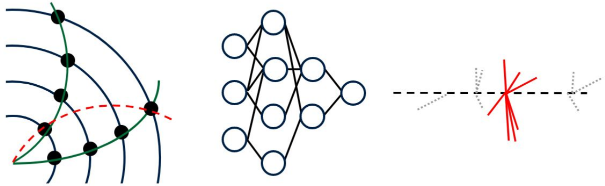
Imperial College London

# Track and Vertex Finding for the CMS Level-1 Trigger

**Christopher Brown**

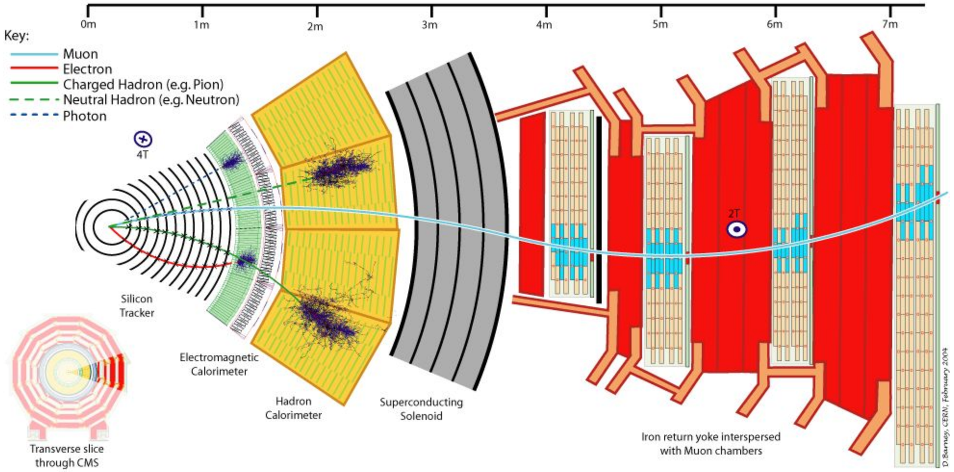
*on behalf of the CMS Collaboration*

31st May 2022



# Current Era CMS

- LHC **40 MHz** bunch crossing rate, need to select events based on **physics potential**, can't store everything
- Two-stage trigger
  - Level - 1 hardware based trigger, quick **partial event reconstruction**, **100 kHz** output, **< 4 μs** latency. Only muon and calorimeter data
  - High level trigger, **full event reconstruction** with full granularity detector data with all parts, **1 kHz** output, CPU farm

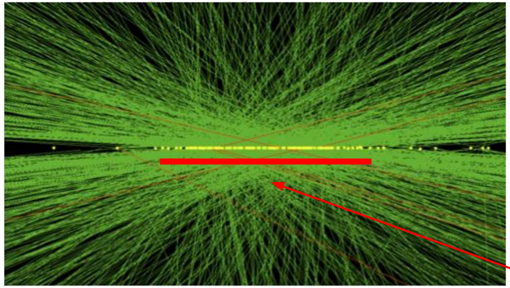



c.brown19@imperial.ac.uk

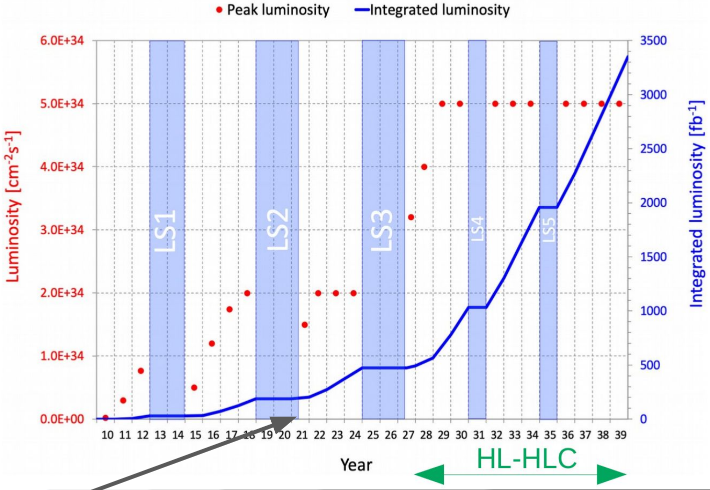
# High Luminosity LHC

- HL-LHC -> expected to deliver 3000 fb$^{-1}$
- Good for **rare physics searches** and **precision measurements** of SM
- Will see increased number of simultaneous proton-proton interactions per bunch crossing (pile up PU).
- **High PU** (up to 200) bad for current era triggering
- Level-1 trigger in HL-LHC rate would be 4 MHz to maintain current physics sensitivity, new trigger needed for HL-LHC utilising tracker tracks for the first time

