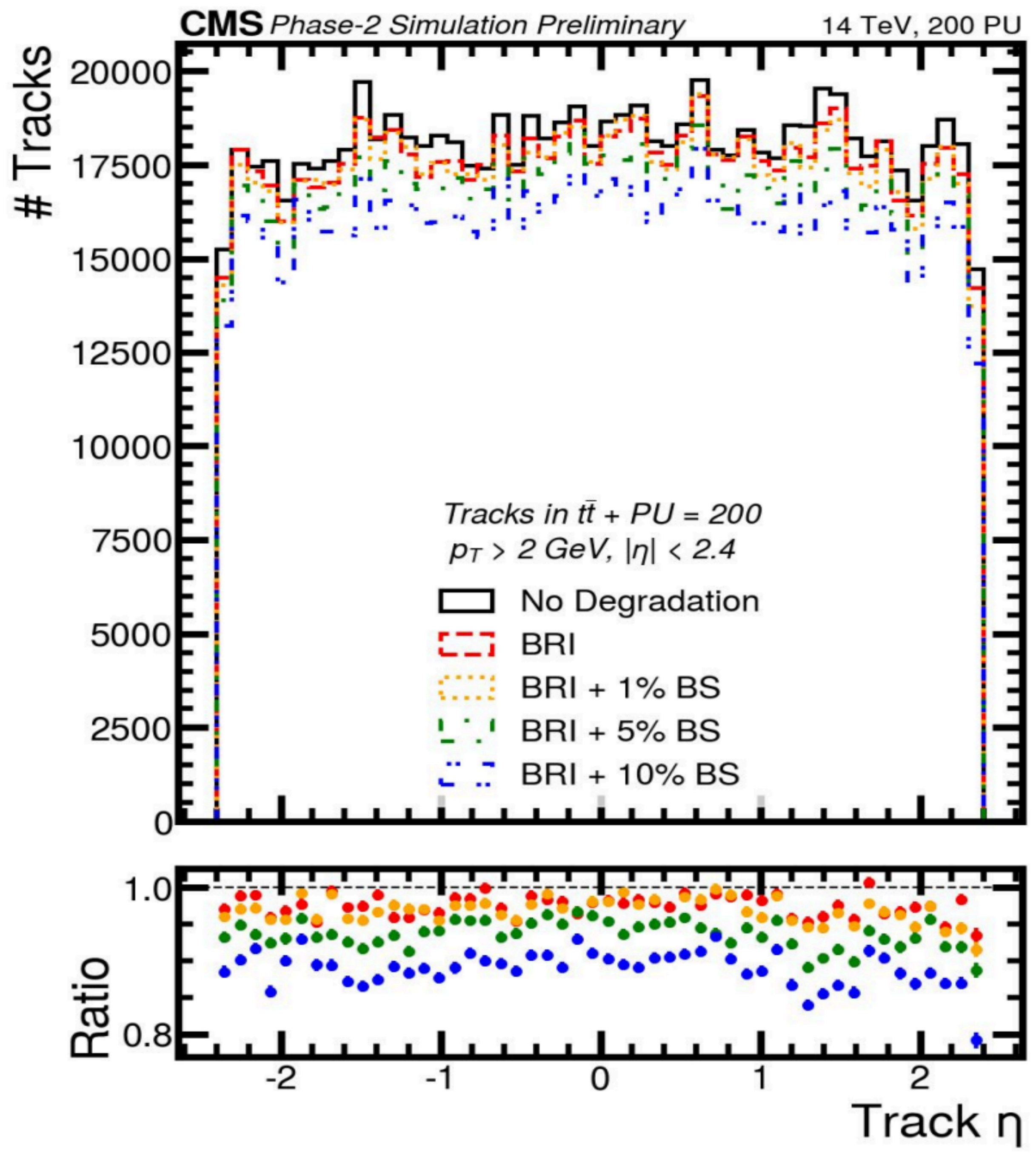




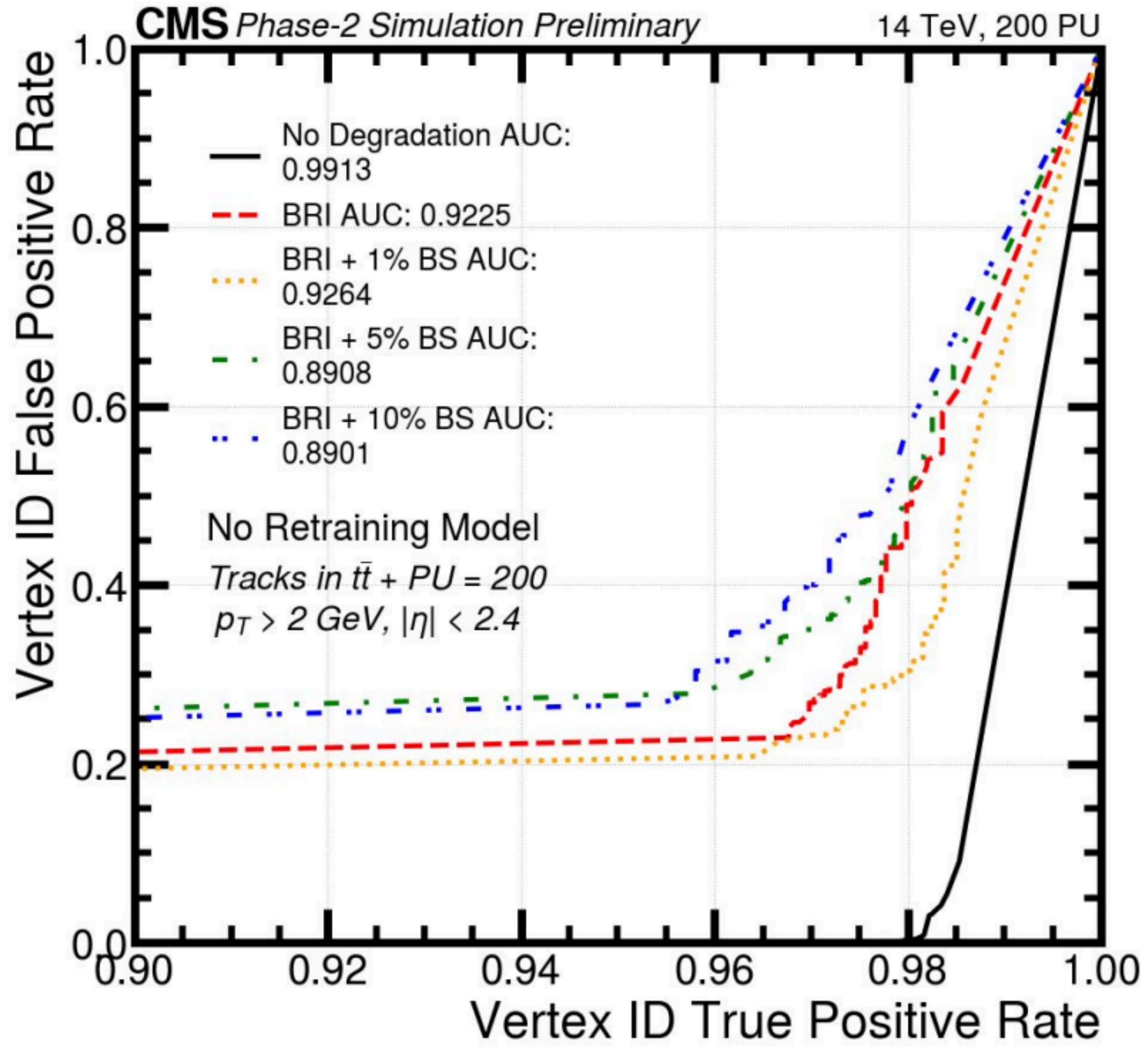
2306.09873 , 2312.14265

Continual Learning

- On-detector ML has no re-do button
  - Cannot just reprocess with new network if conditions change
- Continual learning method uses mix of original and new data to retrain model
  - Better performance than simple retraining (or no retraining)
- Important consideration especially when conditions can change significantly
- Example from CMS considers degradations in L1 tracking



