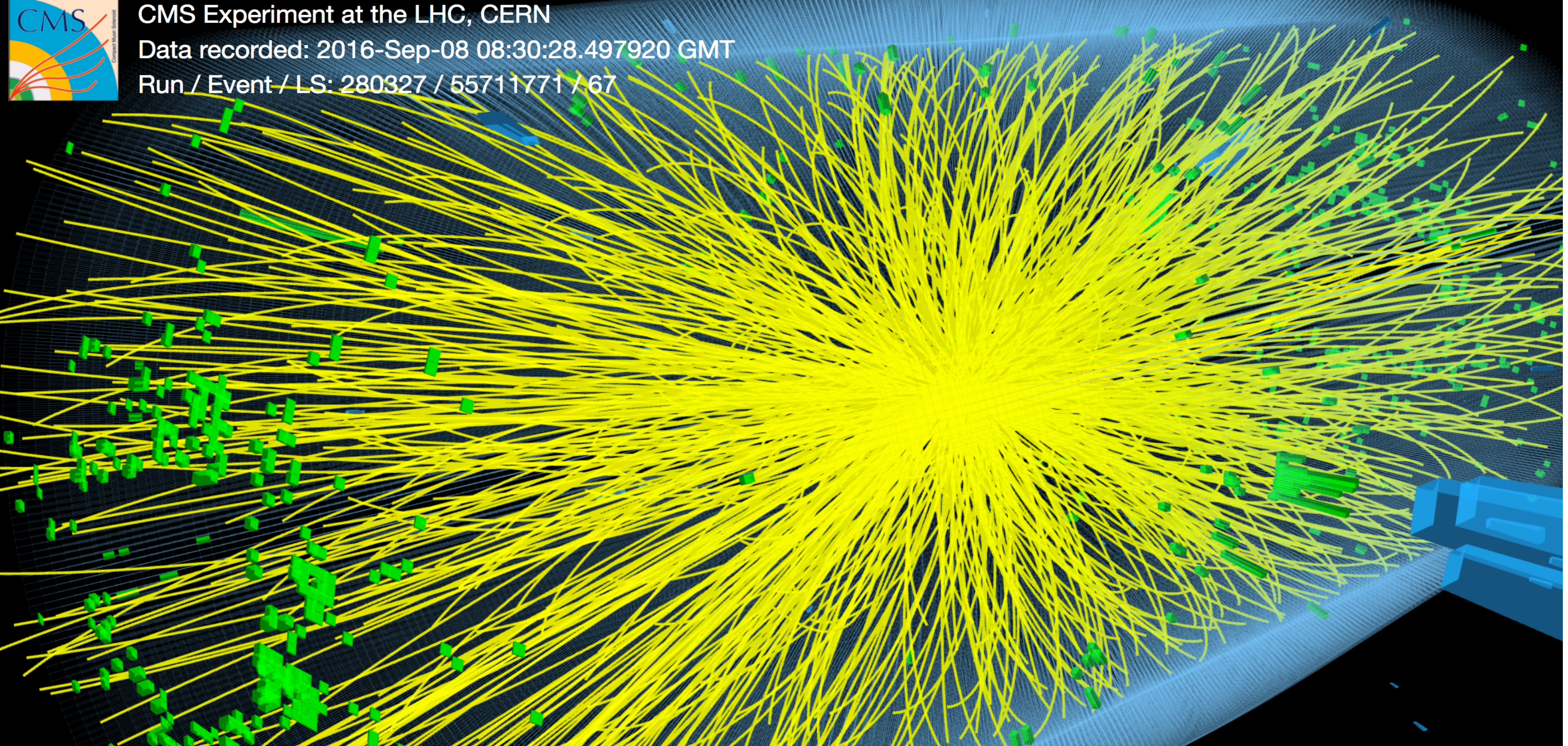
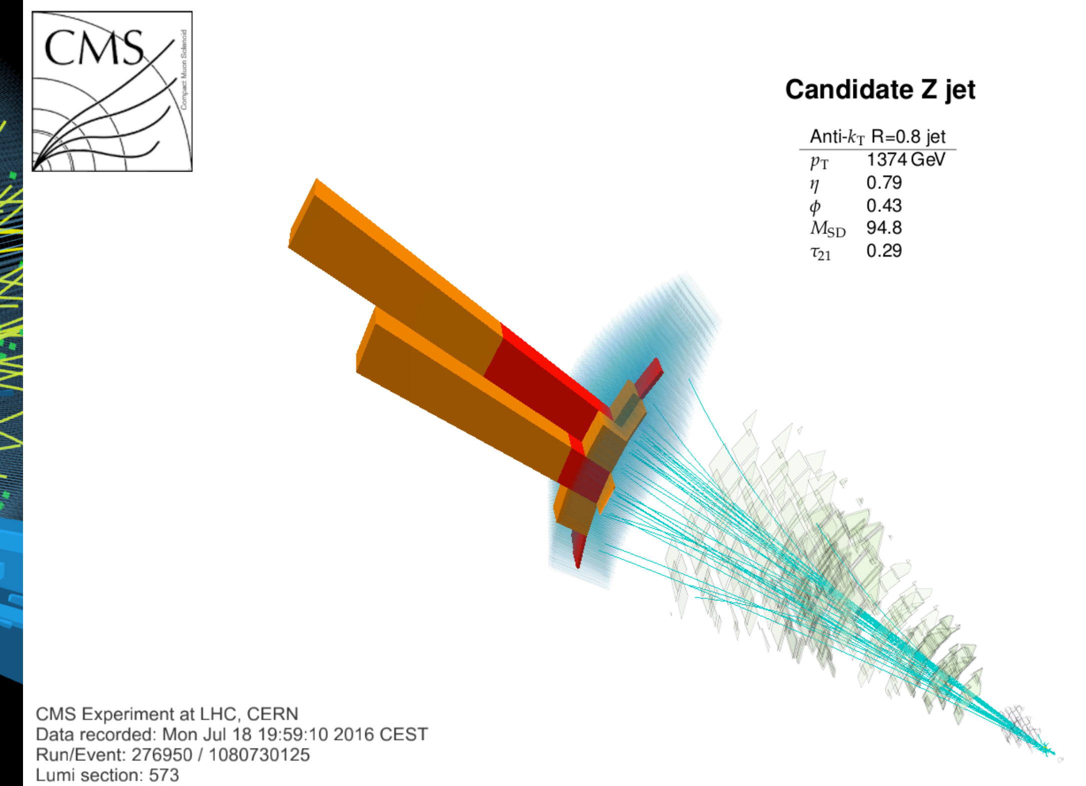




Particle flow and PUPPI @ L1 for CMS at HL-LHC

Can we trigger on boosted jets at 25ns?

Jennifer Ngadiuba (CERN)

on behalf of the CMS Collaboration

BOOST 2018
July 16 - 20
Campus de Jussieu, Paris

The challenge: triggering at (HL-)LHC

Squeeze the beams to increase data rates
→ multiple pp collisions per bunch crossing (pileup)

2016: <PU> ~ 20-50
2017 + Run 3: <PU> ~ 50-80
HL-LHC: 140-200

CHALLENGE: maintain physics in increasingly complex collision environment
→ untriggered events lost forever!

Sophisticated techniques needed to preserve the physics!

Particle flow and PUPPI
THIS TALK!

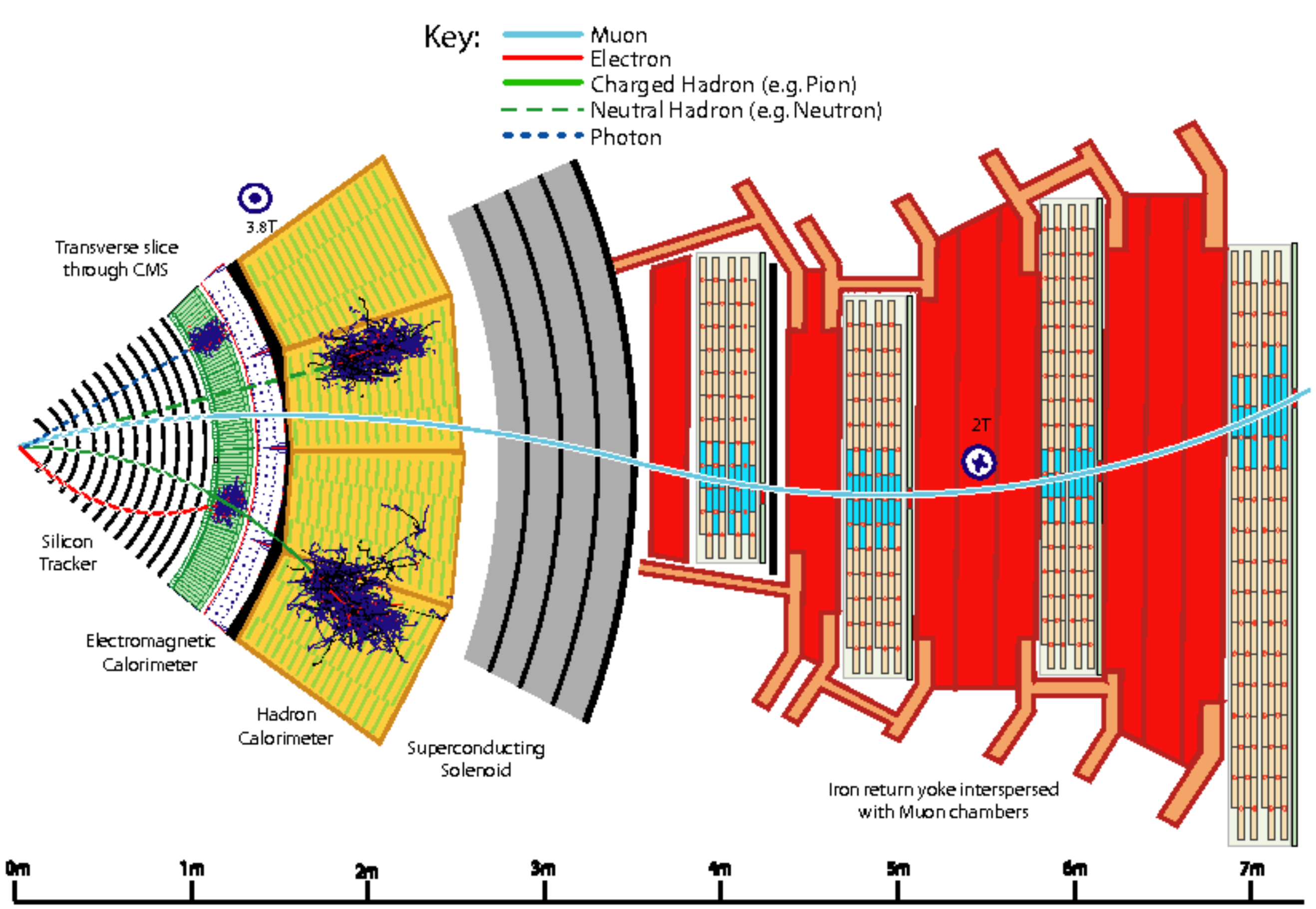
Machine learning
(see talk on Thursday)

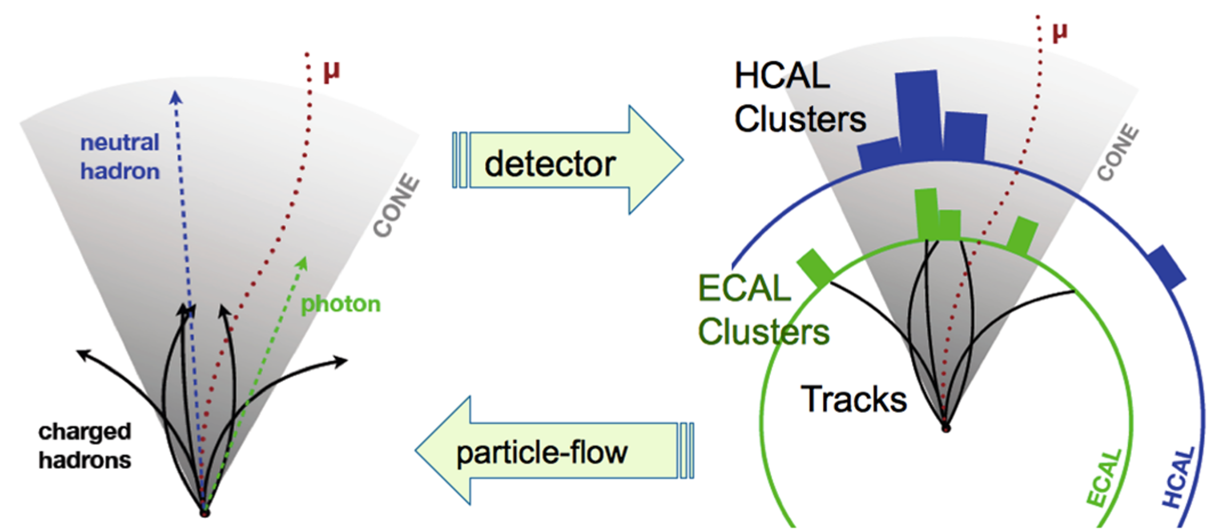
Particle flow

Efficient combination of complementary detector subsystems
Reconstruct and identify individually all particles
Improves any single system energy/spatial resolution

JINST 12 (2017) P10003

Key ingredients: **efficient track reconstruction** for charged particles and **fine granularity calorimeter**