High pile up HL-LHC



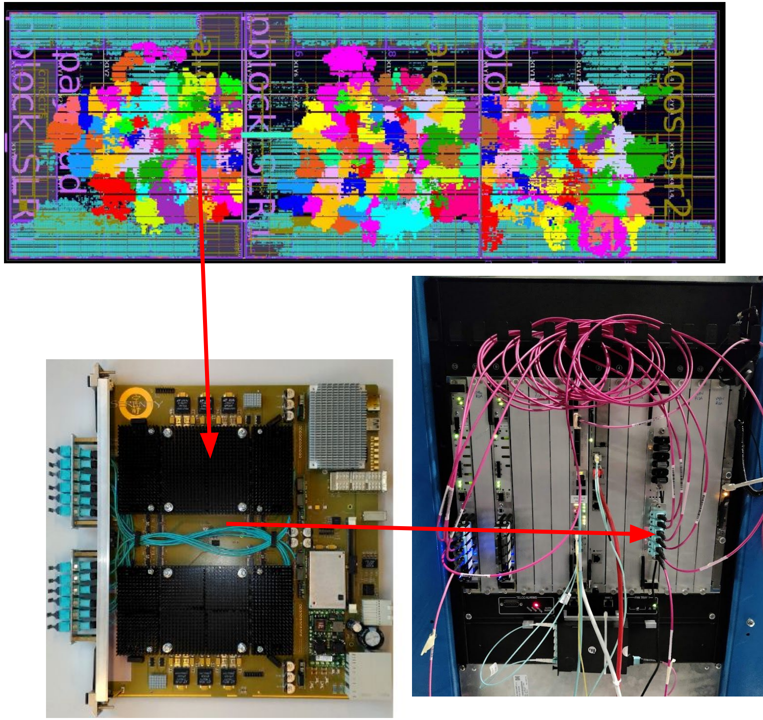
~ 10 cm

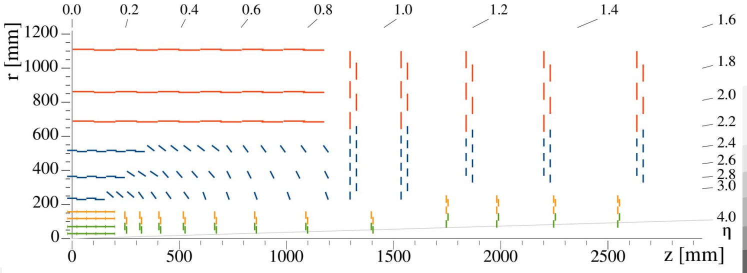



We are here: start of run 3

# CMS Phase-2 Upgrade

- Extensive upgrade program to all parts of the detector, new all-FPGA L1 trigger running at **750 kHz** with increased latency to **12.5 µs** -> more complex algorithms possible
- All new tracker, larger η (up to 3.8) coverage with inner tracker
- **Tracker tracks** for the first time at L1 trigger -> full 40 MHz readout η < 2.4 with outer tracker
- Track finding and L1 trigger implemented on Xilinx Ultrascale+ FPGAs, **latency and resource usage** of every algorithm critical

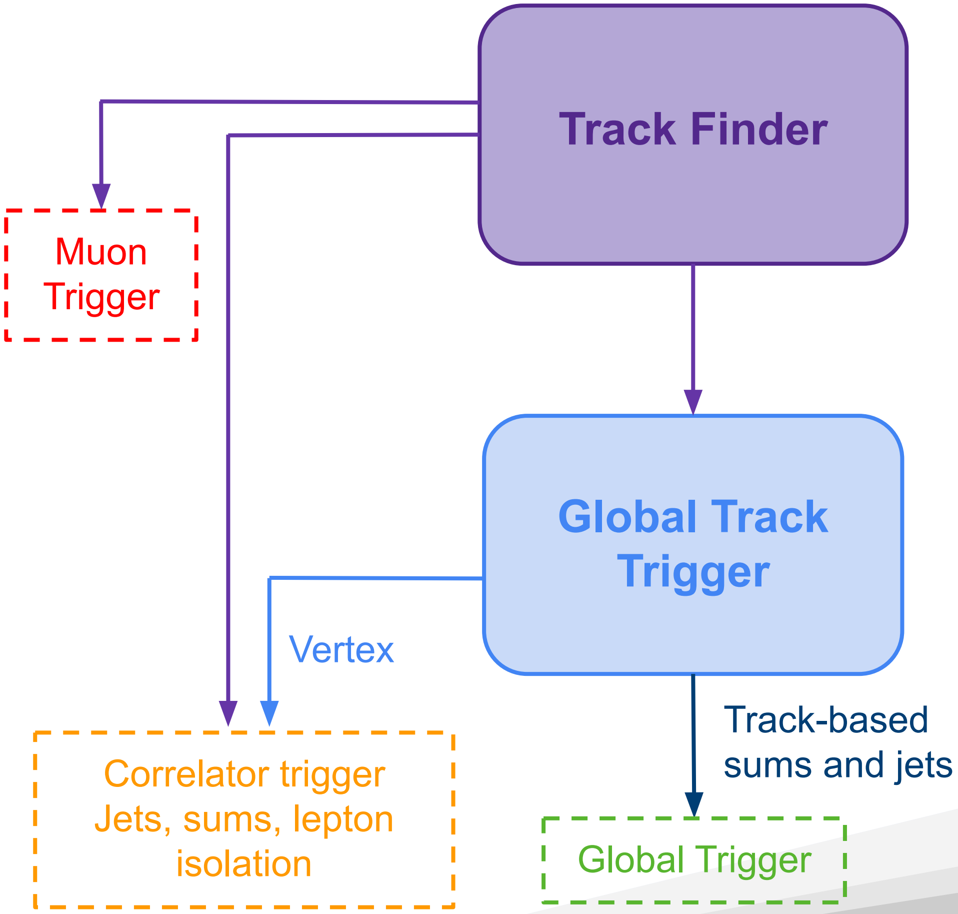





c.brown19@imperial.ac.uk

# Using tracks in L1 Trigger

- Muon to **tracker track matching** in Muon trigger
- **Primary vertex finding**, Track based $E_T^{miss}$ and track based Jet finding in global track trigger
- Vertex + **associated tracks**
  - **Cleaner** energy sums
  - Better lepton isolation
  - **Complex algorithms** possible with reduced inputs