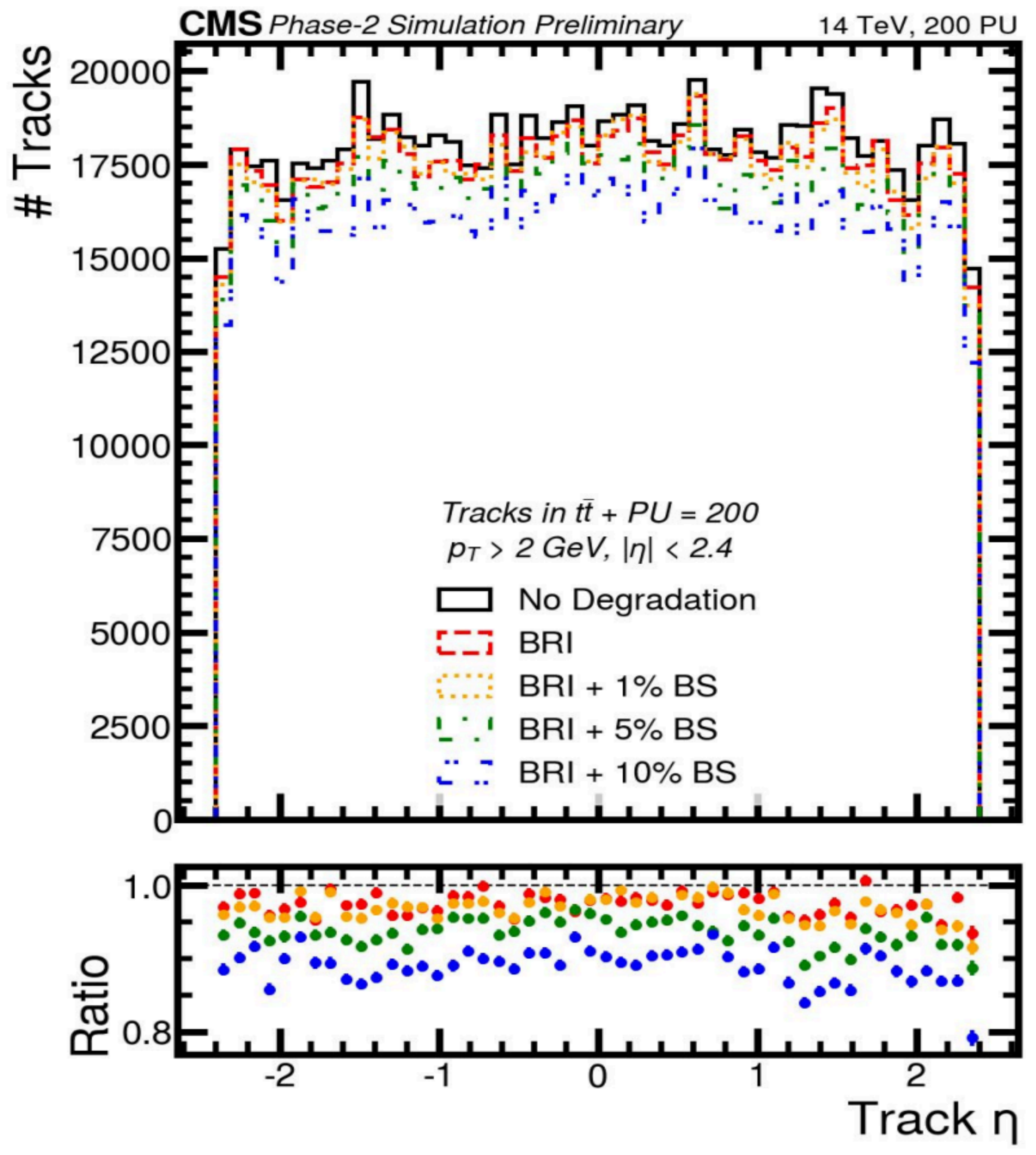
No Retraining Model



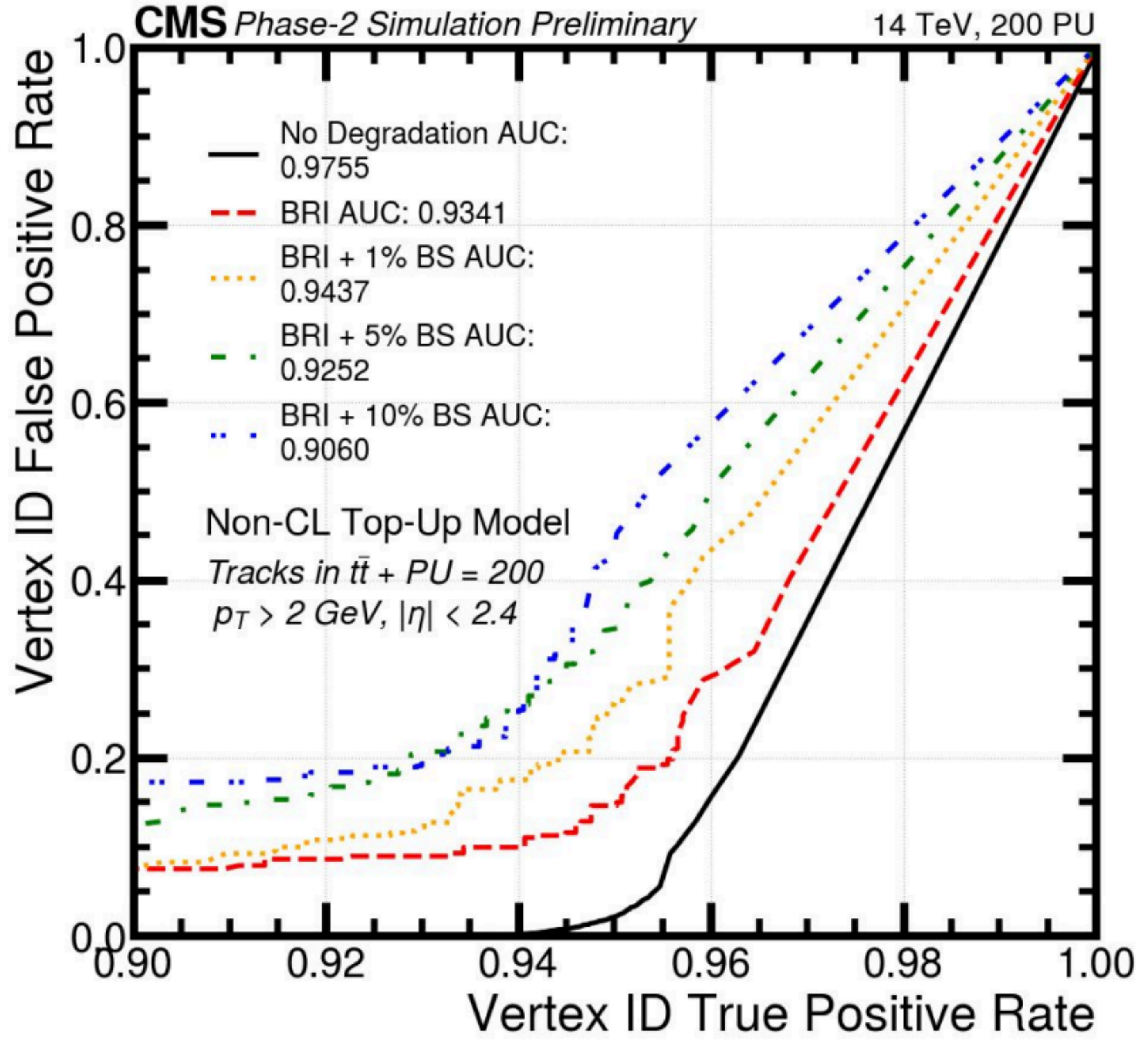
CMS DP-2023/022

Continual Learning

- On-detector ML has no re-do button
  - Cannot just reprocess with new network if conditions change
- Continual learning method uses mix of original and new data to retrain model
  - Better performance than simple retraining (or no retraining)
- Important consideration especially when conditions can change significantly
- Example from CMS considers degradations in L1 tracking