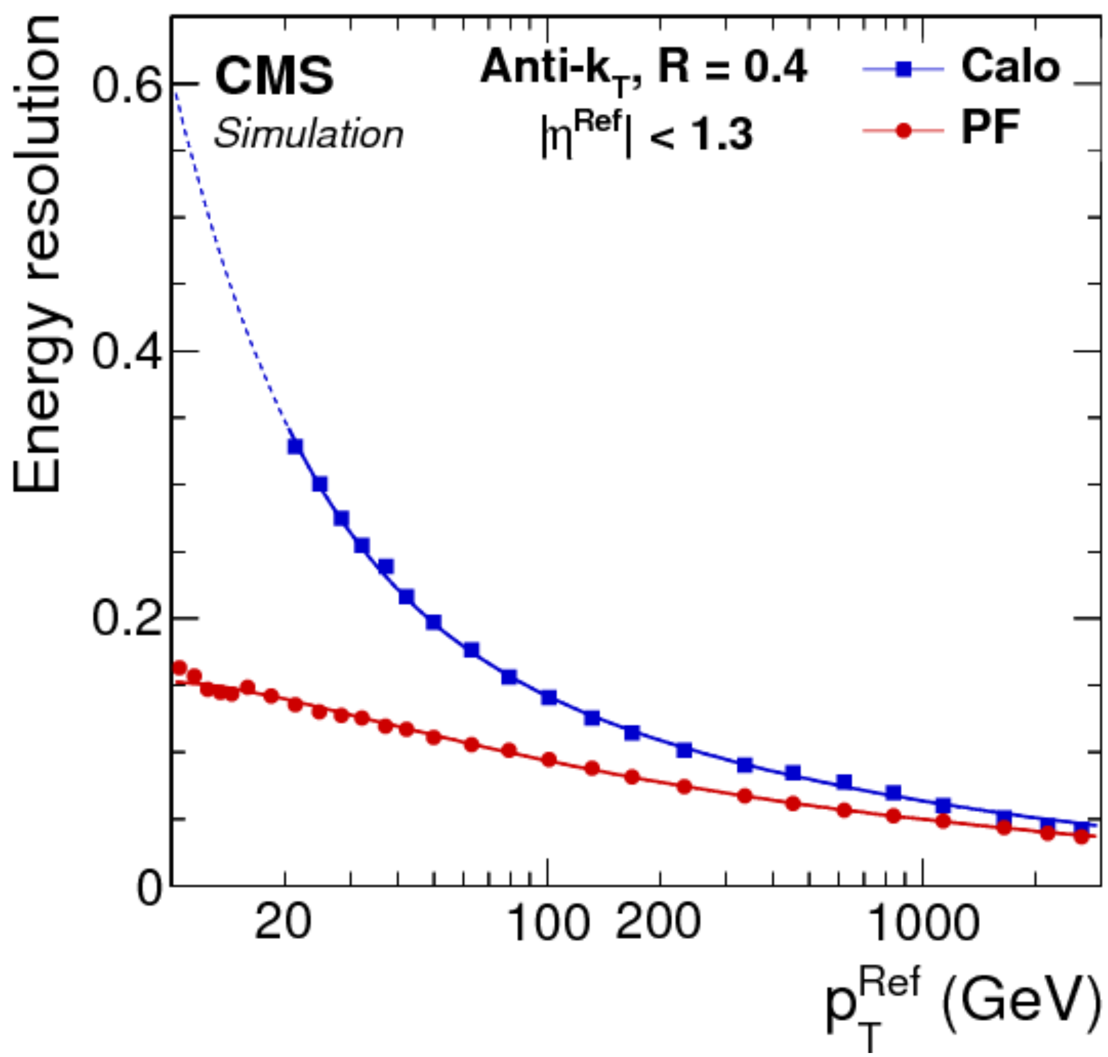


muons, electrons, photons, neutral hadrons, charged hadrons

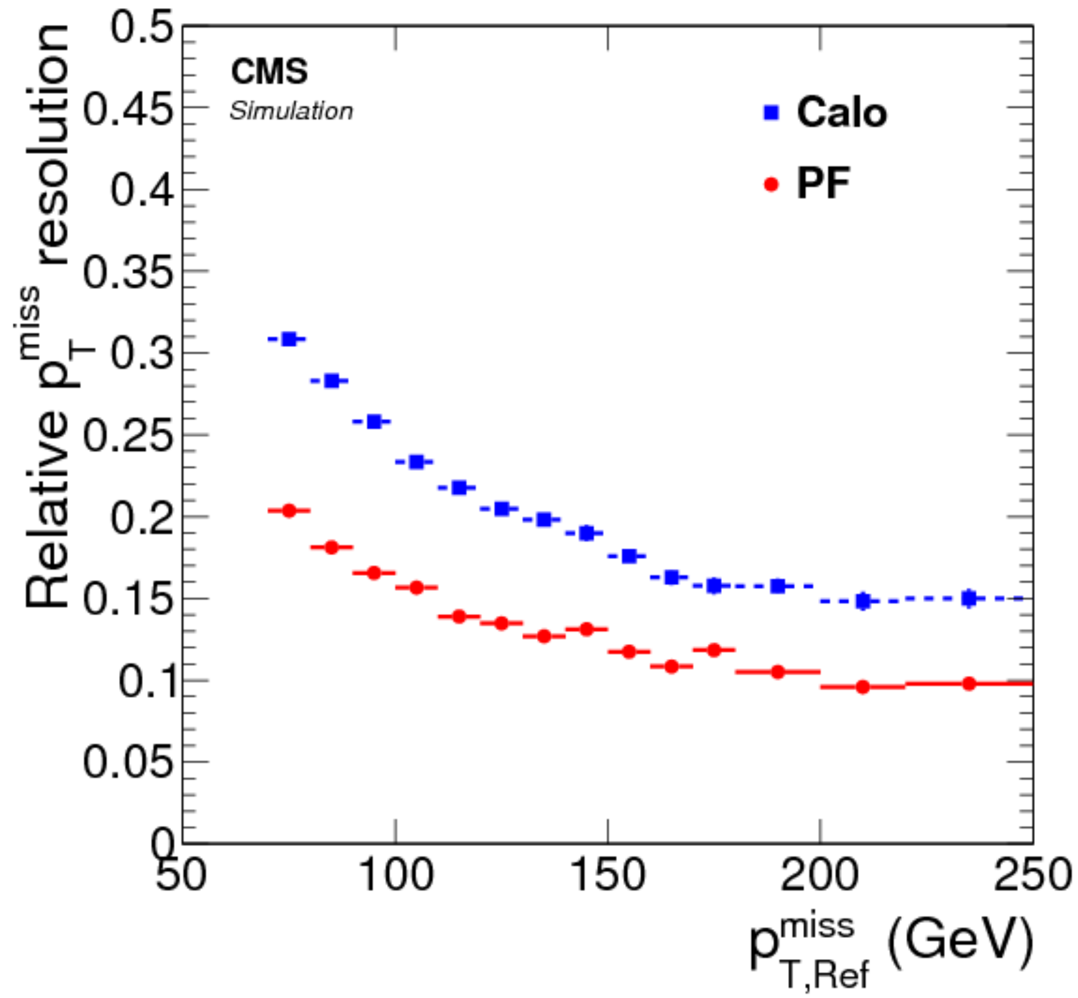
Particle flow

Widely used in offline and HLT event reconstruction since LHC Run 1

Particle flow physics performance impact JINST 12 (2017) P10003



Improved jet p_T resolution



Improved missing p_T resolution

PUPPI

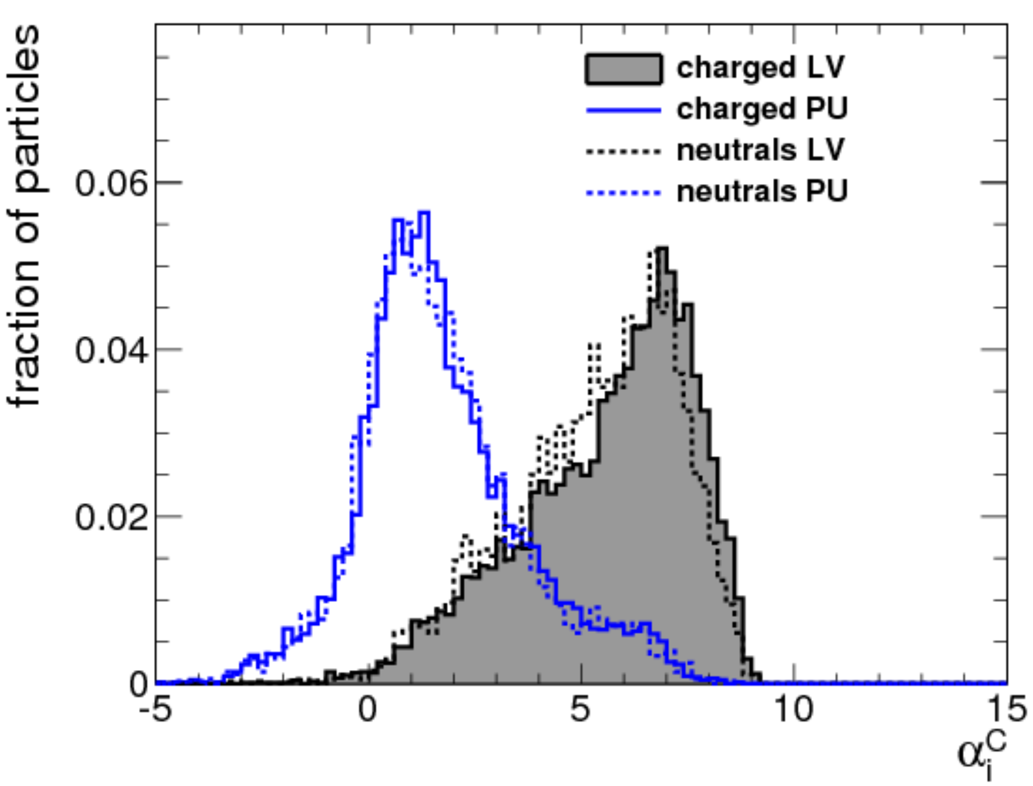
Particle flow: reconstructs all particles, also from PU interactions

PileUp Per Particle Identification (PUPPI):
an algorithm that determines, per particle, a weight for **how likely** a particle is from PU
key insight: using QCD ansatz to infer neutral pileup contribution

1. Define a local discriminant, **α**, between PU and LV (leading vertex)

$$\alpha_i^C = \log\left[\sum_{j\in \mathrm{Ch,LV}} \frac{p_{T,j}}{\Delta R_{ij}}\Theta(R_0 - \Delta R_{ij})\right]$$

2. Get data driven α distribution for PU using charged PU tracks



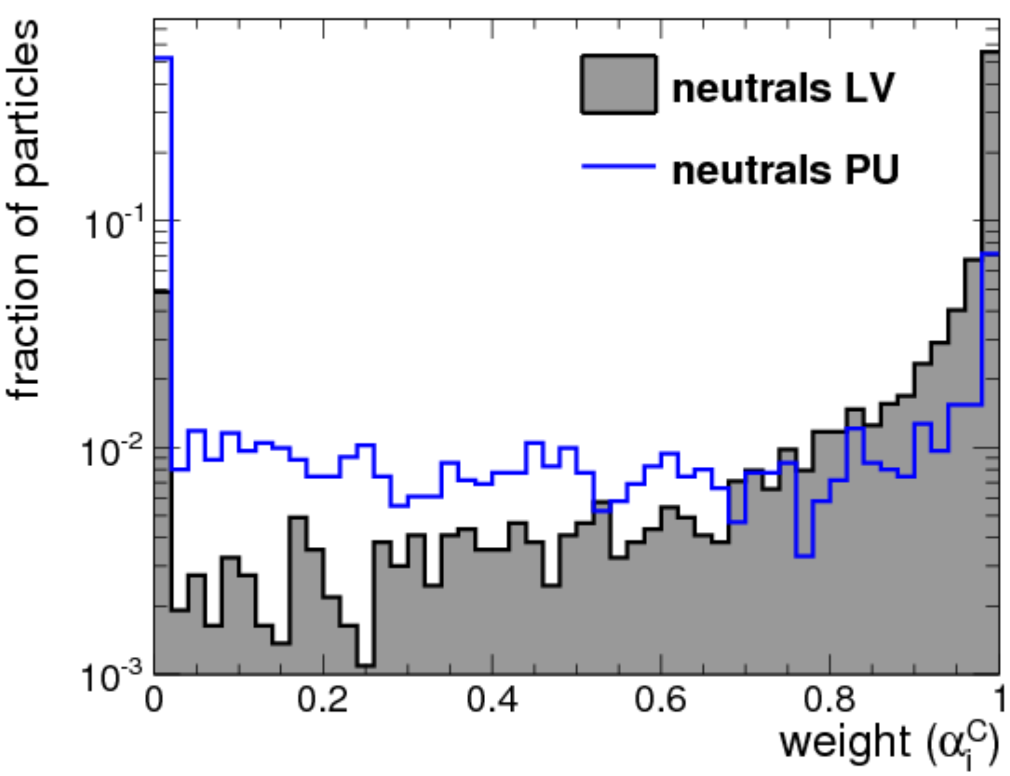
PUPPI

Particle flow: reconstructs all particles, also from PU interactions

PileUp Per Particle Identification (PUPPI):
an algorithm that determines, per particle, a weight for **how likely** a particle is from PU
key insight: using QCD ansatz to infer neutral pileup contribution

1. Define a local discriminant, **α**, between PU and LV (leading vertex)
2. Get data driven **α** distribution for PU using charged PU tracks
3. For the neutrals, ask "how un-PU-like is **α** for this particle?", compute a **weight**
4. Reweight the 4-vector of the particle by this **weight**, then proceed to interpret the event as usual

$$\alpha_i^C = \log\left[\sum_{j \in \mathrm{Ch,LV}} \frac{p_{T,j}}{\Delta R_{ij}} \Theta(R_0 - \Delta R_{ij})\right]$$