**Tracker Inputs**

**Track Finder**

Tracklet Road Search

Kalman Filter

Track Quality

**Global Track Trigger**

Baseline Approach

Improved Baseline

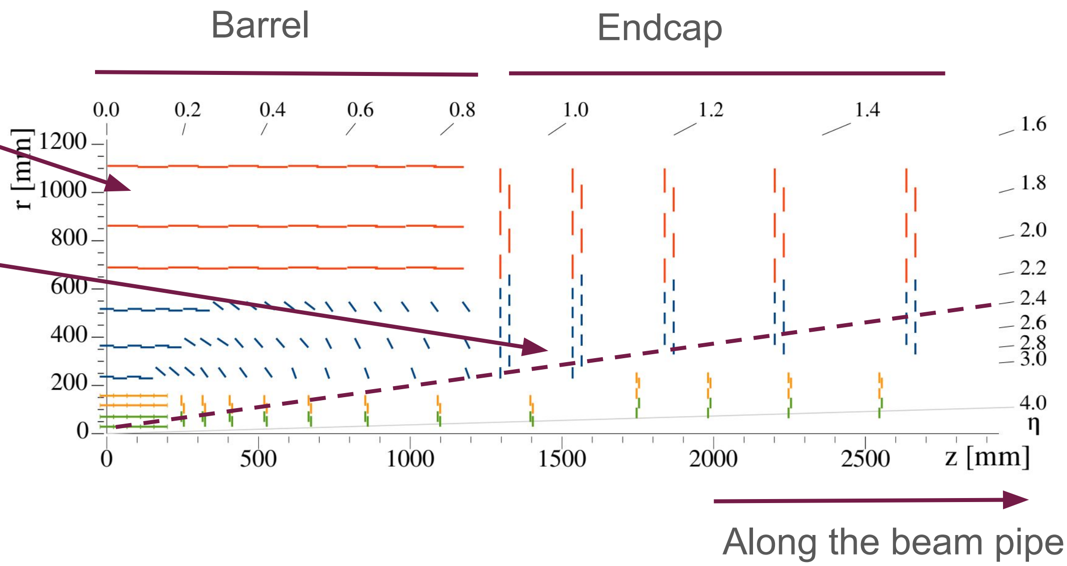
End-to-end NN approach

Firmware Implementation

**Demonstration**

# Track Finder Inputs

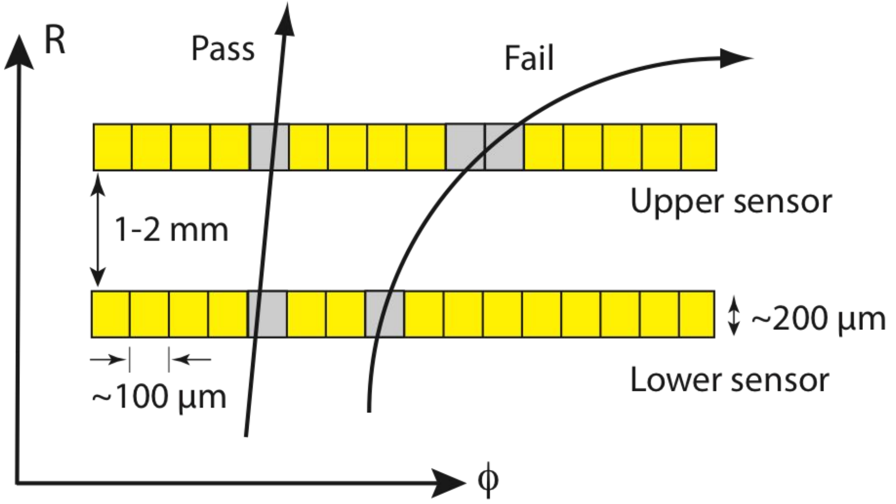
- **Outer tracker** only, 6 barrel layers, 5 endcap layers in red and blue
- Online Track Finding **|η| < 2.4**
- Combinatorics too large to consider every detector hit even from outer tracker

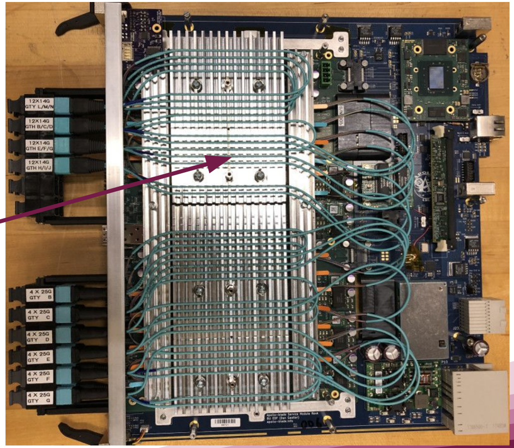



c.brown19@imperial.ac.uk

# Track Finder Inputs

- **$p_T$ modules** -> 2 closely spaced detector layers
  - **Tunable** on-detector $p_T$ cut
  - **10x-20x** reduction in data
  - **Online track finding** possible
- **> 15k stubs** per bunch crossing **$p_T$ > 2 GeV**, bunch crossing rate **40 MHz**
- **~ 200 tracks** $p_T$ > 2 GeV per crossing to reconstruct in **4 µs**
- Exploit **parallelism** and **regional division** of outer tracker, multiple copies of track finding algorithm on 162 boards