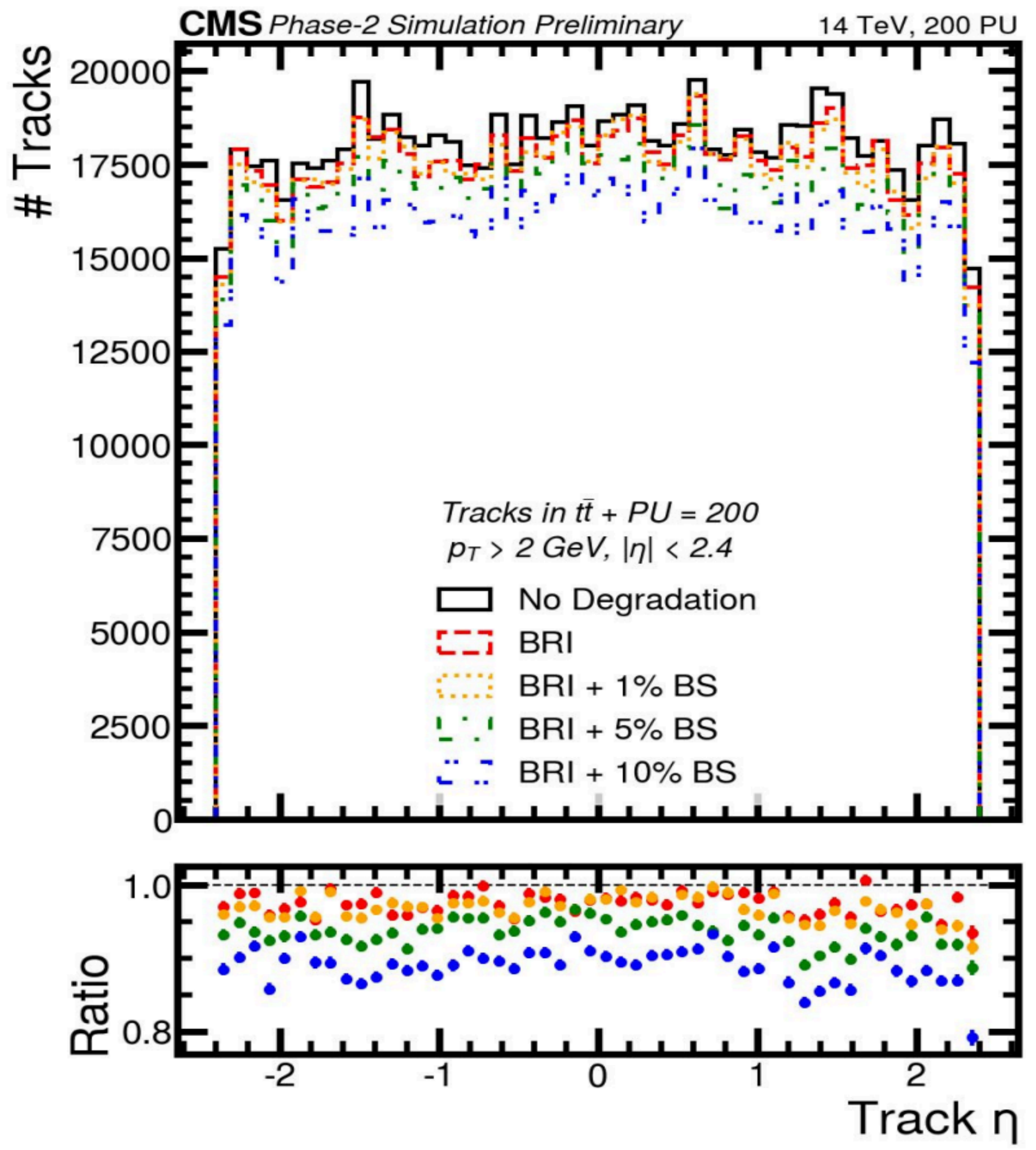
Non-CL Top-Up Model



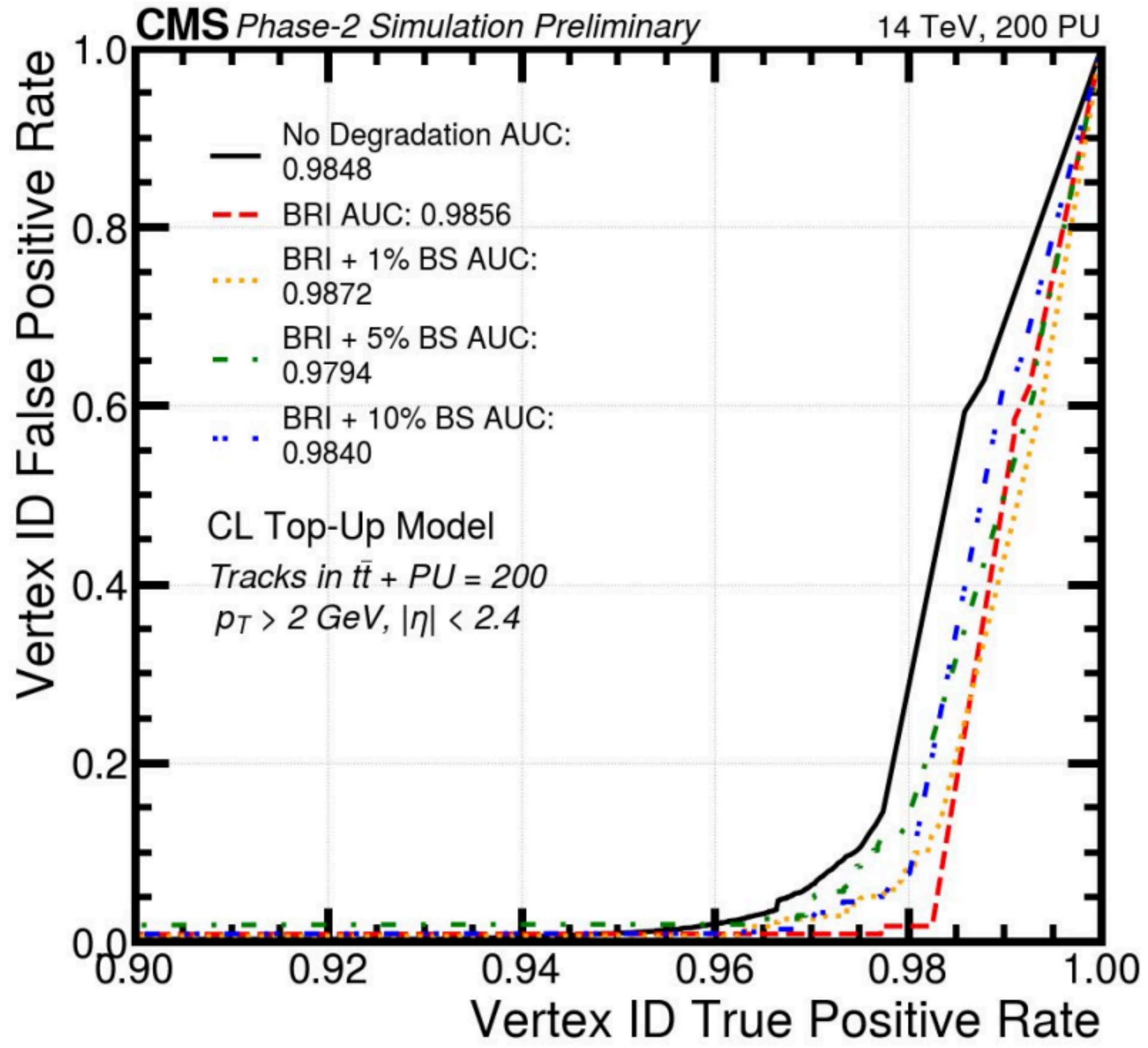
CMS DP-2023/022

Continual Learning

- On-detector ML has no re-do button
    - Cannot just reprocess with new network if conditions change
- Continual learning method uses mix of original and new data to retrain model
    - Better performance than simple retraining (or no retraining)
- Important consideration especially when conditions can change significantly
- Example from CMS considers degradations in L1 tracking



CL Top-Up Model



CMS DP-2023/022

Future Opportunities & Challenges

- We do not drive Xilinx product development (although they do pay attention to us)
  - Can we make use of new advances like AI engines? Can we learn from them?
- Streaming readout?
  - Lots to learn from LHCb [1], EIC
  - AI/ML in networking?
- Algorithm and hardware development should be considered simultaneously (codesign)
  - More difficult the closer we go to detectors, but vital for maximizing performance