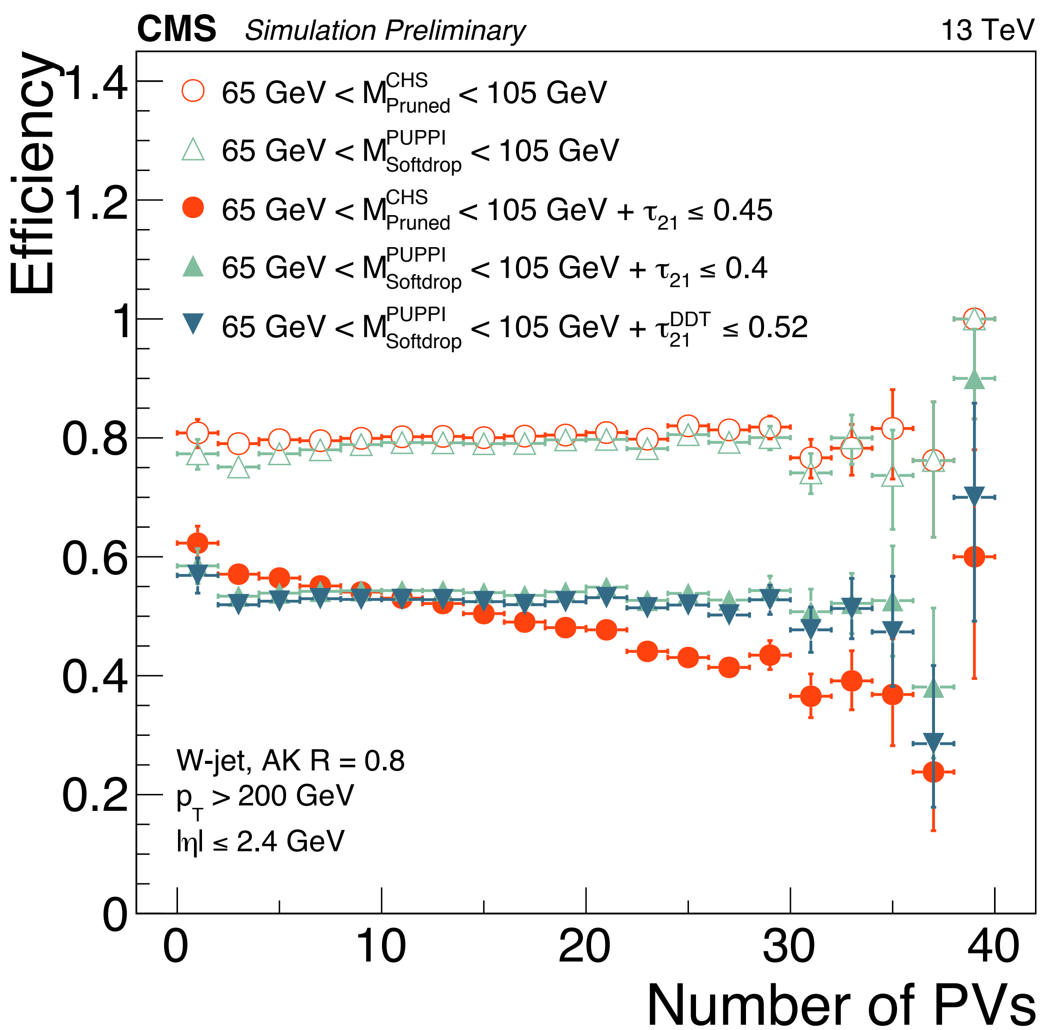


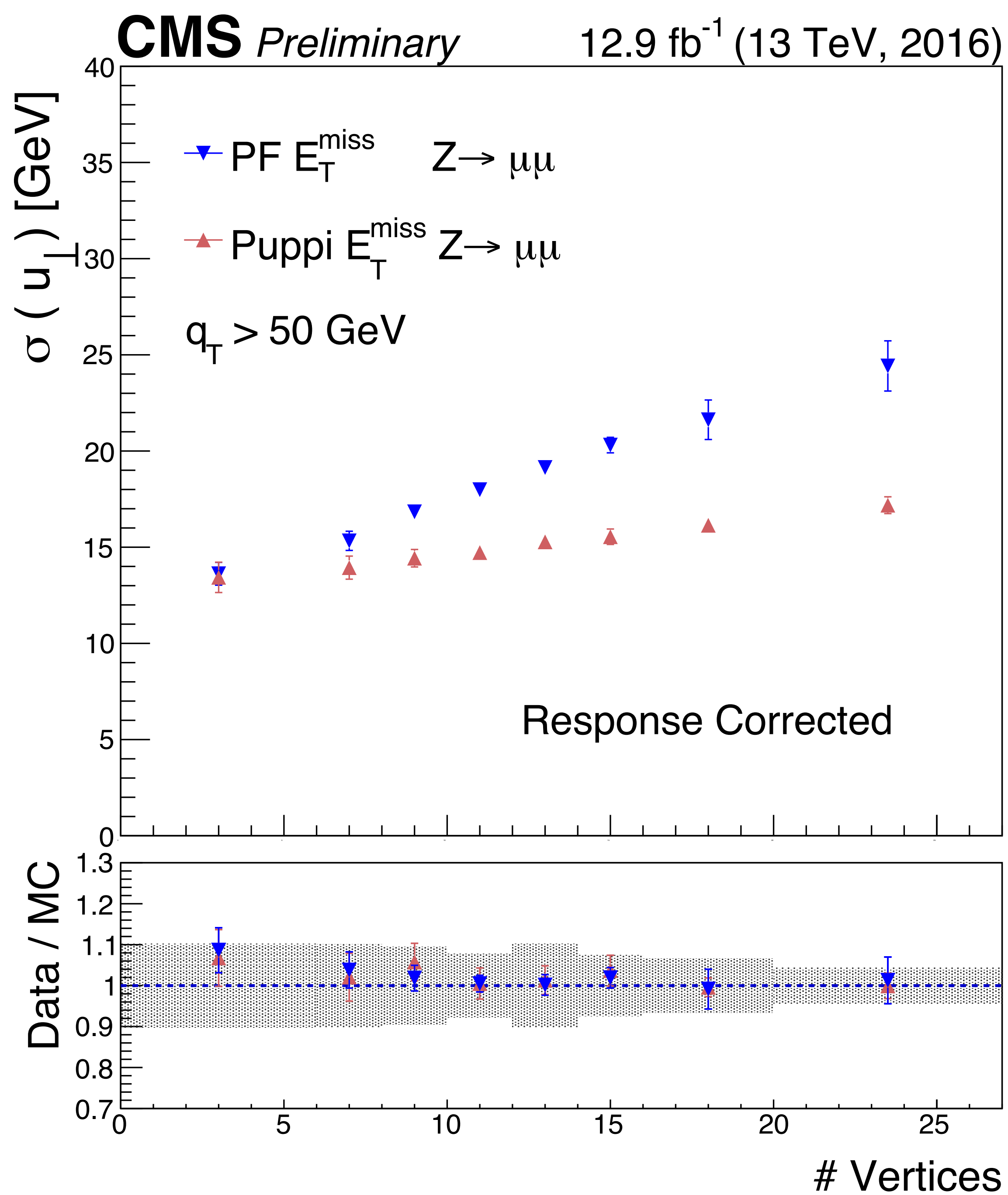
PUPPI

arxiv.1407.6013 (original),
LHCC-P-008 (CMS Phase-2 upgrade),
JME-14-001, JME-16-003, JME-16-004
many CMS physics analyses…

Large gain with PUPPI, especially at high pile-up

Jet p_T and missing p_T resolution
Fake jet rate
Jet substructure
Lepton isolation

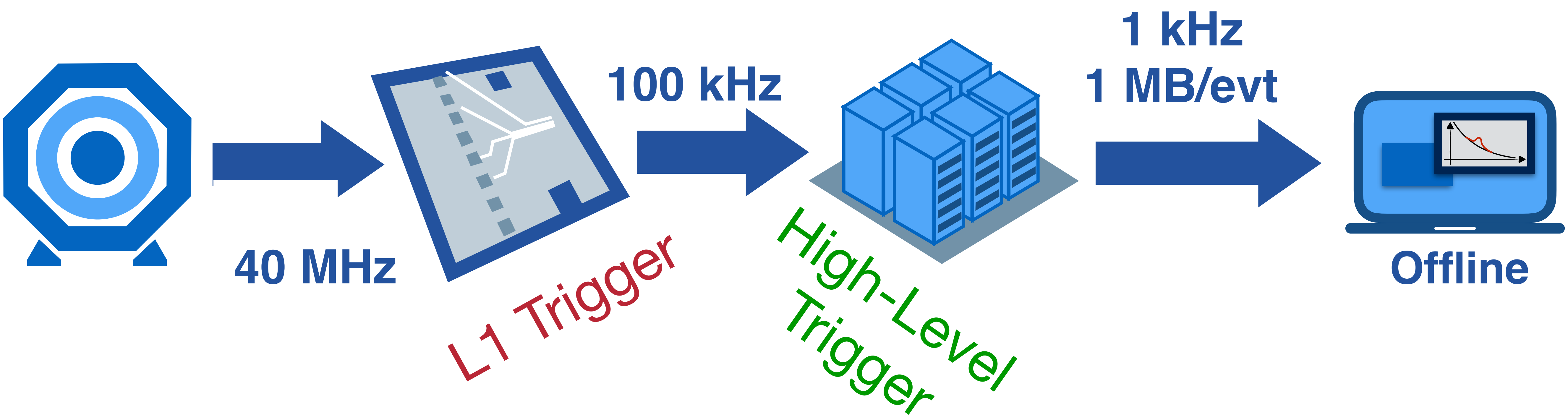




Results from BOOST17, see new results in A. Benecke's talk on Tuesday

The CMS trigger system

Triggering typically performed in multiple stages



Trigger decision to be made in O(μs)

Latencies require all-FPGA design

Computing farm for detailed analysis of the full event

Latency O(100 ms)

For HL-LHC upgrade: latency and output rates will increase by ~ 3 (from 3.8 → 12.5 μs @ L1)

Upgraded CMS trigger for HL-LHC

Support 40 MHz readout of track segments
Reconstruct all tracks with $p_T > 2$ GeV, $|\eta| < 2.4$ within a latency of 4 us

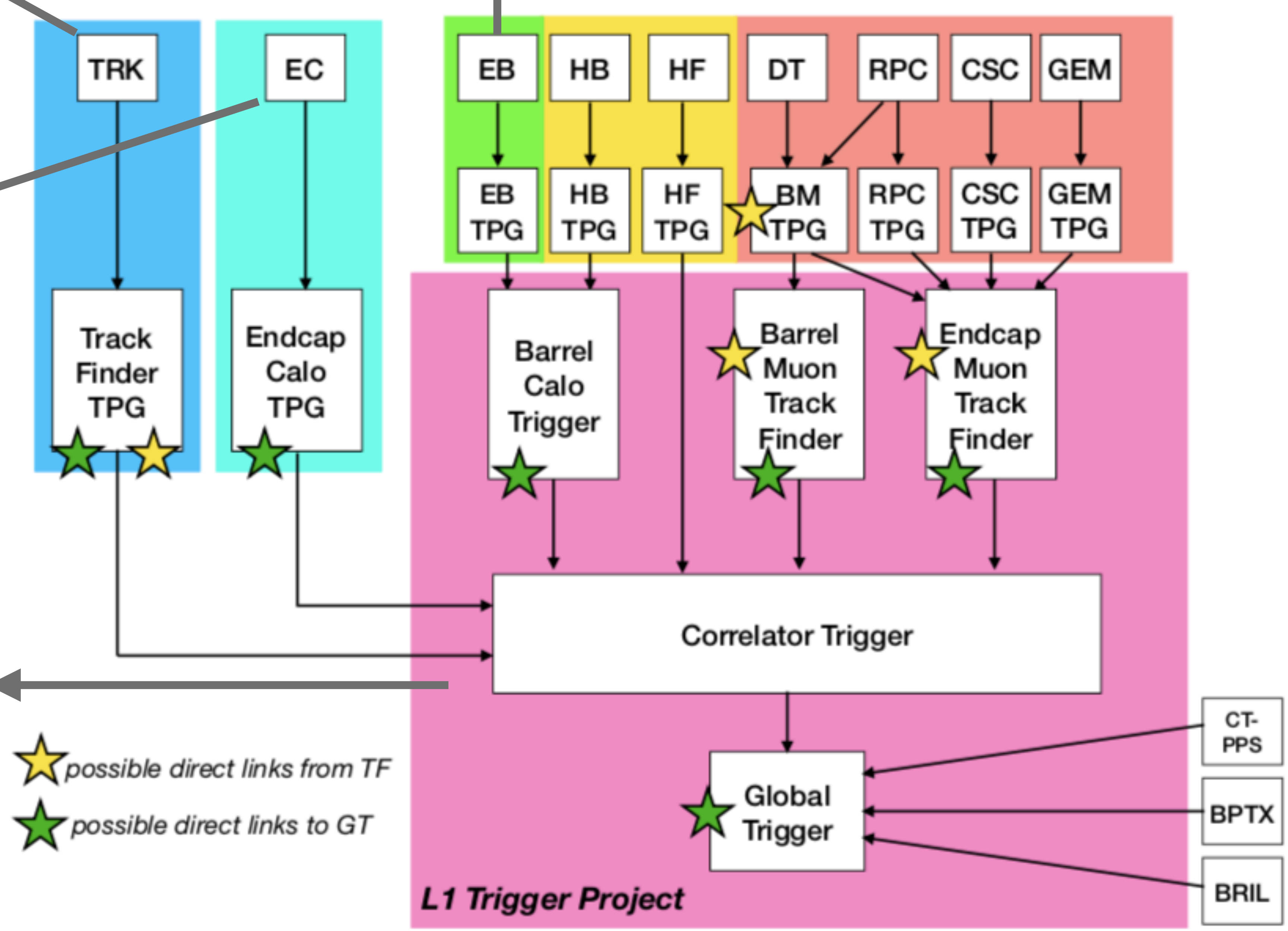
Updated backend for electromagnetic barrel calorimeter (EB) supporting **full granularity readout** at L1 ($\Delta\eta \times \Delta\phi \sim 0.02 \times 0.02$)
nb,current L1 granularity: $\Delta\eta \times \Delta\phi \sim 0.1 \times 0.1$