c.brown19@imperial.ac.uk

**Tracker Inputs**

**Track Finder**

Tracklet Road Search

Kalman Filter

Track Quality

**Global Track Trigger**

Baseline Approach

Improved Baseline

End-to-end NN approach

Firmware Implementation

**Demonstration**

# Hybrid Track Finding Algorithm

## Tracklet Road Search

- Form track candidates

## Track Fitting

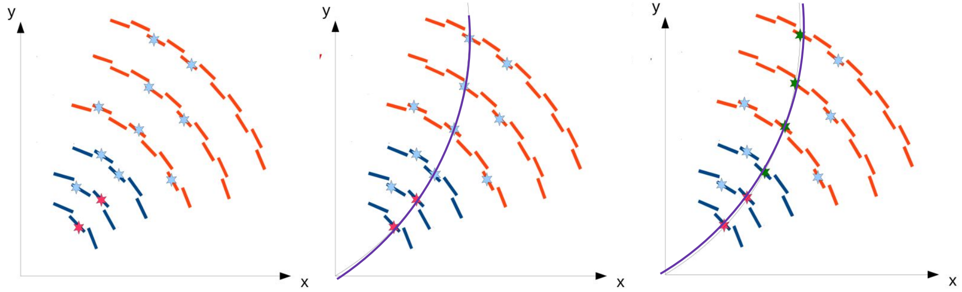
- Combinatorial Kalman Filter

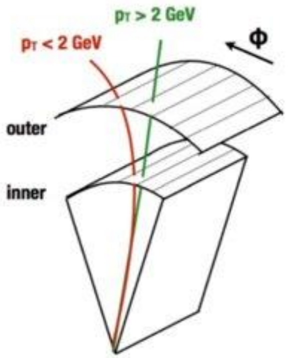
## Track Quality

- Calculate $\chi^2$ from KF residuals or use a BDT

# Tracklet Road Search

- Find stubs in adjacent layers, **tracklet seeds**
- Create track candidate from tracklet seed and **project to other layers**
- Find stubs along projection and add to track candidate



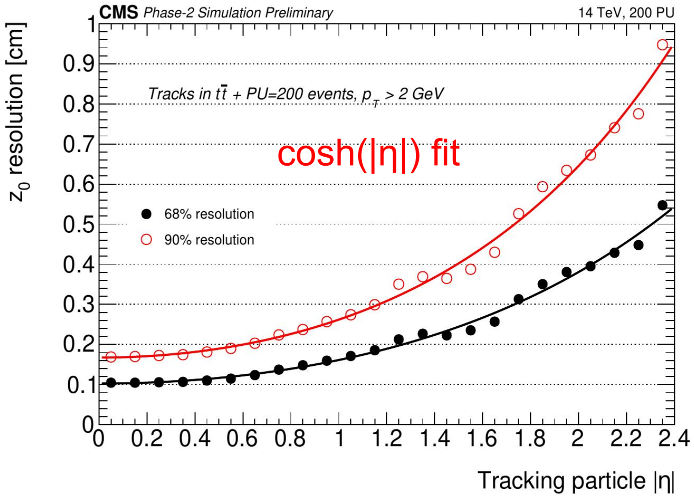



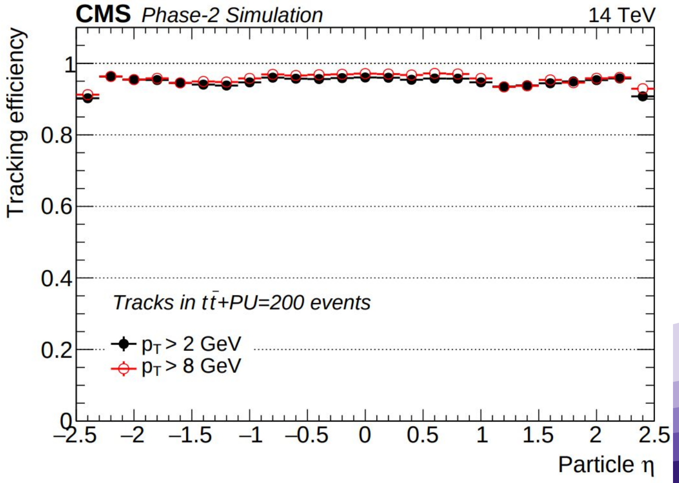

- Huge combinatorics -> **15k stubs**, can't consider all of them
- Split every tracker region into further slices
- Only some stubs are compatible with inner and outer slices so reduce number of candidates
- **8 different combinations** of layers are used to form tracklet seeds -> good η efficiency with latency and resource usage within budget