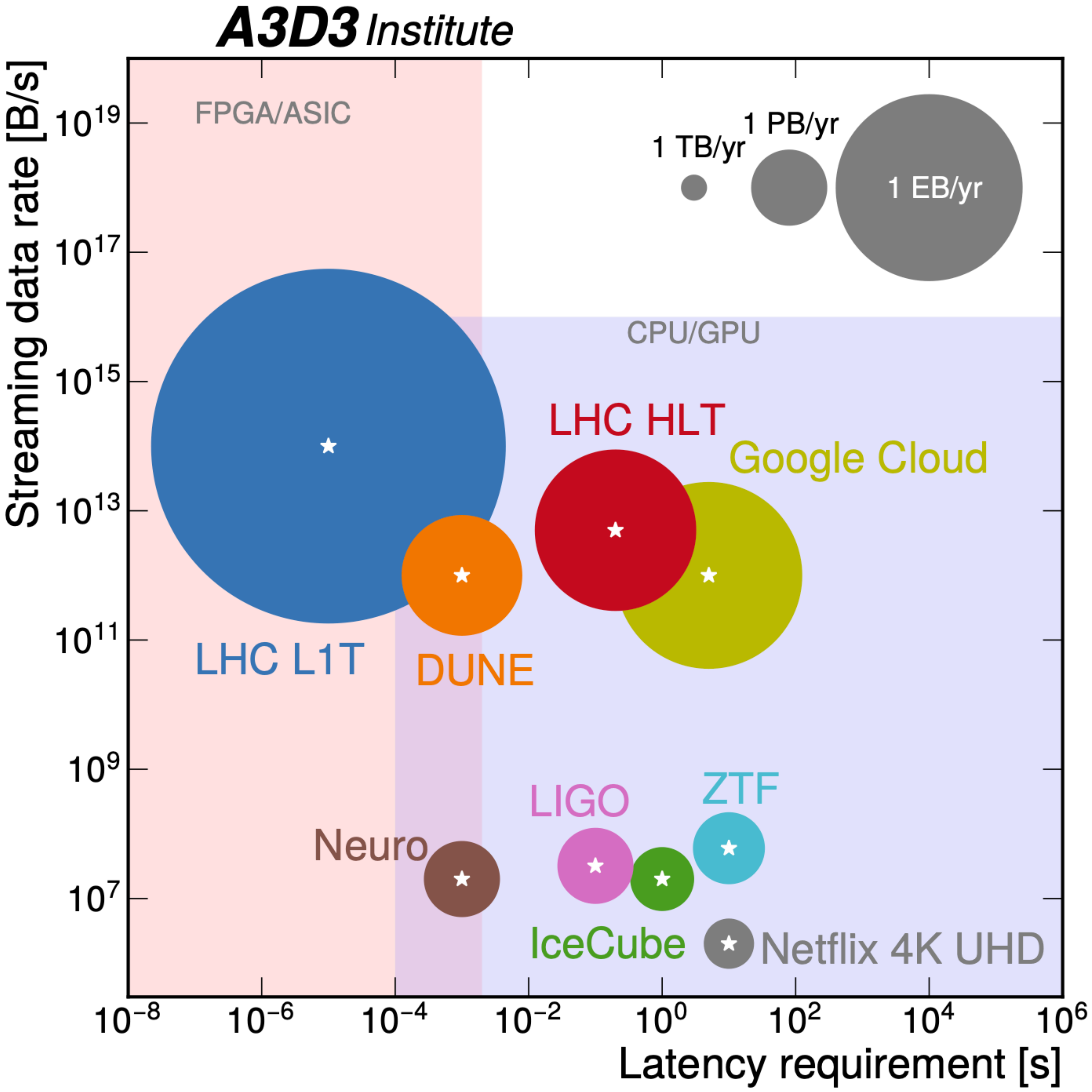
Codesign analogy stolen shamelessly from Ryan

VS

[1] https://indico.cern.ch/event/1387540/contributions/6153414/attachments/2948376/5181940/FMLSC2024%20(2).pdf

Conclusions

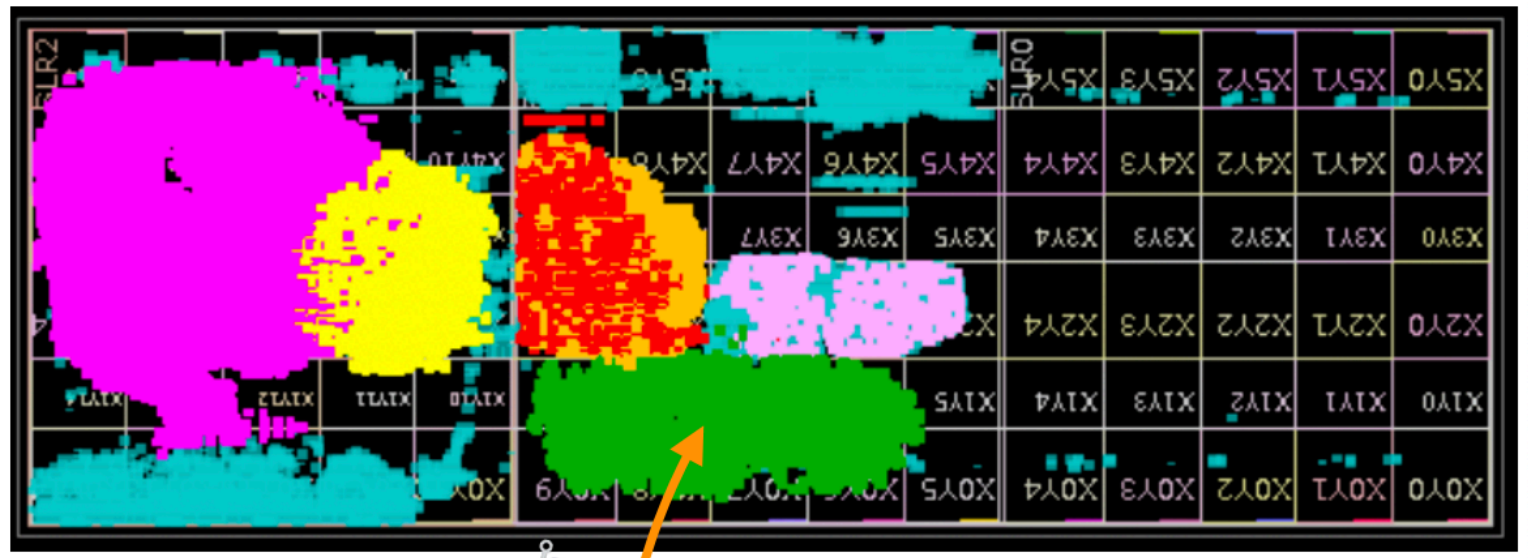
- Advancing ML on-detector can help contribute to maximizing physics of our experiments
- Many challenges
  - Constraints, stability, implementation (along with all the usual ML challenges!)
- These challenges may differ but many appear in other fields, areas too
  - LHCb, EIC, accelerators, Belle-II, DUNE, …
- Exciting times!