High-granularity endcap calorimeter

LHCC-P-008

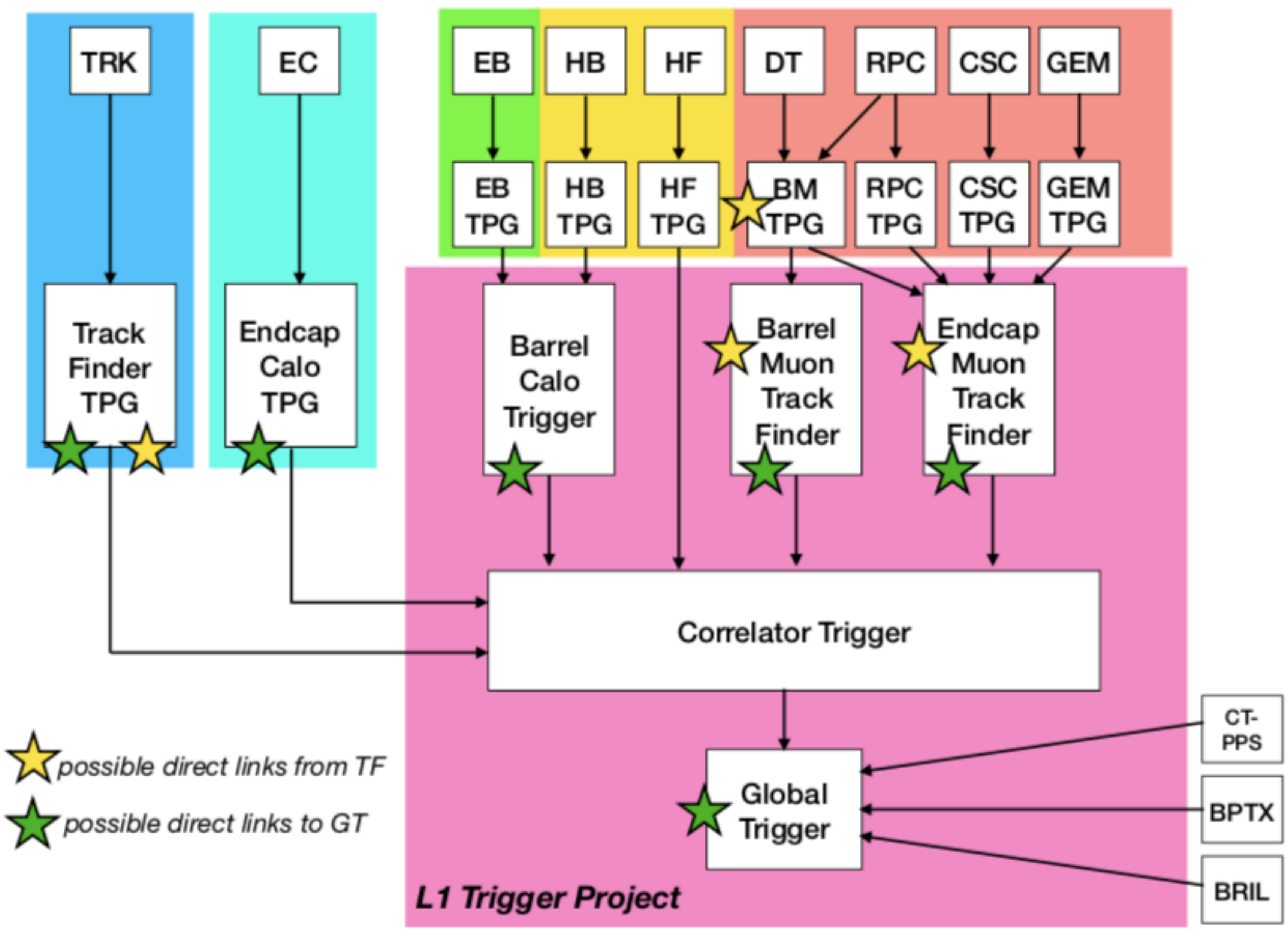
LHCC-2017-009

New L1 trigger system with **improved processing power**, a **global correlator** layer, and **deeper buffers** in all subsystems to allow a total latency up to 12.5 μs

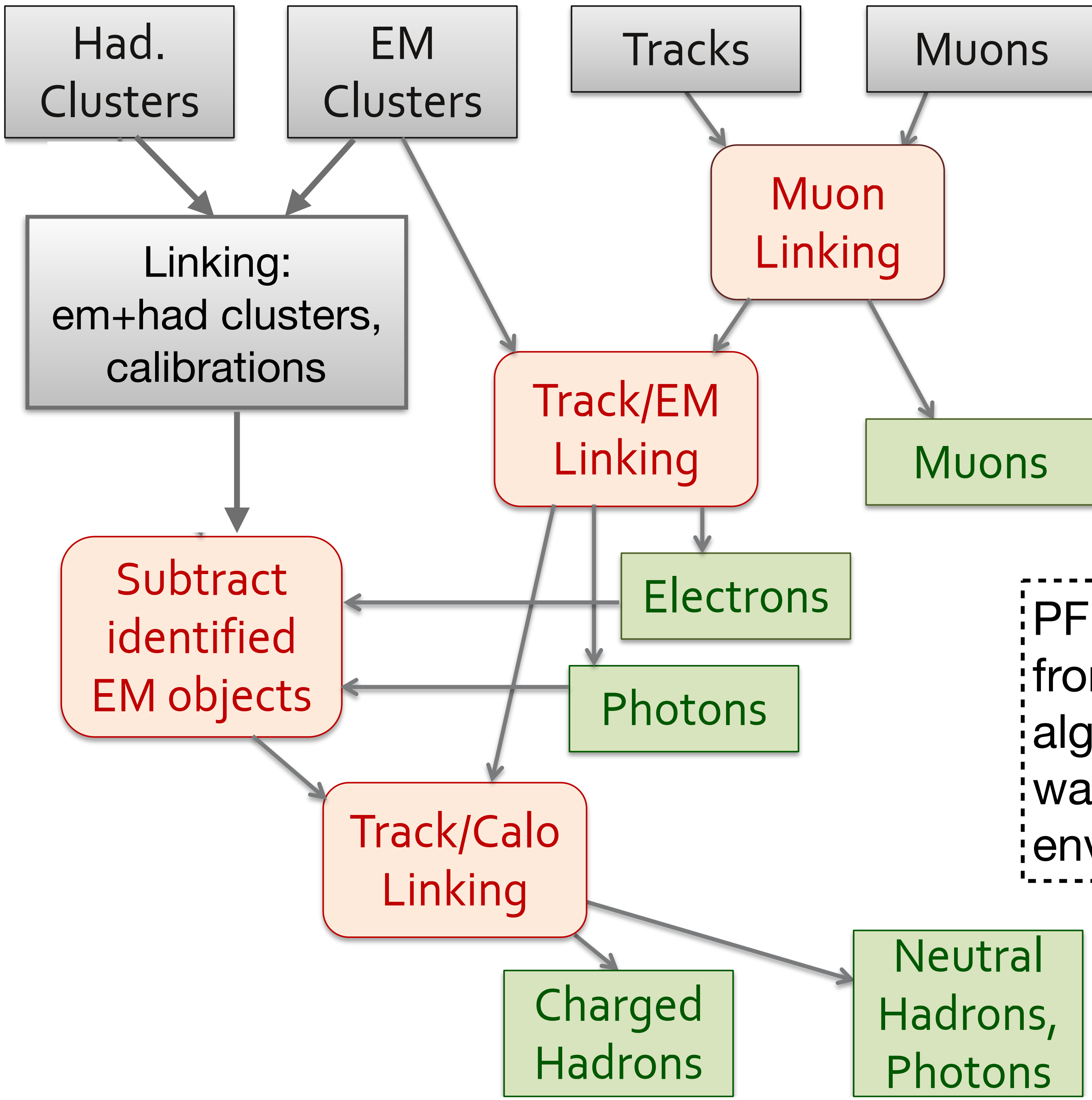


Upgraded CMS trigger for HL-LHC

Addition of **tracks** to L1 trigger is a game-changer
Re-think algorithms: how to best combine *tracking, calorimeter, and muon information*
The big challenge: mitigation of **pileup**

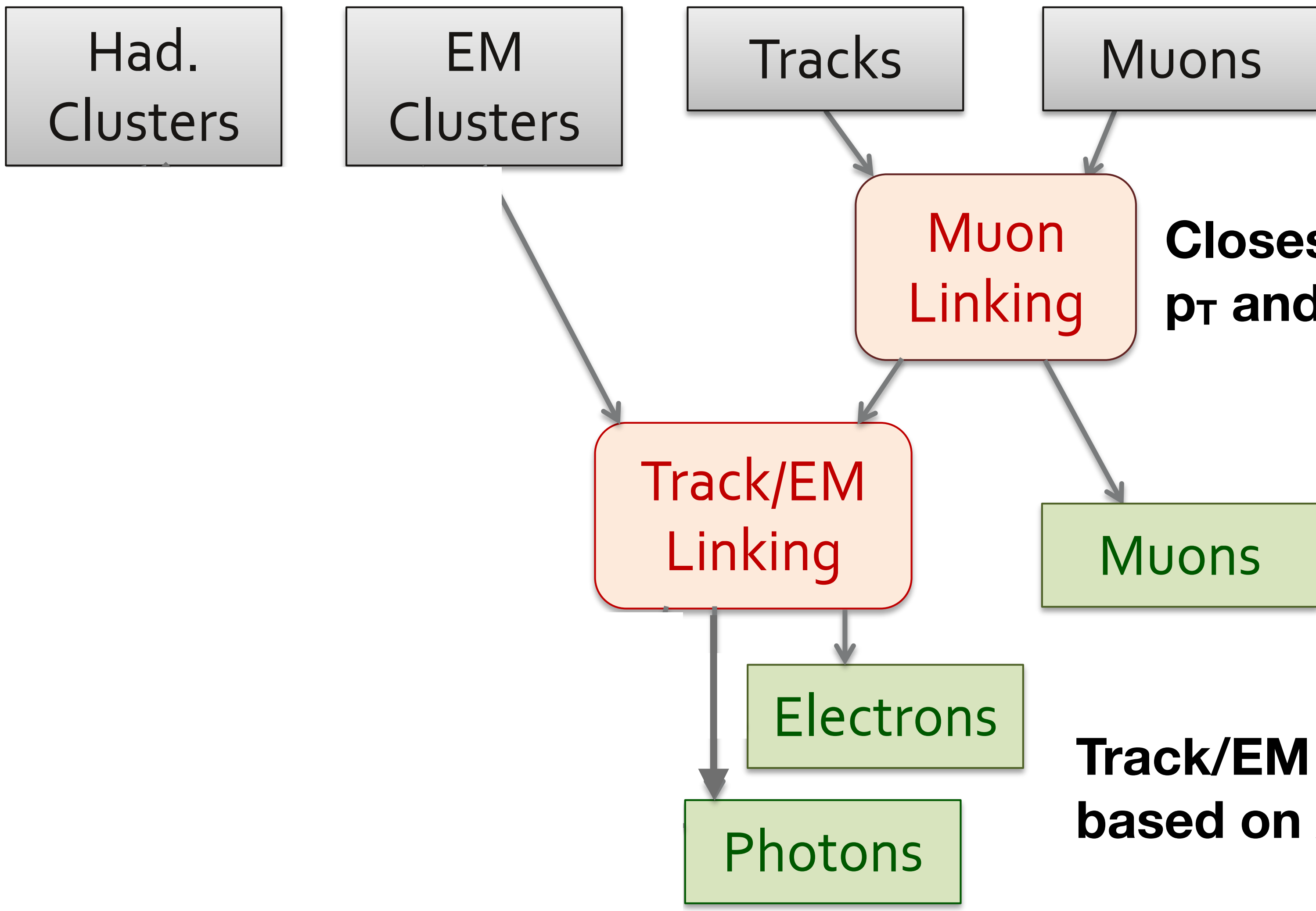


PF algorithm @ L1



PF algorithm was re-designed from first principles, as the algorithm used offline and at HLT was not suitable to the L1 environment

PF algorithm @ L1



Inputs to the algorithm

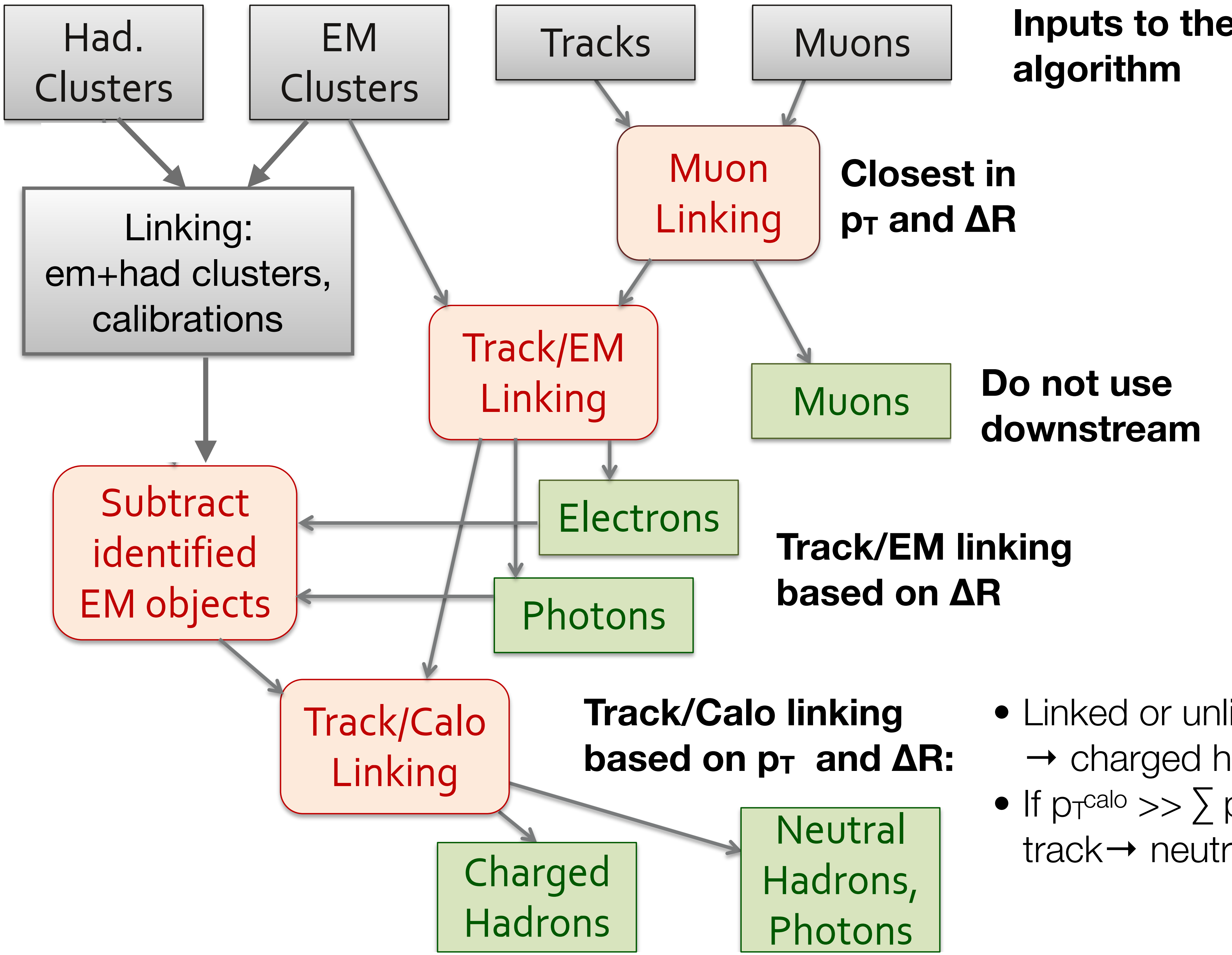
Closest in p_T and ΔR

Do not use downstream

Track/EM linking based on ΔR:

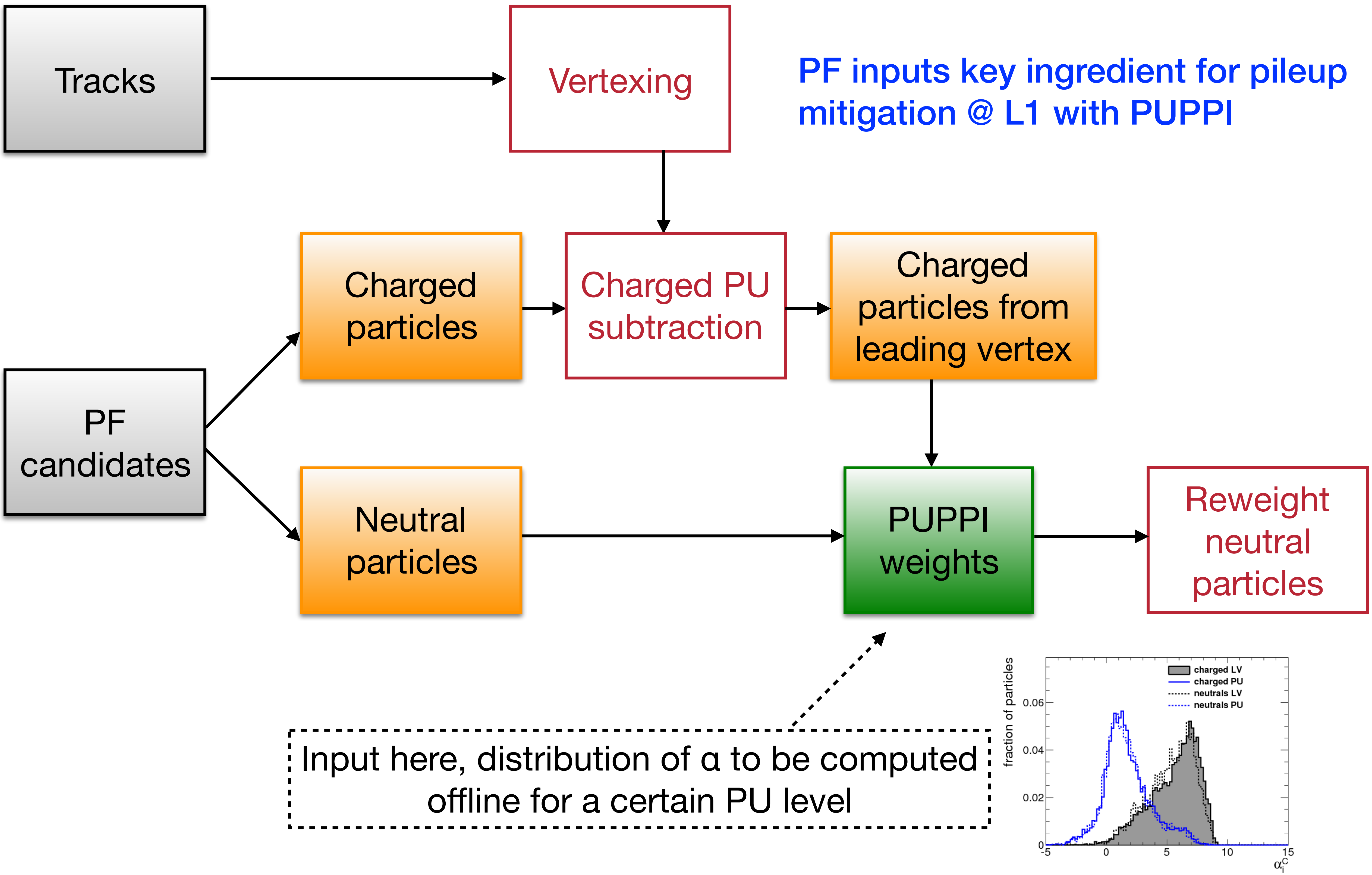
- Clusters with no associated tracks → photon
- If $p_T^{cluster} \geq \sum p_T^{track}$ within uncertainties → electron
- If $p_T^{cluster} >> \sum p_T^{track}$ → electron+photon

PF algorithm @ L1



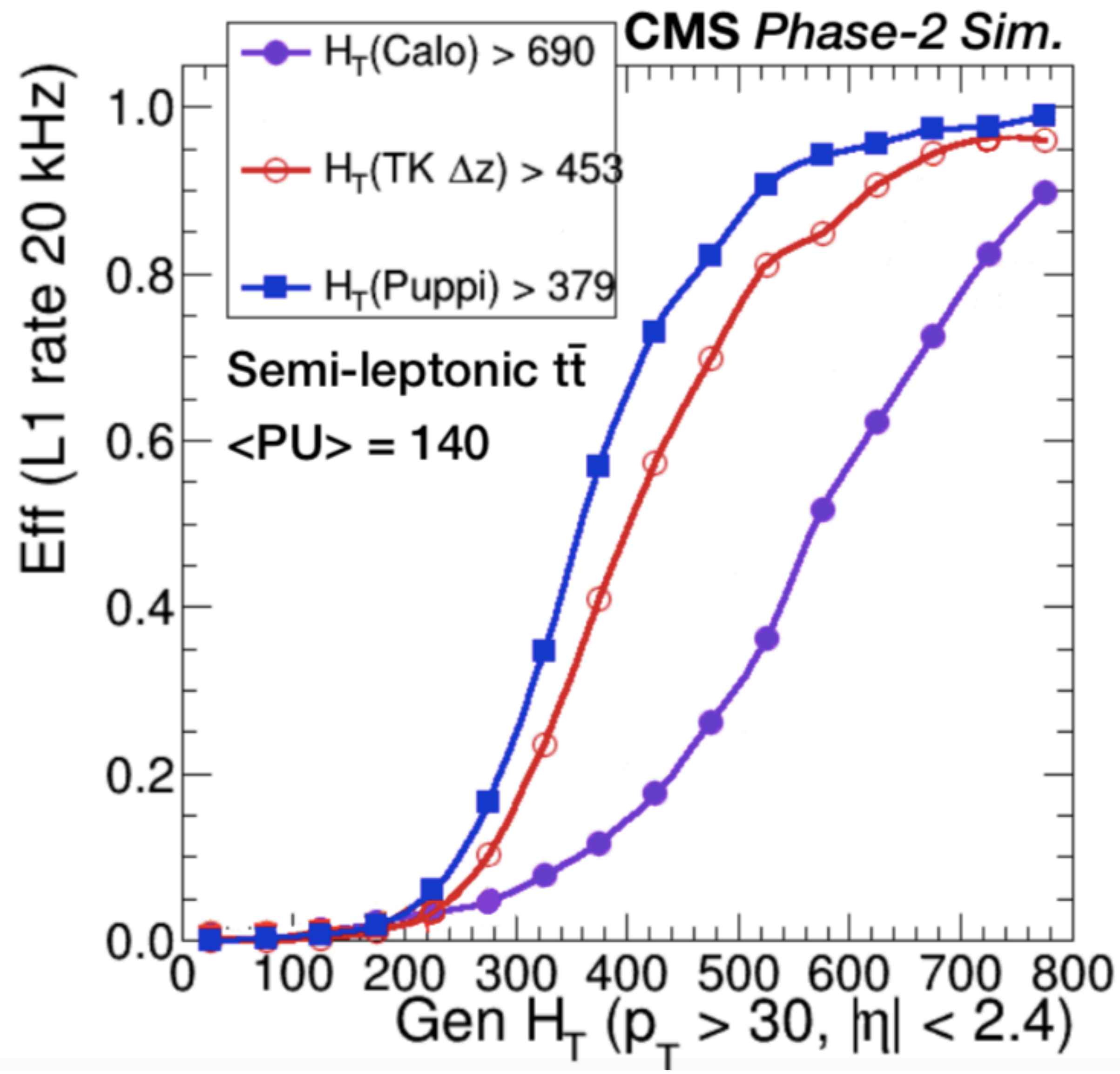
- Linked or unlinked low p_T tracks → charged hadron
- If $p_T^{calo} >> \sum p_T^{track}$ or unlinked track → neutral hadron or photon

PF+PUPPI algorithm @ L1

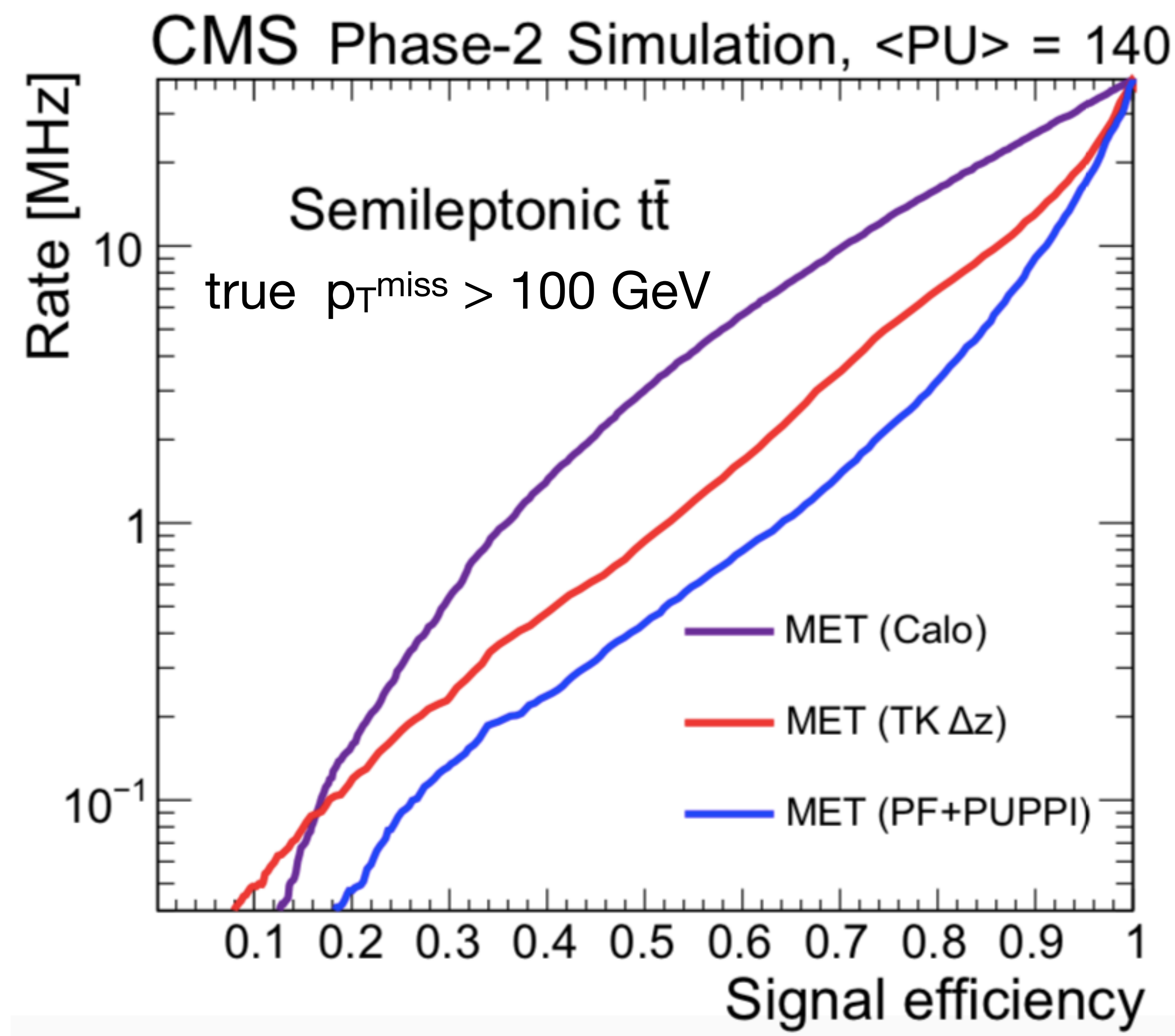


Performance for MET and jets

Gains in rate reduction, H_T and p_T^{miss} resolution, signal efficiency, lower threshold



For a fixed L1 accept rate of 20 kHz:
lower threshold and sharper turn-on



Missing p_T:
better trade-off between L1 rate and signal efficiency

LHCC-2017-009

What are FPGAs?