c.brown19@imperial.ac.uk

# Track Fit - Kalman Filter

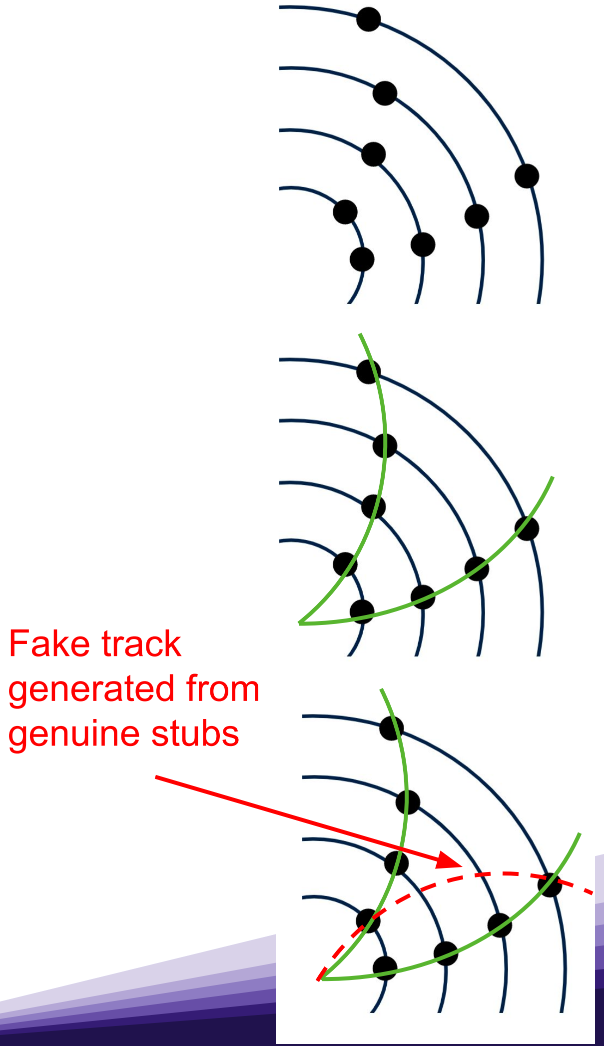
- Start with track candidate from tracklet stage and iteratively add associated stubs **updating track parameters and fit**
- Kalman Filter written for FPGA
- Complete within **1 μs**
- Final step to package tracks into **96-bit track word** and route in η for rest of trigger



# Track Quality

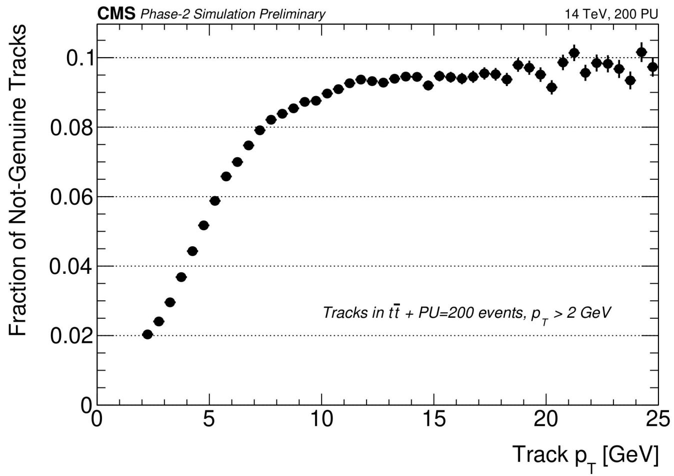
- Not genuine or ‘**fake**’ track not matched to a monte carlo event generated track based on detector hit matching
- Represent a **significant fraction** of produced tracks at high $p_T$
- Issue for downstream algorithms
- Extra **$\chi^2$ cuts** performed downstream give handle on fake tracks



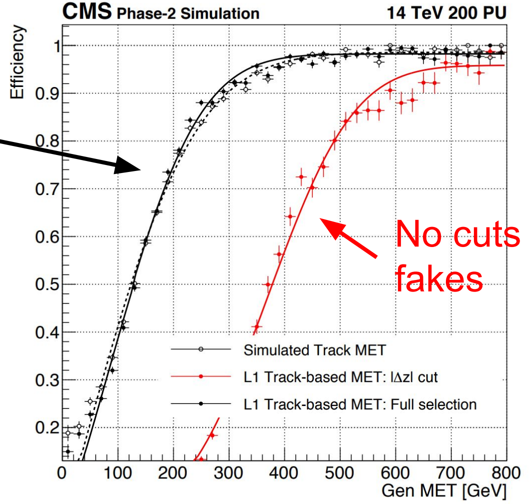

c.brown19@imperial.ac.uk

# Track Quality

- Not genuine or ‘**fake**’ track not matched to a monte carlo event generated track based on detector hit matching
- Represent a **significant fraction** of produced tracks at high $p_T$
- Issue for downstream algorithms
- Extra **$\chi^2$ cuts** performed downstream give handle on fake tracks