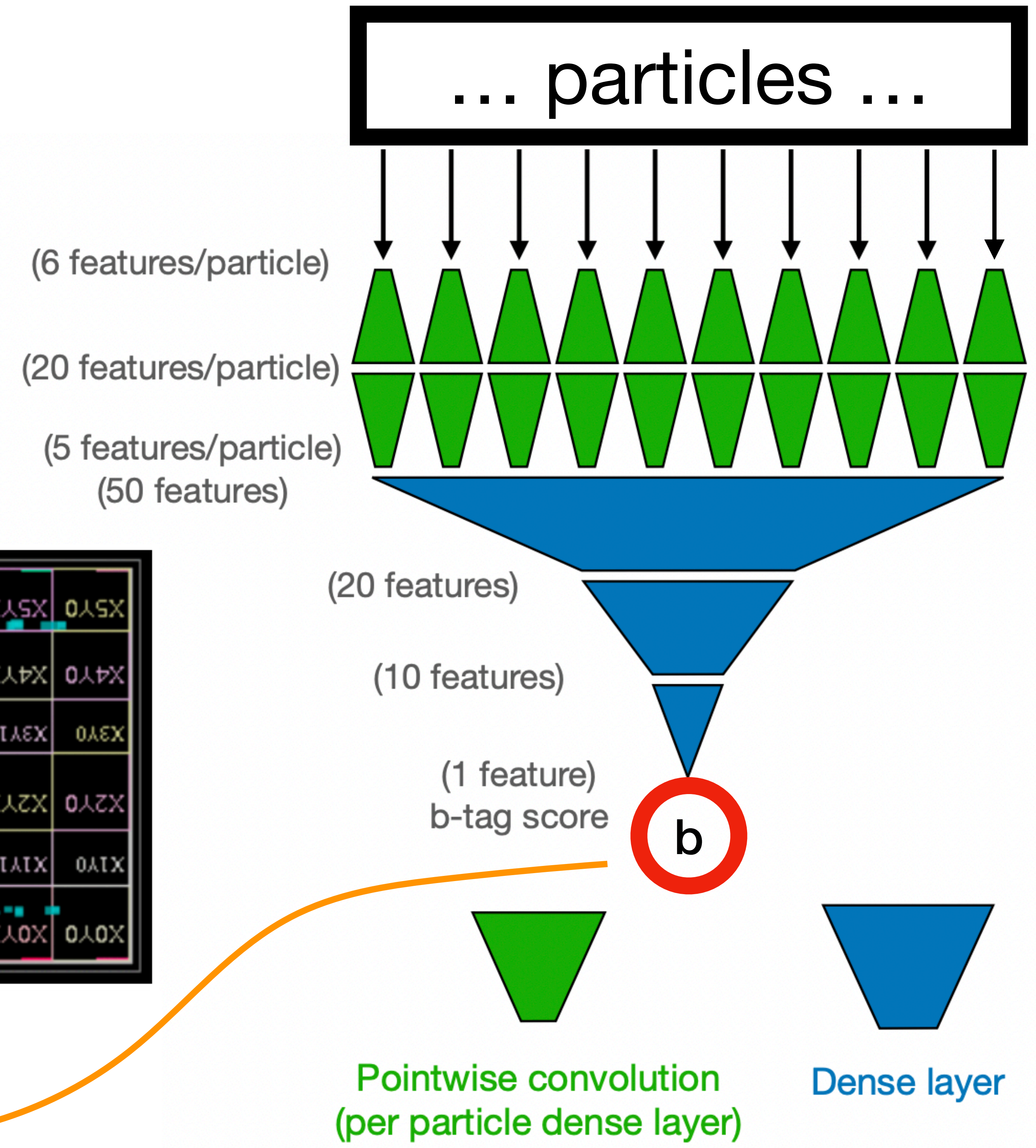


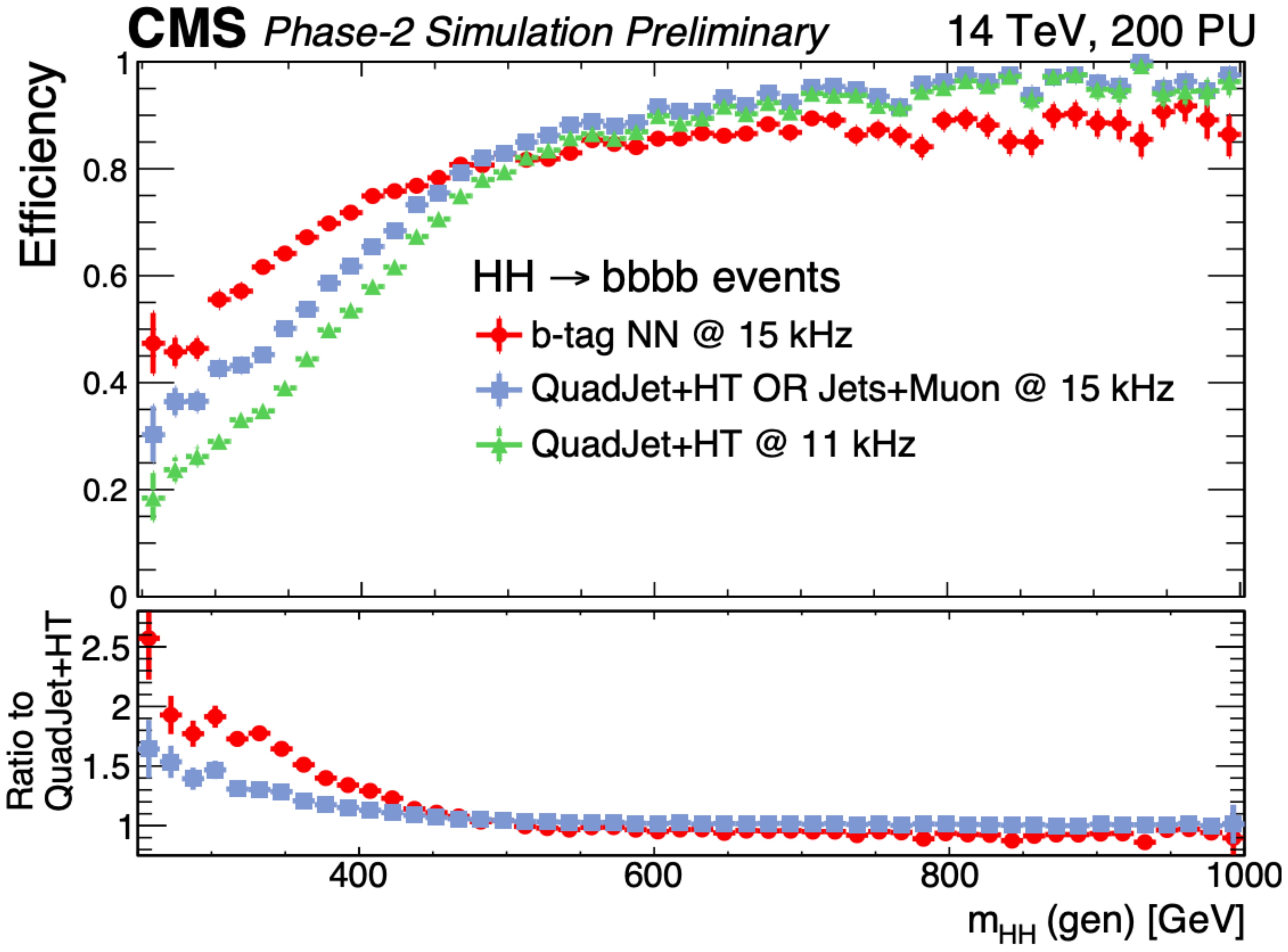
BACKUP

L1 b-quark Identification

- NN trained to identify b-quarks using collection of particles
- Architecture includes featurizers that act on each particle individual
- **Significantly improved acceptance for HH→bbbb events with low m_{HH}** (compared to traditional cut-based methods)