Latencies at L1 trigger require all-FPGA design

Field Programmable Gate Arrays are reprogrammable integrated circuits

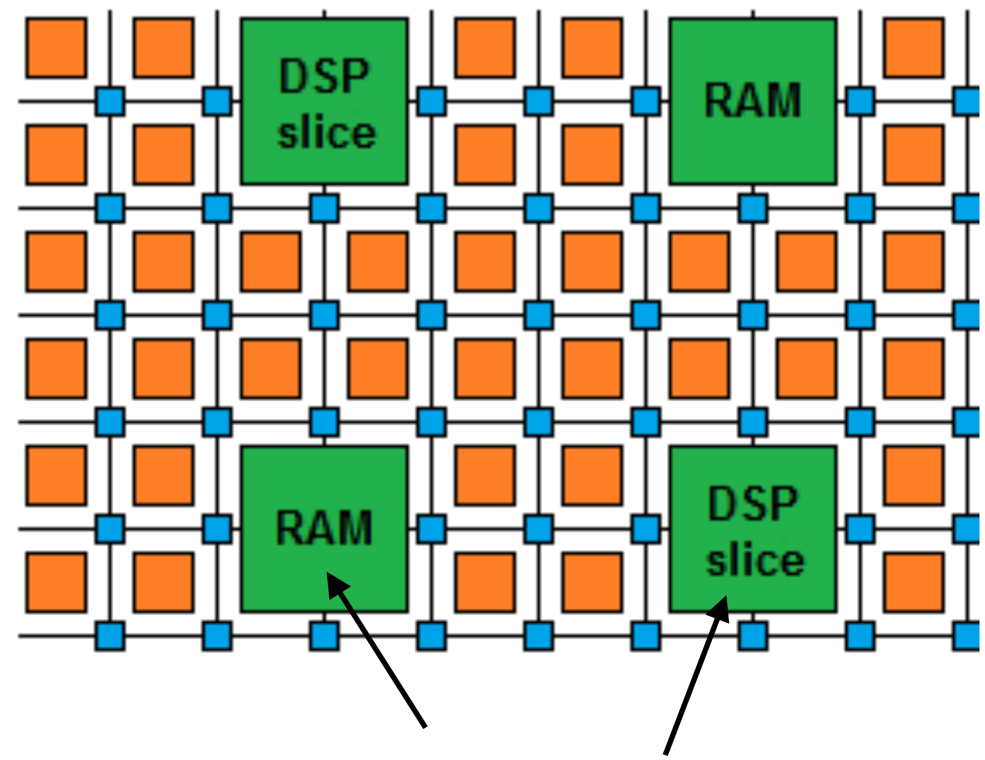
Contain array of **logic cells** embedded with **DSPs**, **BRAMs**, etc.

High speed input/output to handle the large bandwith

Support **highly parallel** algorithm implementations

Low power (relative to CPU/GPU)

FPGA diagram



Digital Signal Processors (DSPs): logic units used for multiplications

Random-access memories (RAMs): embedded memory elements

Flip-flops (FF) and look up tables (LUTs) for additions

How are FPGAs programmed?

Latencies at L1 trigger require all-FPGA design

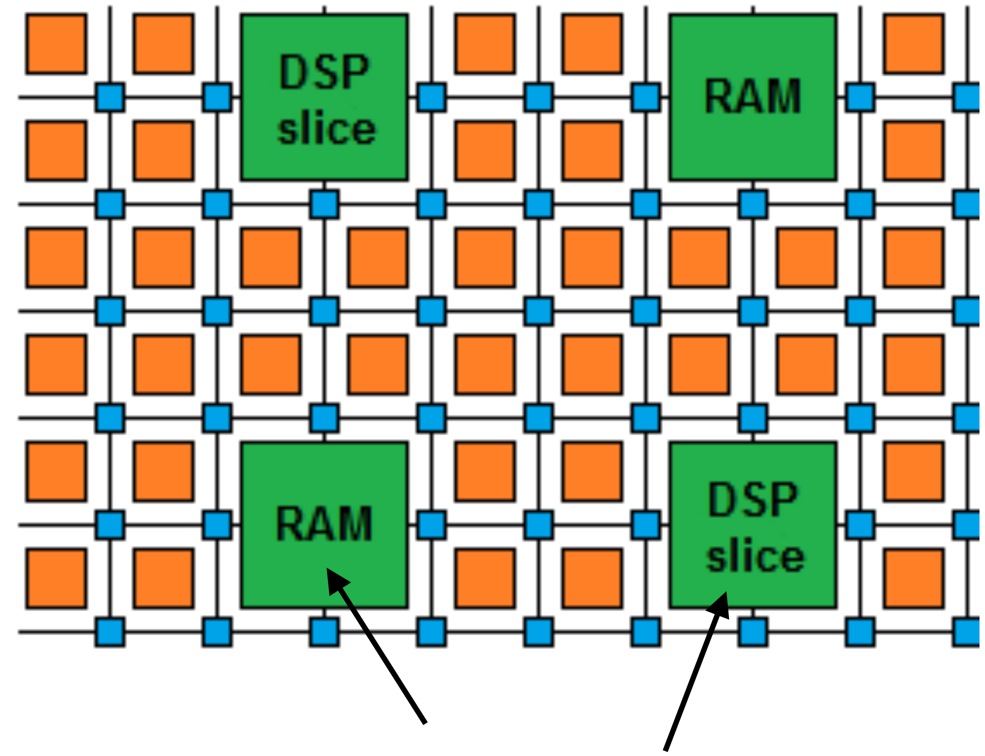
High Level Synthesis is used to compile the algorithms into a firmware block (IP core)

- generate standard register-transfer level (RTL) code for FPGA from more common C/C++ code
- pre-processor directives and constraints used to optimize the timing
- firmware for PF developed in 2-3 months, only physicists

We use Xilinx Vivado HLS

Main reference: *https://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_1/ug902-vivado-high-level-synthesis.pdf*

FPGA diagram



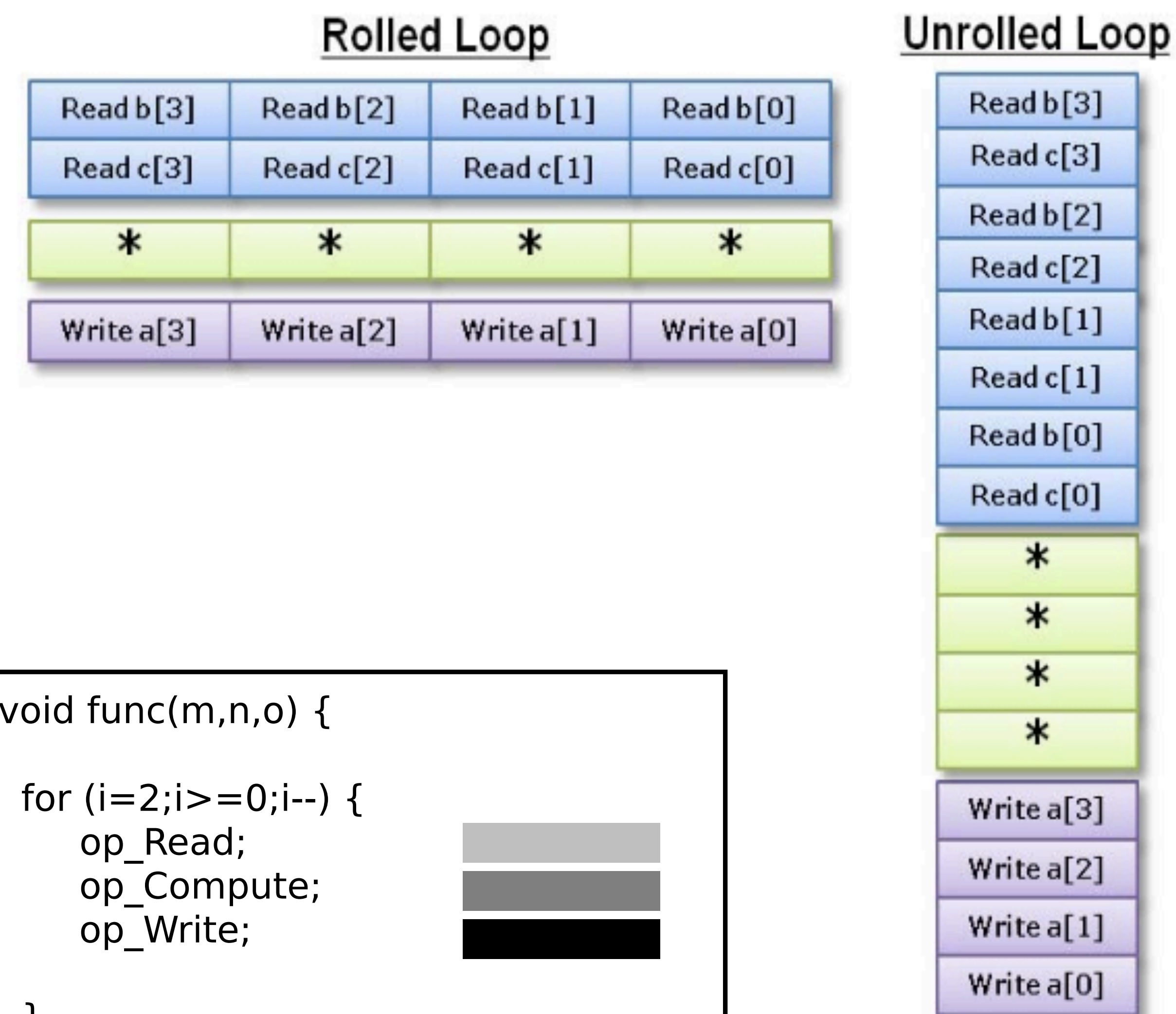
Digital Signal Processors (DSPs): logic units used for multiplications

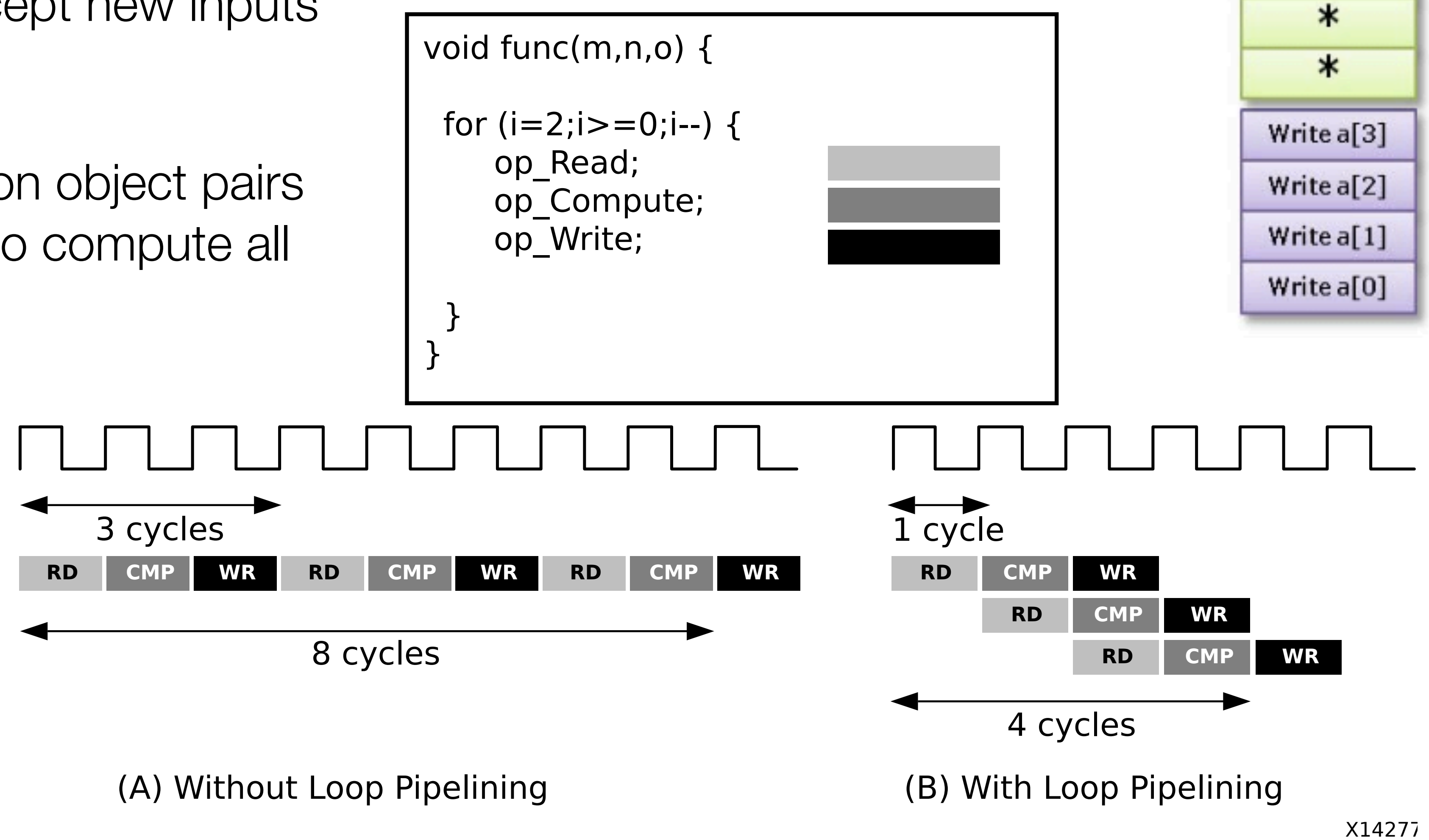
Random-access memories (RAMs): embedded memory elements

Flip-flops (FF) and look up tables (LUTs) for additions

Firmware implementation

- To best profit from FPGA capabilities:
  - use integers instead of floating-point
  - keep the mathematics simple
- Expolit parallelism to **guarantee latency**
  - the algo is **pipelined** to accept new inputs every 1 or 2 clock cycles
  - combinatorical **loops**, e.g. on object pairs in the linking, are **unrolled** to compute all values in parallel





Firmware implementation

- To best profit from FPGA capabilities:
  - use integers instead of floating-point
  - keep the mathematics simple
- Expolit parallelism to **guarantee latency**
  - the algo is **pipelined** to accept new inputs every 1 or 2 clock cycles
  - combinatorical **loops**, e.g. on object pairs in the linking, are **unrolled** to compute all values in parallel