$\chi^2$ cuts on fakes

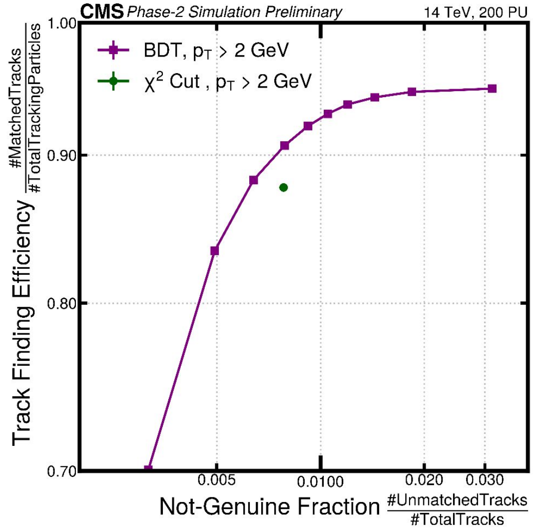


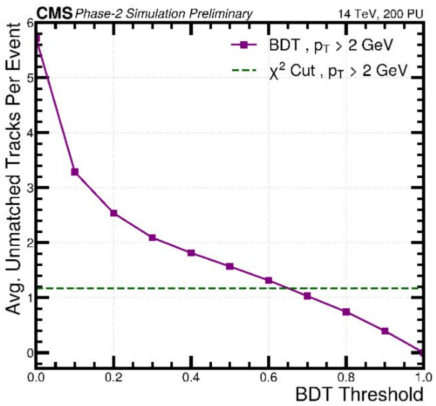

c.brown19@imperial.ac.uk

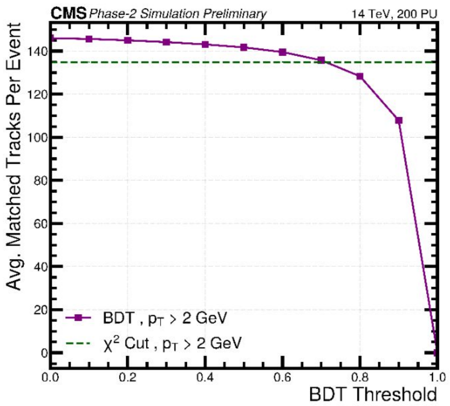
# Track Quality - Boosted Decision Trees

- Trained **BDT** on track features:

  ($\phi$, $\eta$, $z_0$ , $\chi^2_{bend}$ , #stubs, #missing layers $_{interior}$ , $\chi^2_{r\phi}$ , $\chi^2_{rz}$ )
- Lightweight BDT, **depth of 3** with **60 iterations**
- **Outperforms** additional strict $\chi^2$ cuts used in downstream trigger
- Implemented in firmware, completes inference within **33 ns**, small fraction ( < 1%) of total FPGA resource usage