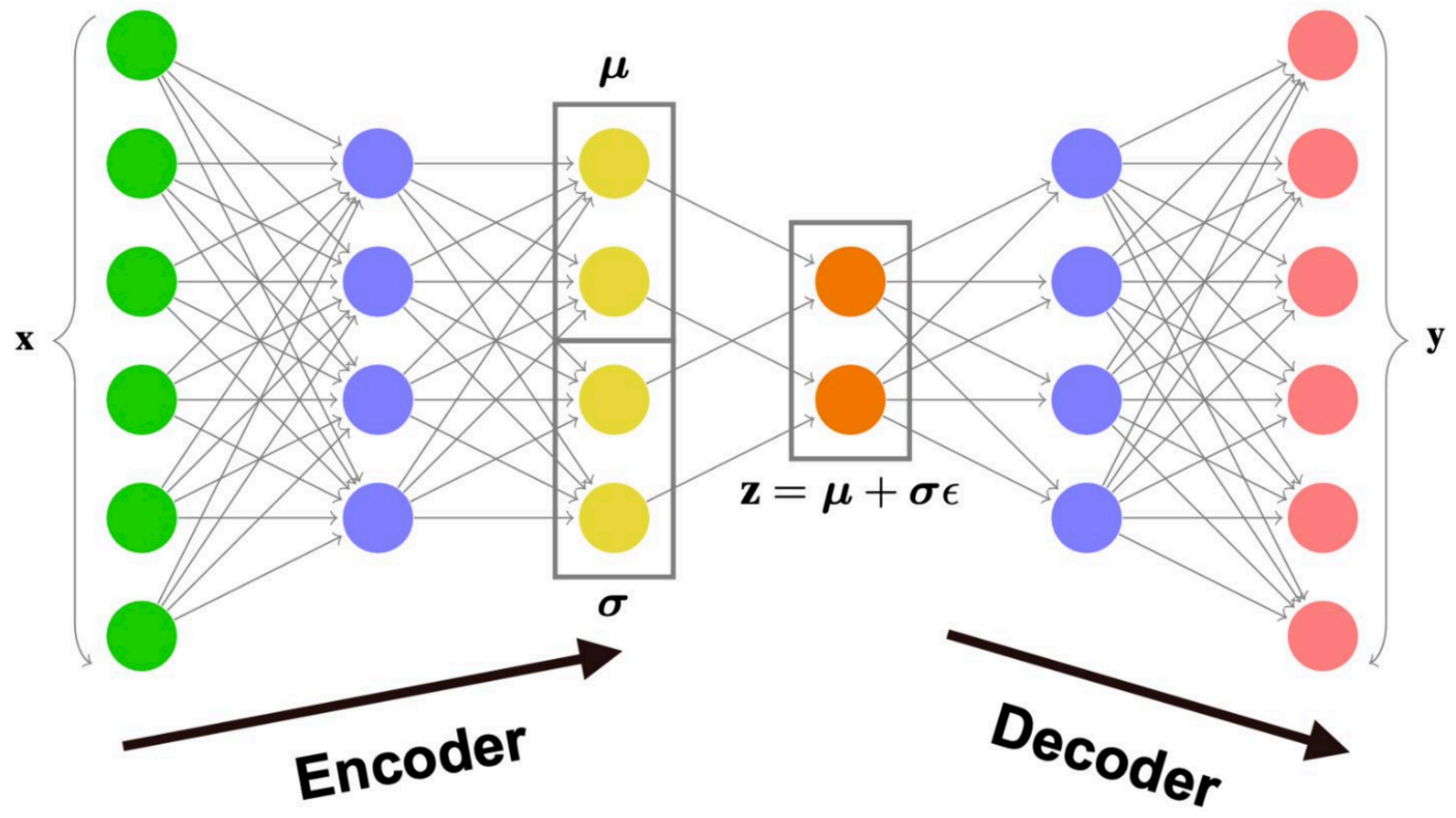


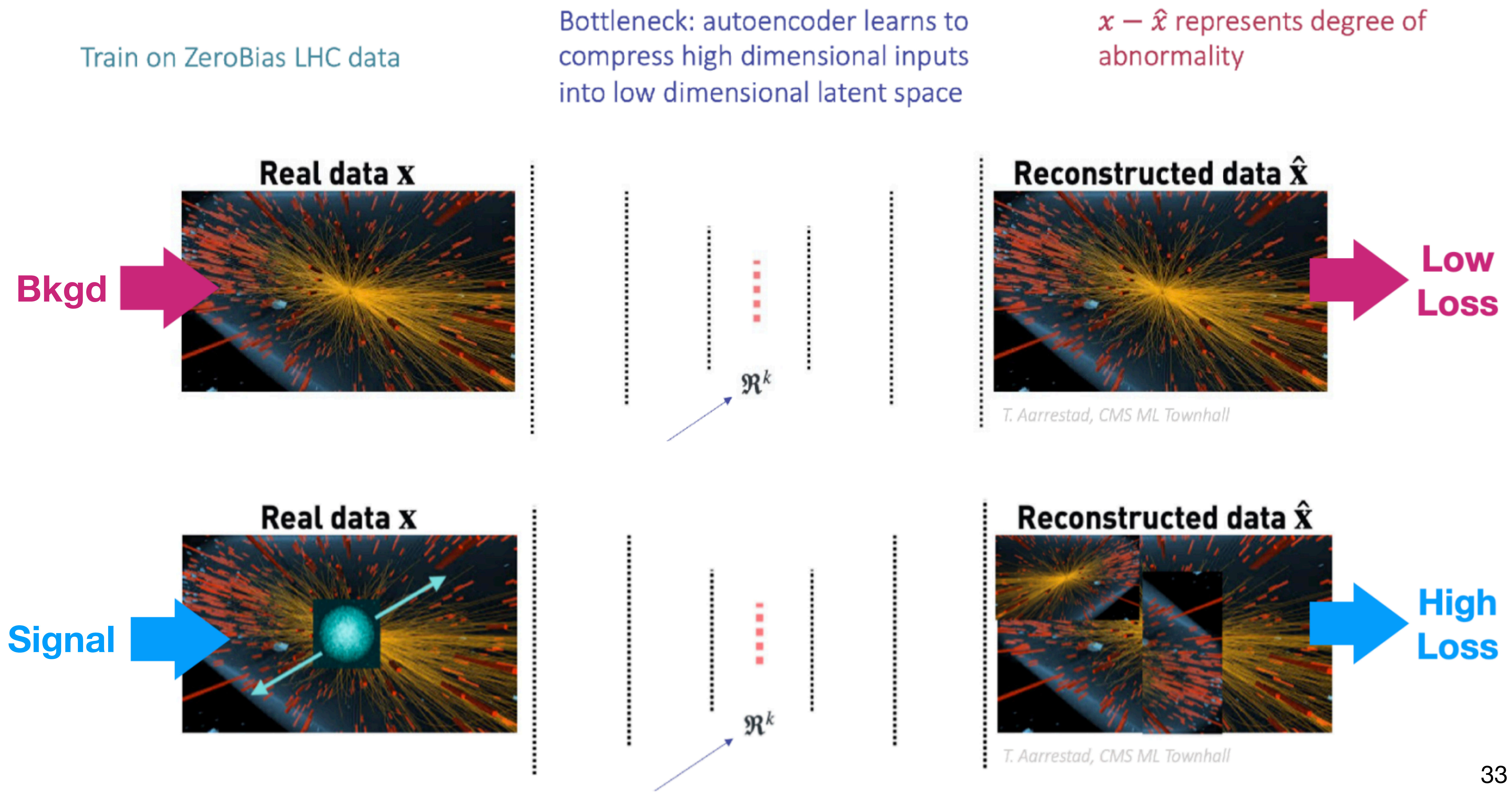


CMS DP-2022/021

L1 Trigger AD

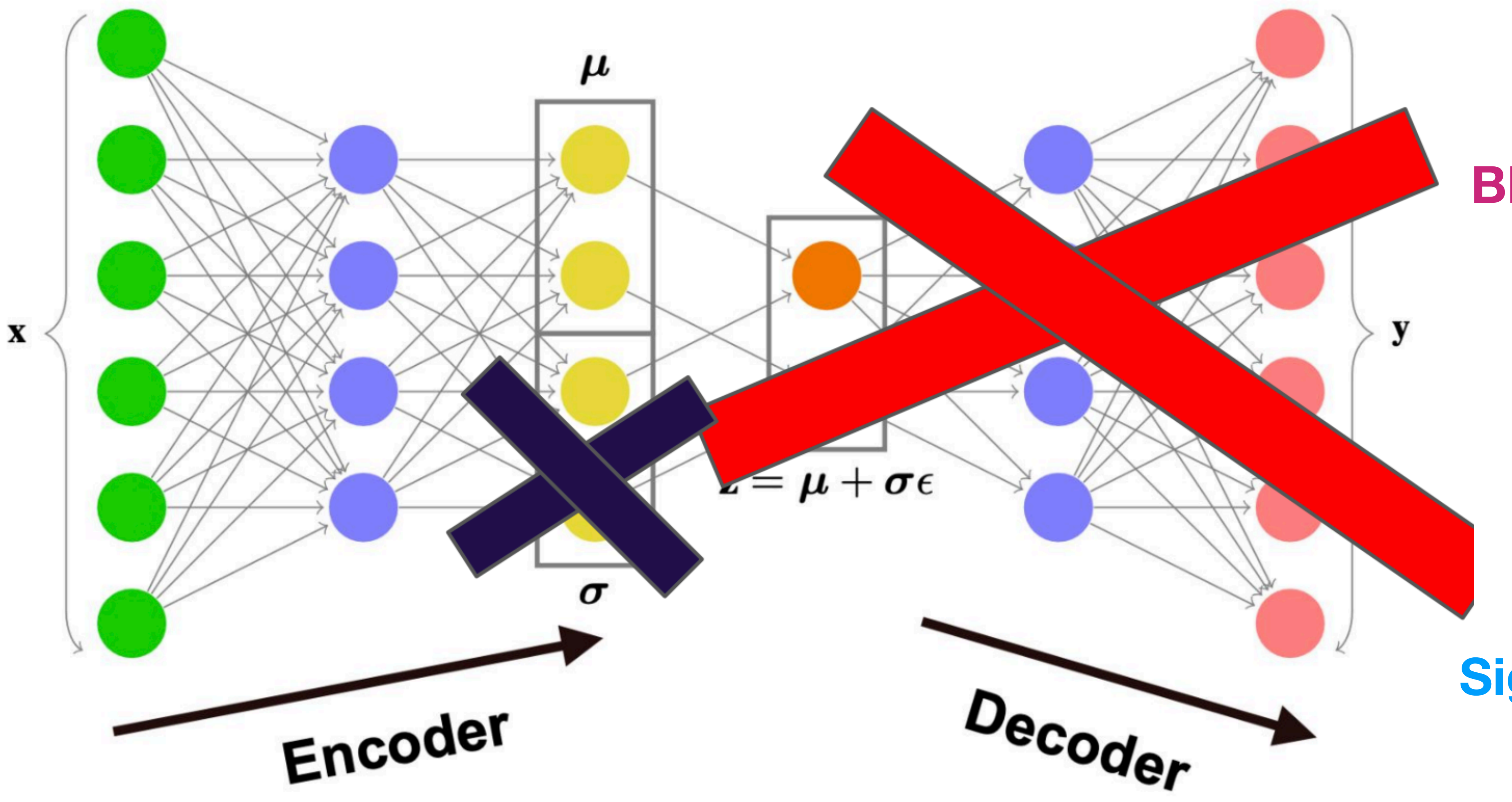
- Most common AD algorithms are autoencoders (AEs)
- Can reduce network size by removing decoder, using latent space directly (allows to achieve <50 ns latency)