Preliminary estimate from HLS

1 Xilinx Virtex Ultrascale+ (VUP9)
4 $\Delta\eta \times \Delta\phi$ = 0.6 x 0.6 regions
25 tracks + 20 clusters every BX
~ 40% resource usage
0.7 μs latency: 550 ns for PF, 150 ns for PUPPI

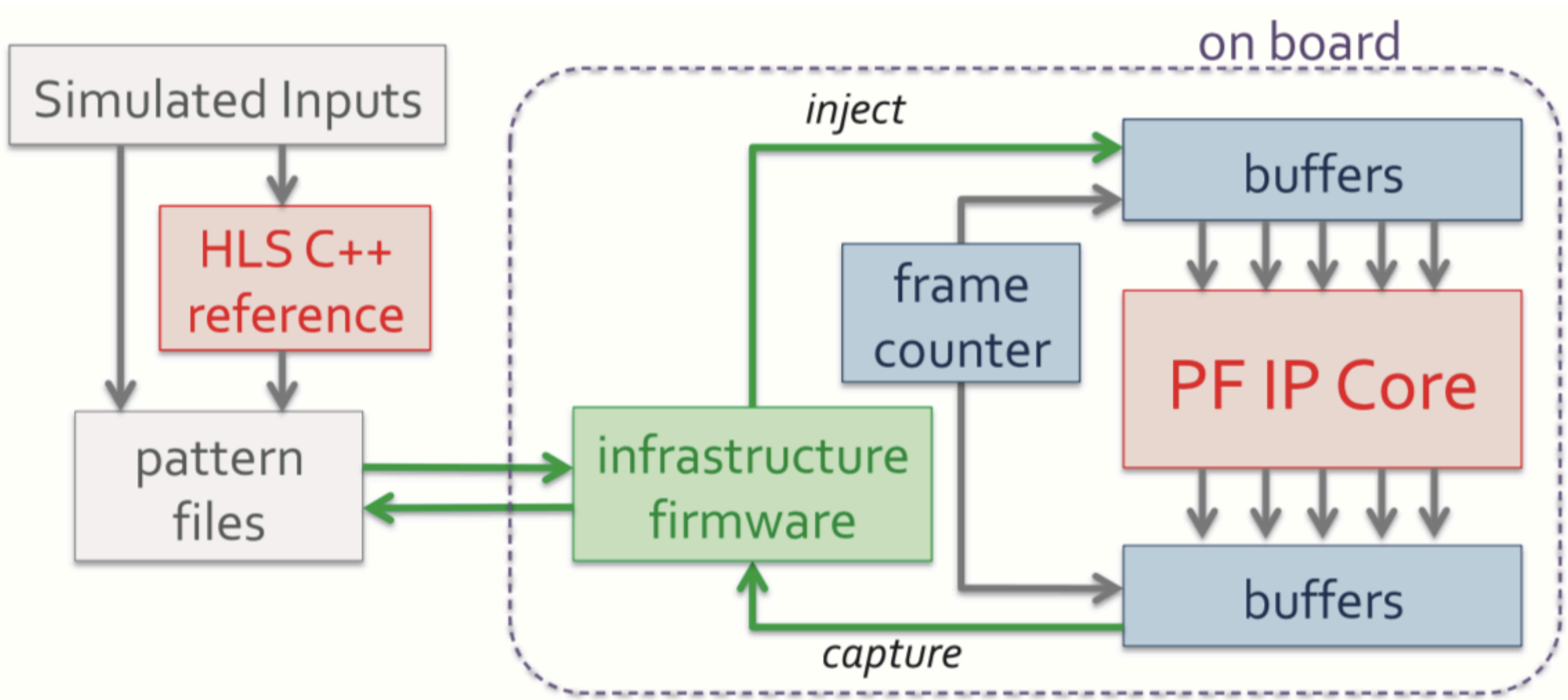
- The PF+PUPPI is inherently a **regional algorithm** → different detector regions can be processed independently and in parallel
  - complexity and FPGA resource use depend on the maximum allowed number of input objects, determined by the size of the detector region

Hardware demonstration

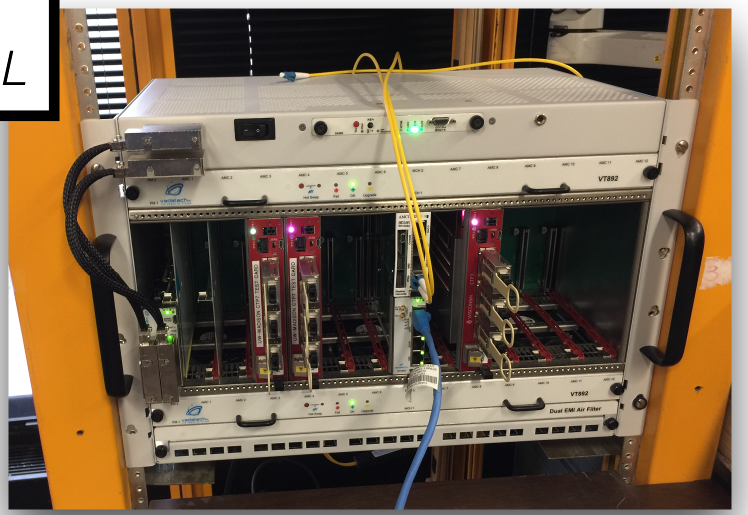
A proof-of-concept implementation running on current and early prototype trigger boards

based on Xilinx Virtex-7 FPGAs, VU9P with development kit and on Amazon AWS

Interface the core with the board infrastructure using IPbus or AXI-PCIe to **inject input patterns** from CMS detector simulation into the core, and **the output is checked for bitwise identity** with the expections from HLS.



Test stand @ LPC, FNAL

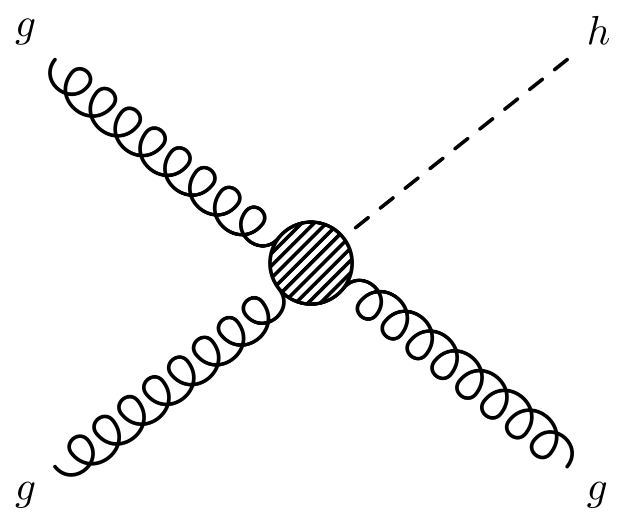


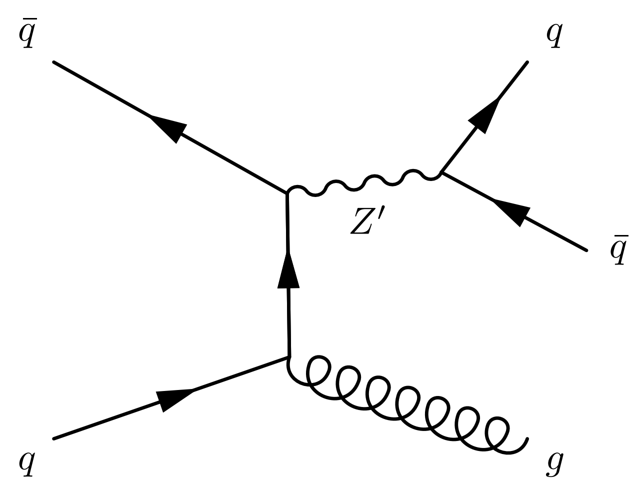
Test stand @ CERN

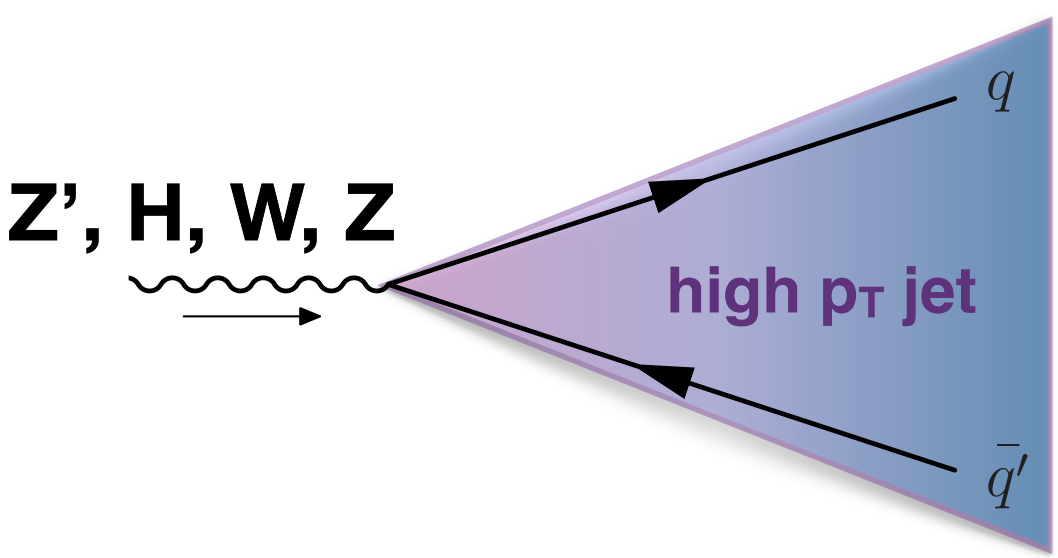
Summary and outlook

Bringing advanced physics algorithms to the hardware trigger!

Proof-of-concept for **PF+PUPPI running at L1**
Large physics gain: H_T, missing p_T, jet, lepton isolation







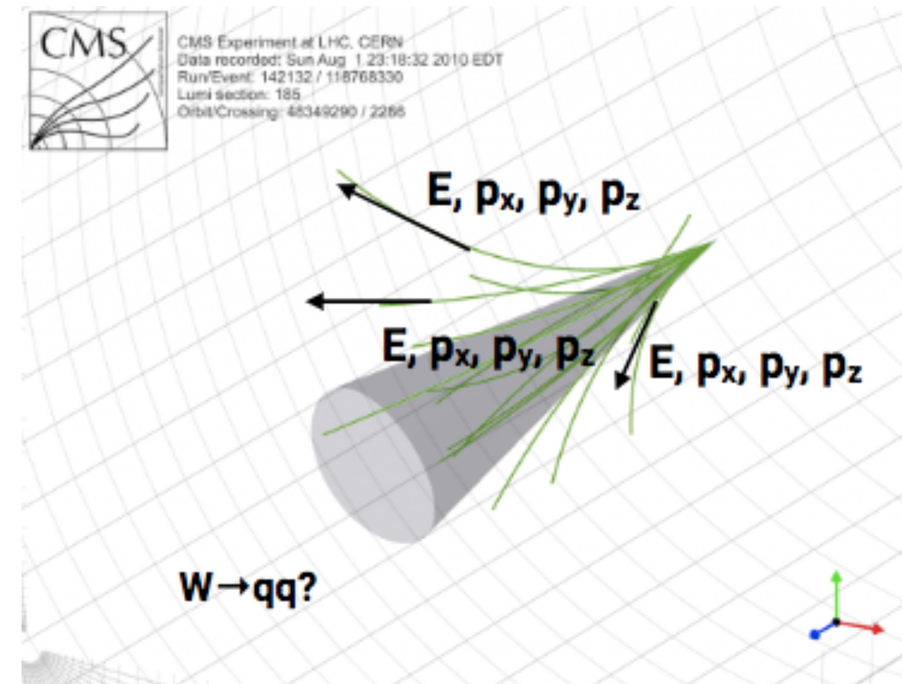
With the availability of particle candidates with PU mitigation @ L1, *can we trigger on these topologies at 25 ns?*

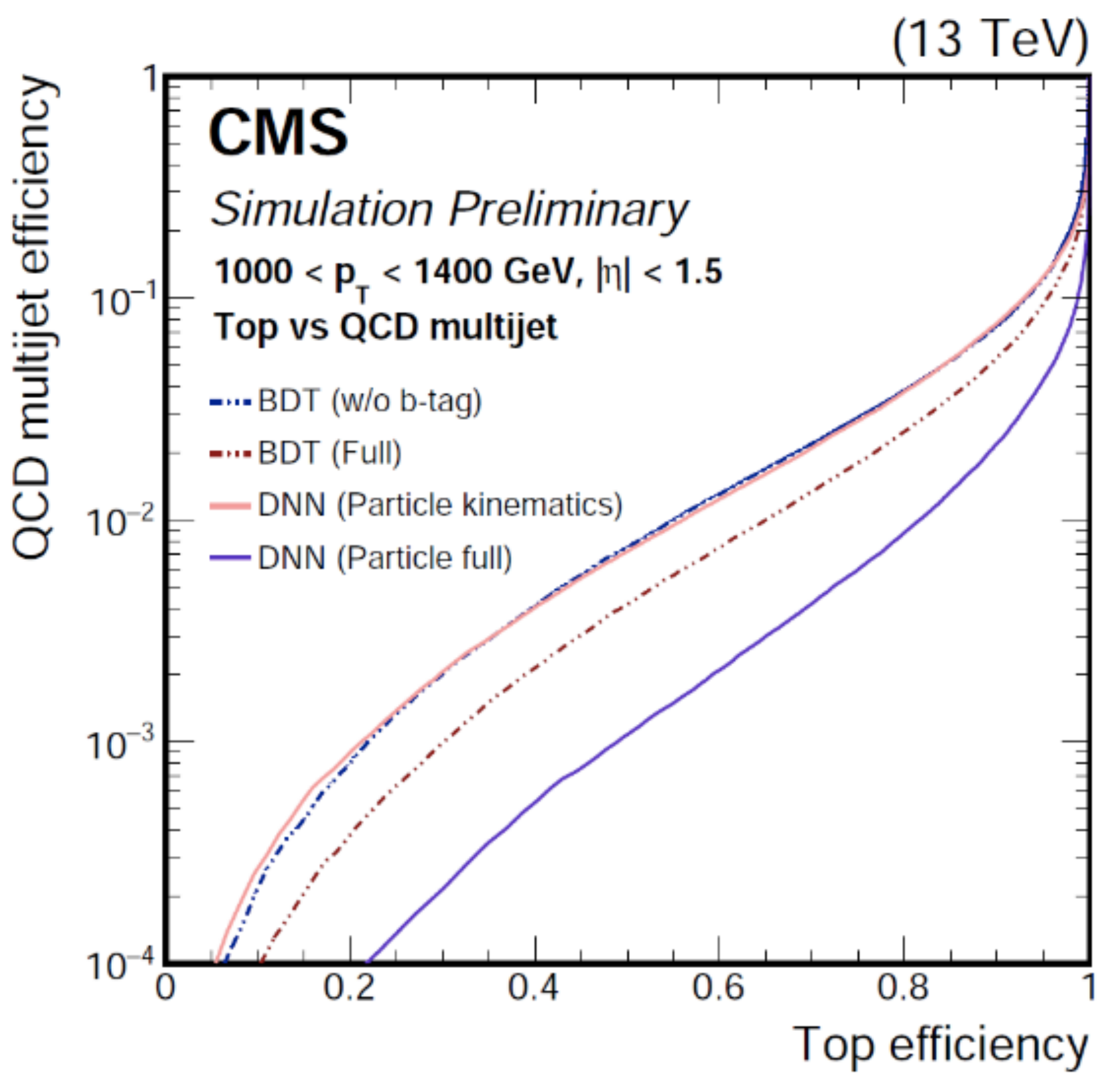
Large space here for further improvement in signal acceptance at high pileup!