c.brown19@imperial.ac.uk

**Tracker Inputs**

**Track Finder**

Tracklet Road Search

Kalman Filter

Track Quality

**Global Track Trigger**

Baseline Approach

Improved Baseline

End-to-end NN approach

Firmware Implementation

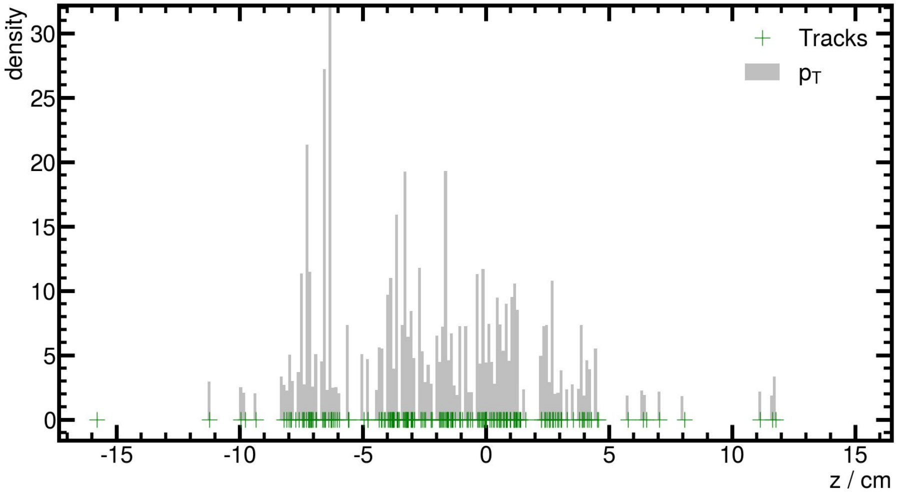
**Demonstration**

# Baseline Vertex Finding Chain

Track Finding

Produces $O(100)$ tracks per event **> 2 GeV**, with PU 200

Along the beam pipe




c.brown19@imperial.ac.uk

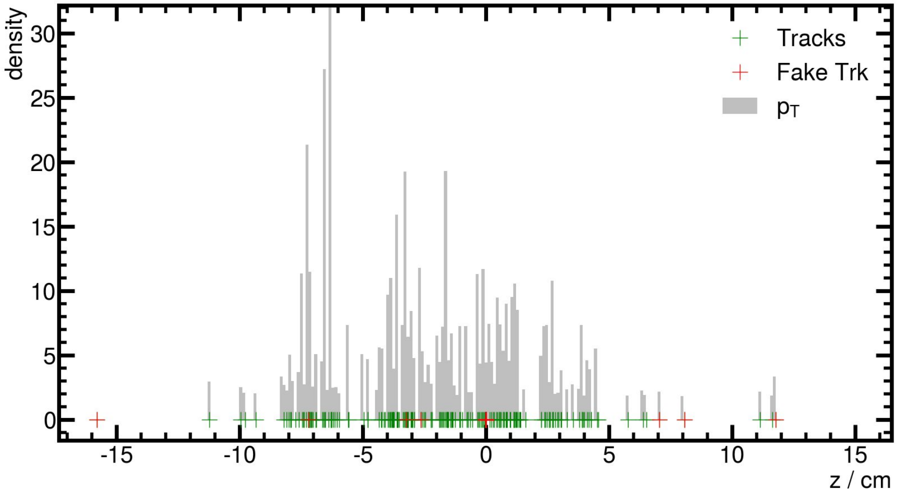
# Baseline Vertex Finding Chain

Track Finding

Produces *O*(100) tracks per event **> 2 GeV**, with PU 200

Track Quality

Based on **$\chi^2$ parameters** from track finding, simple cuts



# Baseline Vertex Finding Chain

Track Finding

Produces *O*(100) tracks per event **> 2 GeV**, with PU 200

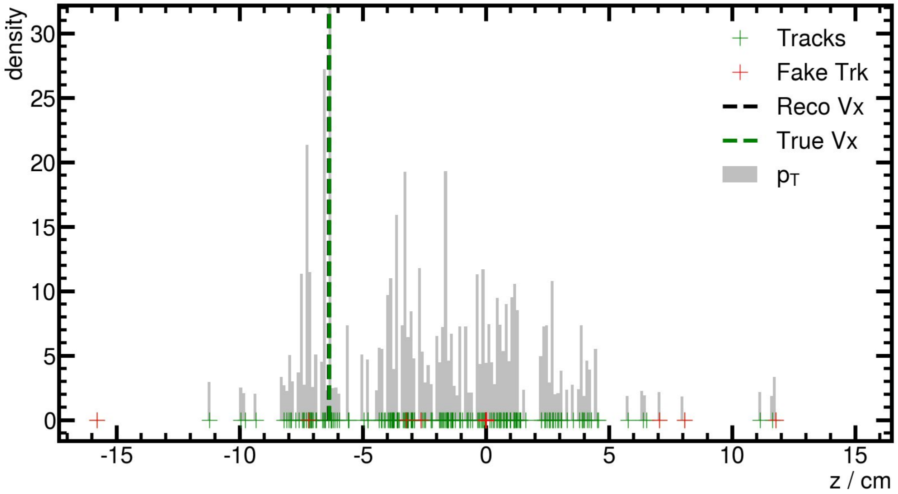
Track Quality

Based on **$\chi^2$ parameters** from track finding, simple cuts

Vertex Finding

Histogram all tracks in $z_0$ **weighted by $p_T$**, find **3 consecutive bins** with highest $p_T$

Good vertex reconstruction due to high $p_T$ peak




c.brown19@imperial.ac.uk

# Baseline Vertex Finding Chain

Based on track $z_0$ resolution

| $\eta$ range | $\|\Delta z\,(z_{PV}, z_{trk})\|$ (cm) |
|---|---|
| $0 \leq \|\eta\| < 0.7$ | 0.4 |
| $0.7 \leq \|\eta\| < 1.0$ | 0.6 |
| $1.0 \leq \|\eta\| < 1.2$ | 0.76 |
| $1.2 \leq \|\eta\| < 1.6$ | 1.0 |
| $1.6 \leq \|\eta\| < 2.0$ | 1.7 |
| $2.0 \leq \|\eta\| < 2.4$ | 2.2 |

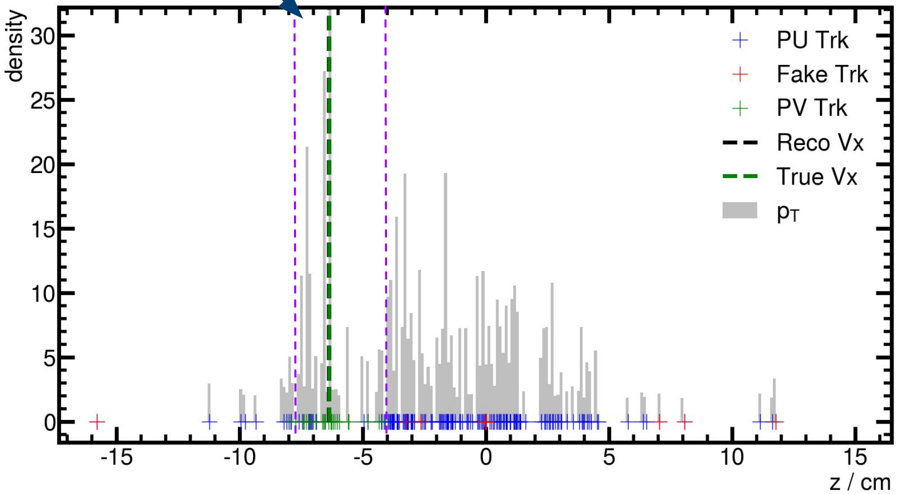
**Track Finding** — Produces $O(100)$ tracks per event **> 2 GeV**, with PU 200