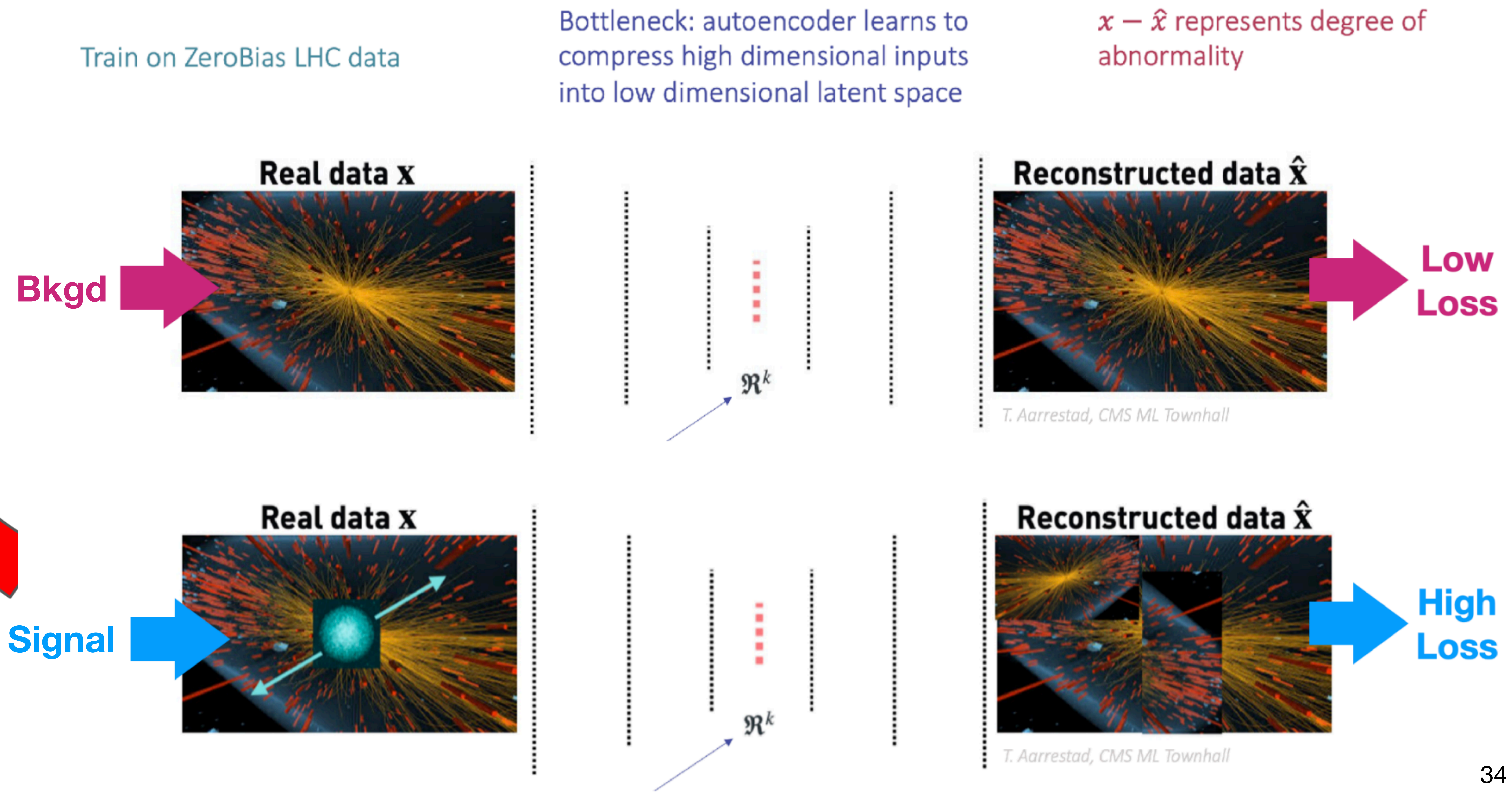




L1 Trigger AD

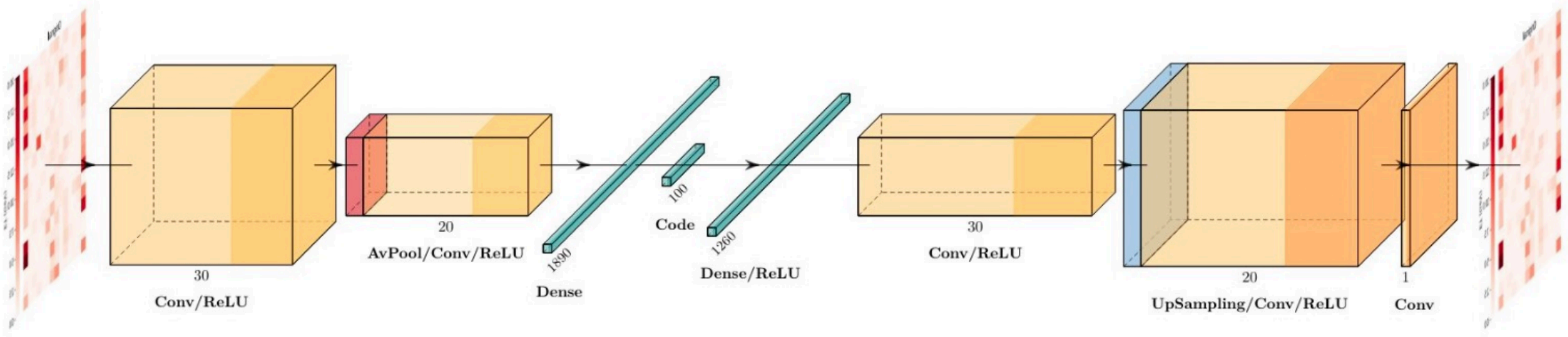
- Most common AD algorithms are autoencoders (AEs)
- Can reduce network size by removing decoder, using latent space directly (allows to achieve <50 ns latency)





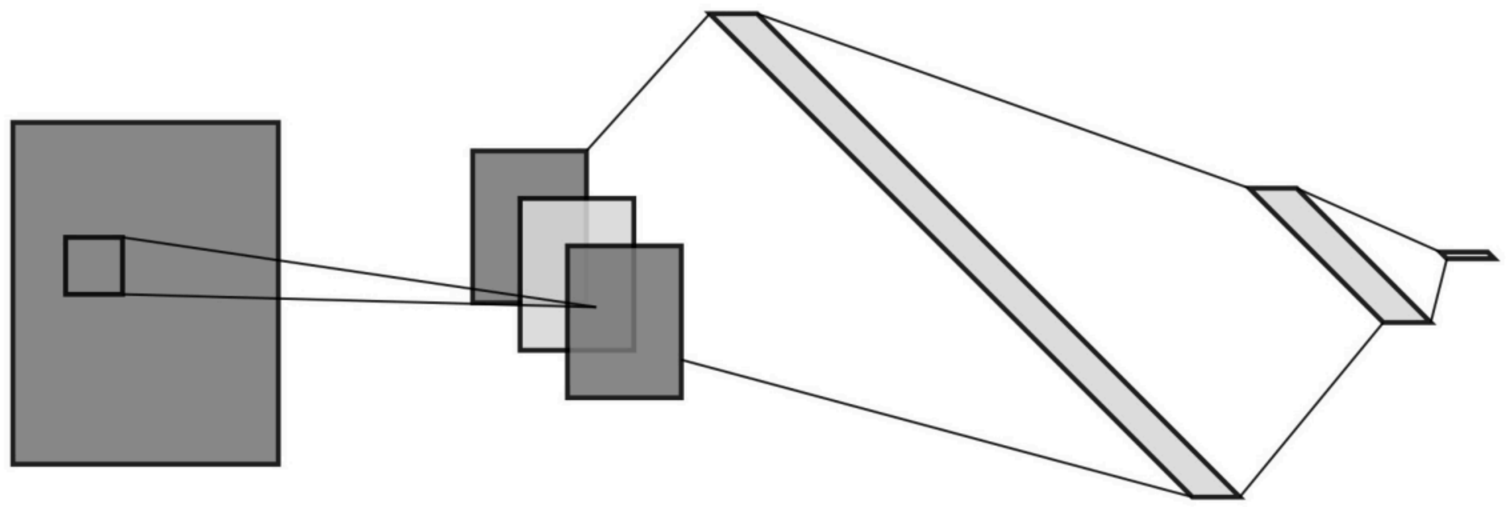
L1 Trigger AD

- Most common AD algorithms are autoencoders (AEs)
- Can reduce network size by knowledge distillation, training student network to predict teacher network MSE (allows to achieve <50 ns latency)