How about machine learning?

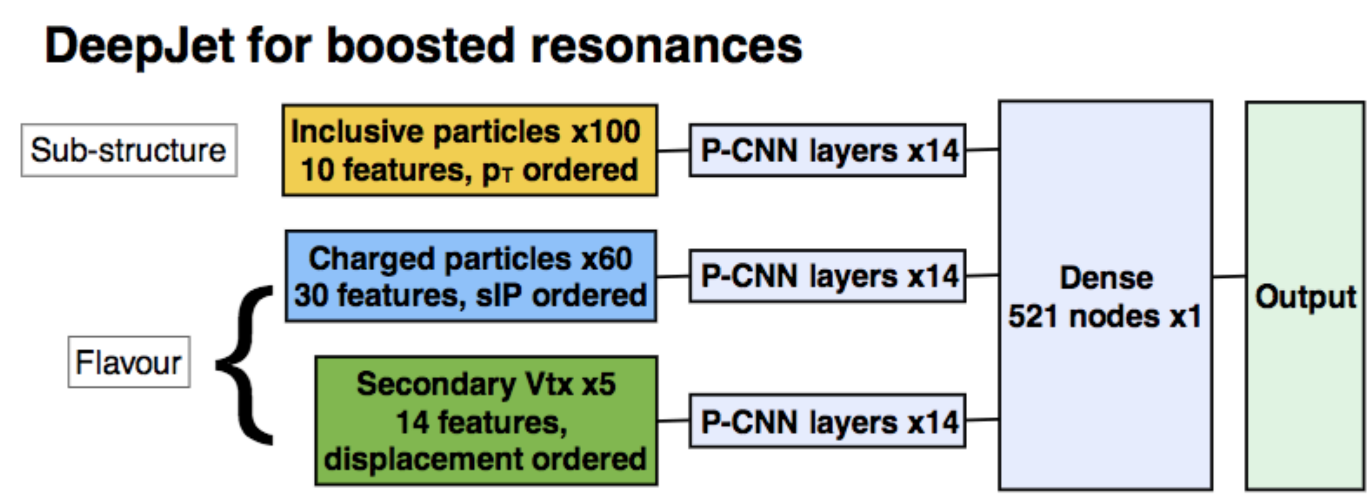
Many new ML algorithms developed for offline boosted-jet tagging using only kinematics of **particle candidates as input**, few examples:

Lola (G. Kasieczka et al. JHEP05(2017)006) → fully connected layers
DeepAK8 and double b-tagger (CMS-DP-2017-049)
→ one-dimensional convolutional layers





NN inference of such models possible on FPGA in L1 latency
(see talk on Thursday)



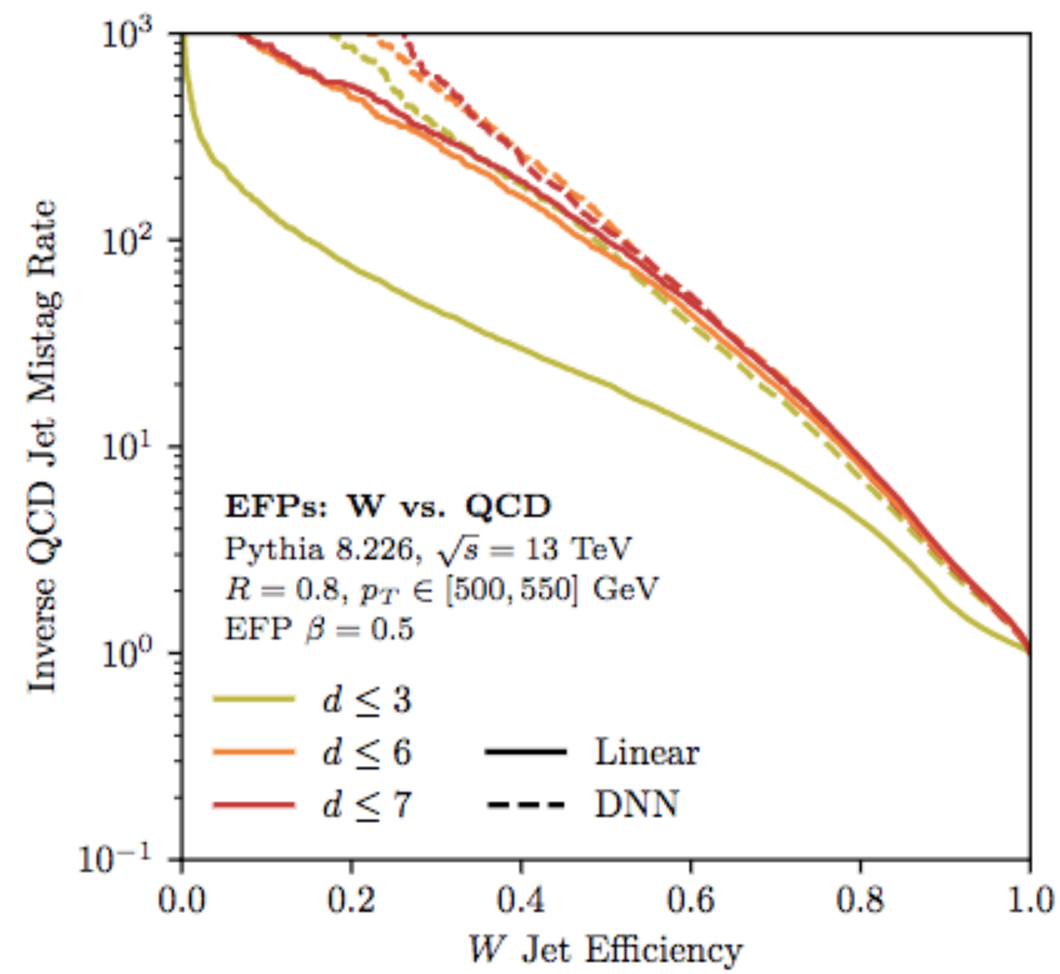
See L. Gouskos's talk on Wednesday

Towards FPGA-friendly jet algos

Energy flow polynomials: complete set of jet substructure observables forming a discrete linear basis for any common jet observables

$$\mathrm{EFP}_G = \sum_{i_1=1}^{M} \cdots \sum_{i_N=1}^{M} z_{i_1} \cdots z_{i_N} \prod_{(k,\ell)\in G} \theta_{i_k i_\ell}$$

M = # jet constituents
z = energy fraction
θ = angles



Combine EFPs to perform linear jet tagging or into a DNN
→ *mainly multiplications/additions on the FPGA*

See J. Thaler's talk on Wednesday

Towards FPGA-friendly jet algos

Anti-kT jet clustering algo preferred by theorists and good for offline reconstruction

sequential clustering not suitable for low latency on FPGA

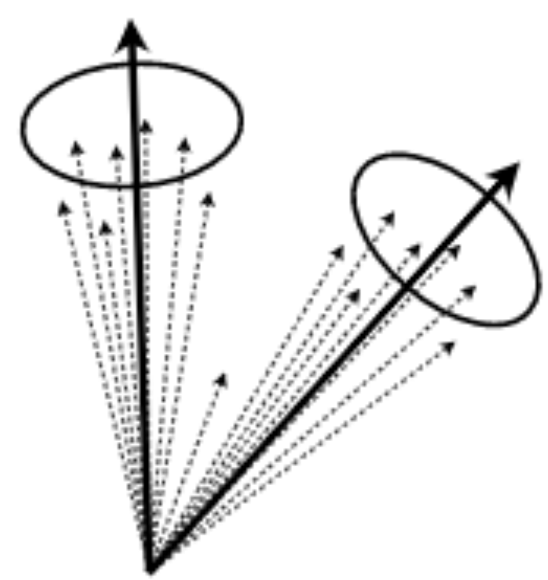
Explore more FPGA friendly algos for L1 applications with use of PF/PUPPI candidates

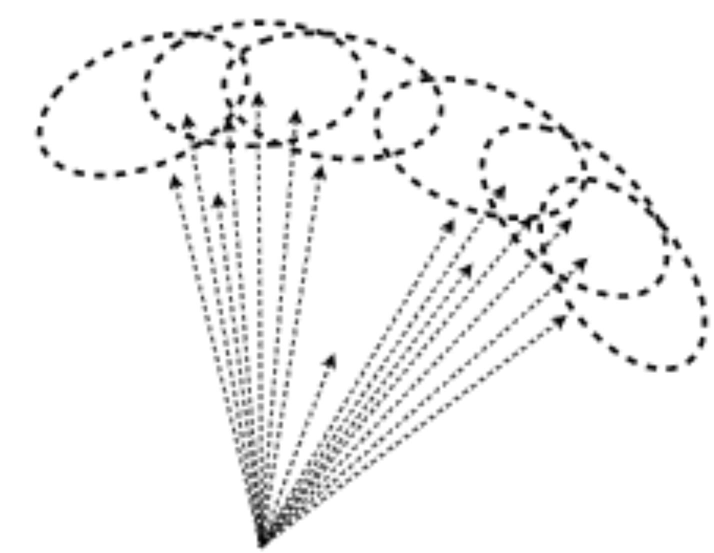
Jets without jets: use **local computation** to characterize jet-like and subjet-like structures w/o jet clustering algo

$$\widetilde{N}_{\rm jet}(p_{T{\rm cut}}, R) = \sum_{i\in{\rm event}} \frac{p_{Ti}}{p_{Ti,R}} \Theta(p_{Ti,R} - p_{T{\rm cut}}),$$

$$\widetilde{H}_T(p_{T{\rm cut}}, R) = \sum_{i\in{\rm event}} p_{Ti}\, \Theta(p_{Ti,R} - p_{T{\rm cut}}),$$

$$\widetilde{\not{p}}_T(p_{T{\rm cut}}, R) = \left| \sum_{i\in{\rm event}} \vec{p}_{Ti}\, \Theta(p_{Ti,R} - p_{T{\rm cut}}) \right|,$$





$$\widetilde{N}_{\rm subjet}(p_{T{\rm subcut}}, R_{\rm sub}) = \sum_{i\in{\rm jet}} \frac{p_{Ti}}{p_{Ti,R_{\rm sub}}} \Theta(p_{Ti,R_{\rm sub}} - p_{T{\rm subcut}}).$$

D. Bertolini et al., JHEP04(2014)013

Trigger:
fast decision

Particle flow and PUPPI:
advandanced reconstruction techniques

Fast PUPPI

Proof-of-principle studies indicate the feasibility of performing Particle Flow reconstruction and PUPPI pileup mitigation in the CMS HL-LHC Level-1 Trigger

Significant physics performance improvements over traditional trigger algorithms

For the first time, possibility to trigger on boosted jets at 25 ns with large gain for physics!

Trigger:
fast decision

Particle flow and PUPPI:
advandanced reconstruction techniques

Fast PUPPI

Backup

Implementation of Puppi proof-of-concept using High level synthesis (HLS) as well

COMPUTE FOR EACH NEUTRAL

[1] define a local discriminant, α, between pileup (PU) and leading vertex (LV)

$$\alpha_i^C = \log \left[\sum_{j \in \mathrm{Ch,LV}} \frac{p_{T,j}}{\Delta R_{ij}} \Theta(R_0 - \Delta R_{ij}) \right]$$

[2] get data-driven α distribution for PU using charged PU tracks

PRECOMPUTE STEP 2 OFFLINE WITH CONSTANTS (FOR GIVEN PILEUP LEVEL)

[3] for the neutrals, ask "how un-PU-like is α for this particle?", compute a weight

DO STEP 3/4 WITH A LOOK-UP TABLE

[4] reweight the four-vector of the particle by this weight, then proceed to interpret the event as usual

RESOURCE USAGE ONLY FEW % OF FPGA AND 100S OF NS LATENCY WITH LITTLE DEGRADATION IN PERFORMANCE

Input data size to the correlator

Table 5.1: Summary of prototype logical input data to the CT.

Input	Object	N bits/object	N objects	N bits/BX	Total BW (Gb/s)
Tracker	Track	100	900	90 000	3 600
Barrel Calo	Cluster	16	2 448	39 168	1 567
Barrel Calo	Tower	32	612	19 584	783
HF	Tower	10	1 440	14 440	553
Endcap Calo	Cluster	128	400	51 200	1 600
Endcap Calo	Tower	16	2 400	38 400	1 536
Barrel Muon	Track	64	36	2 304	92
Endcap Muon	Track	64	36	2 304	92
Total					8 547