$$\mathscr{L} = \|x - x_{\text{pred}}^{\text{teacher}}\|^2$$

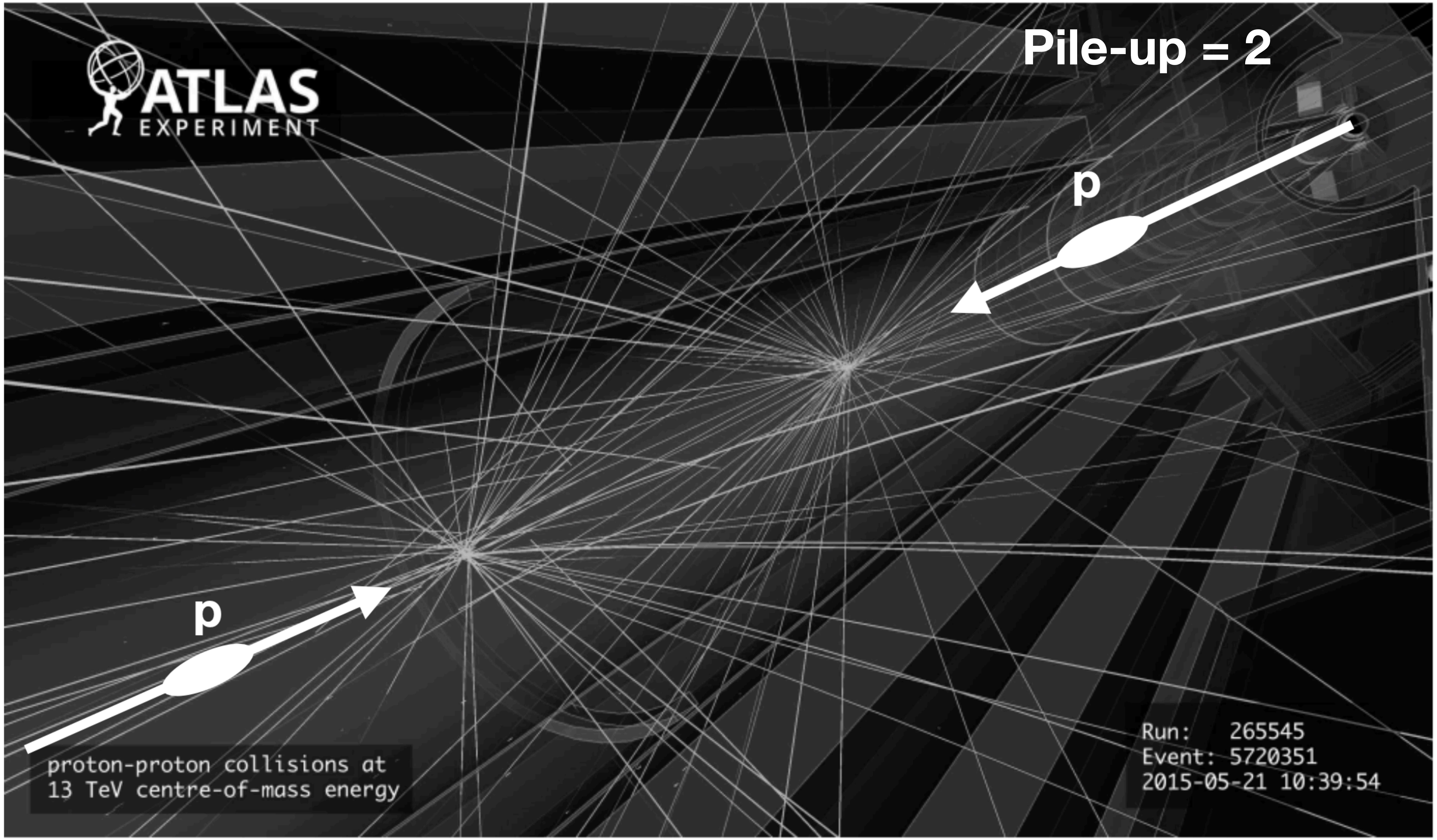
Teacher network



Student network

$$\mathscr{L} = \|(\|x - x_{\text{pred}}^{\text{teacher}}\|^2) - x_{\text{pred}}^{\text{student}}\|^2$$

LHC Pileup (PU)



LHC Pileup (PU) - ~Current

HL-LHC Pileup (PU) - Future

ML Size / Complexity

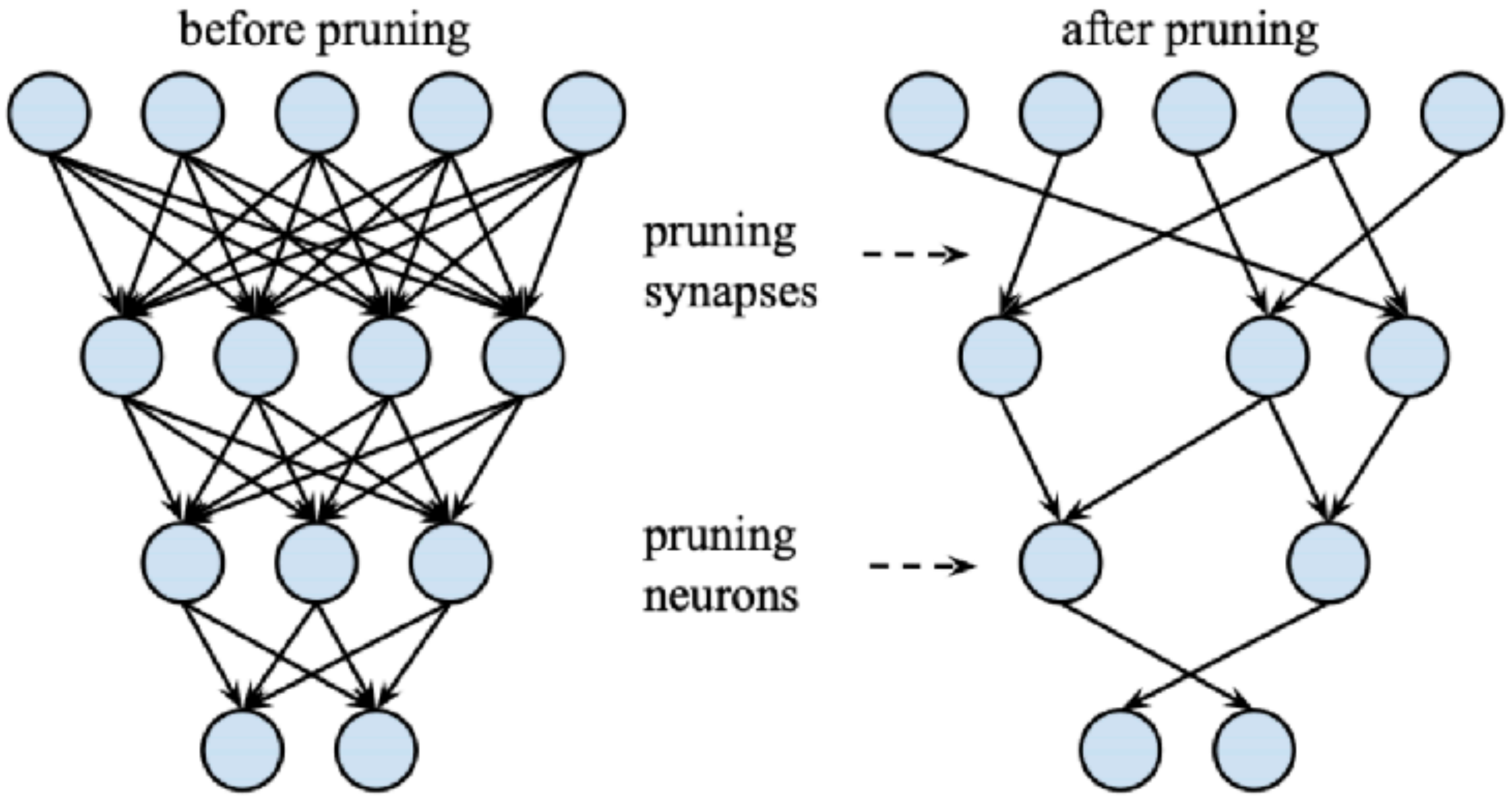
- Regardless of toolkit, big limitation of doing ML fast is device size
  - Bigger device → more resources → more computation → larger ML models

Xilinx Virtex Ultrascale+ VU13P
12288 Multipliers
1.7M LUTs
3.4M FFs
95 Mb BRAM

- Alternatively, is it possible to reduce network size without hurting performance?
  - ***Pruning*** and ***quantization*** are two potential ways

Pruning

- Are all the pieces a given network necessary?
- Many different types of pruning
  - Structured vs. unstructured
- Multiplications by 0 can be completely removed from FPGA design



Quantization

- FPGAs are well suited to fixed-point numbers, not floating point
- Number of bits can be adjusted as needed (impacts accuracy, performance, resources)
- Can greatly reduce number of bits needed by training with knowledge of quantization