**Track Quality** — Based on **$\chi^2$ parameters** from track finding, simple cuts

**Vertex Finding** — Histogram all tracks in $z_0$ **weighted by $p_T$**, find **3 consecutive bins** with highest $p_T$

**Track to Vertex Association** — Fixed **window in $z_0$** or multiple windows based on track η

Association window

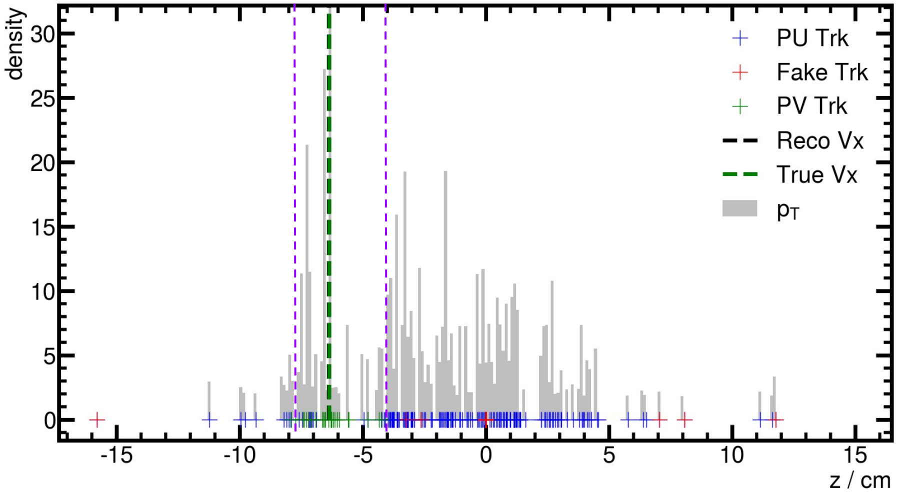



c.brown19@imperial.ac.uk

# Baseline Vertex Finding Chain

| | |
|---|---|
| Track Finding | Produces $O(100)$ tracks per event **> 2 GeV**, with PU 200 |
| Track Quality | Based on **$\chi^2$ parameters** from track finding, simple cuts |
| Vertex Finding | Histogram all tracks in $z_0$ **weighted by $p_T$**, find **3 consecutive bins** with highest $p_T$ |
| Track to Vertex Association | Fixed **window in $z_0$** or multiple windows based on track η |
| Track $E^T_{Miss}$ PF/PUPPI etc. | Downstream Algorithms |

| $\eta$ range | $\lvert\Delta z\,(z_{PV}, z_{trk})\rvert$ (cm) |
|---|---|
| $0 \le \lvert\eta\rvert < 0.7$ | 0.4 |
| $0.7 \le \lvert\eta\rvert < 1.0$ | 0.6 |
| $1.0 \le \lvert\eta\rvert < 1.2$ | 0.76 |
| $1.2 \le \lvert\eta\rvert < 1.6$ | 1.0 |
| $1.6 \le \lvert\eta\rvert < 2.0$ | 1.7 |
| $2.0 \le \lvert\eta\rvert < 2.4$ | 2.2 |



# Vertex Finding Concept

**Baseline**

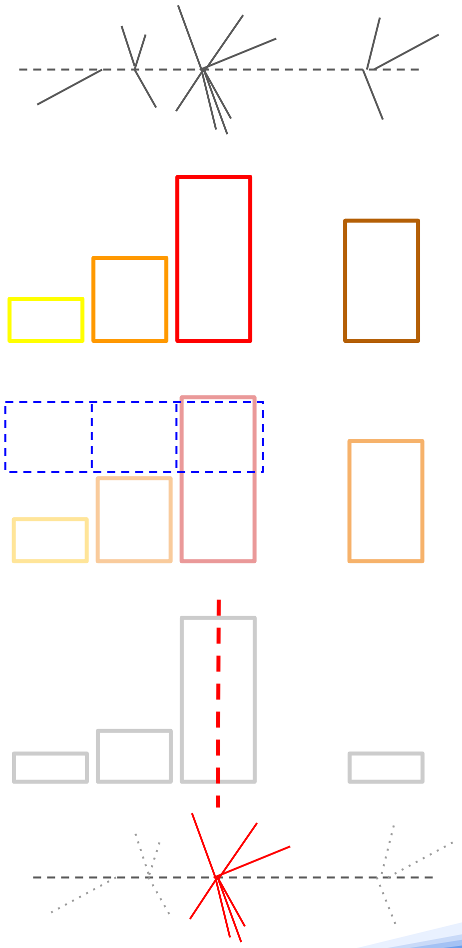
$p_T$ Weighting

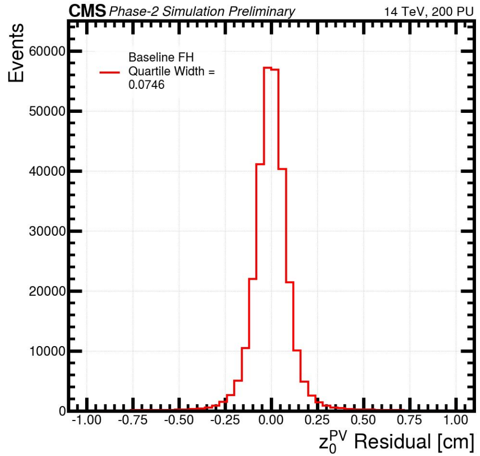
Weighted Histogram

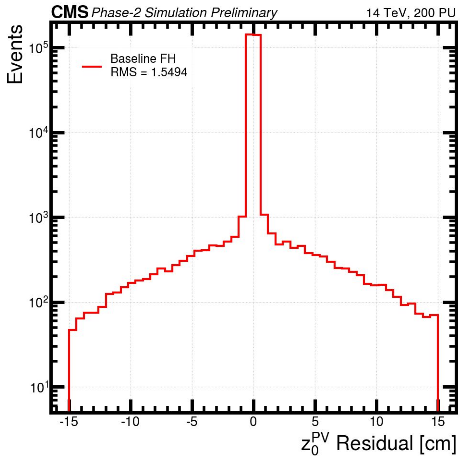
3-Bin Convolution

Peak Finder

$z_0$ window






c.brown19@imperial.ac.uk

# Vertex Finding Concept

**Baseline**

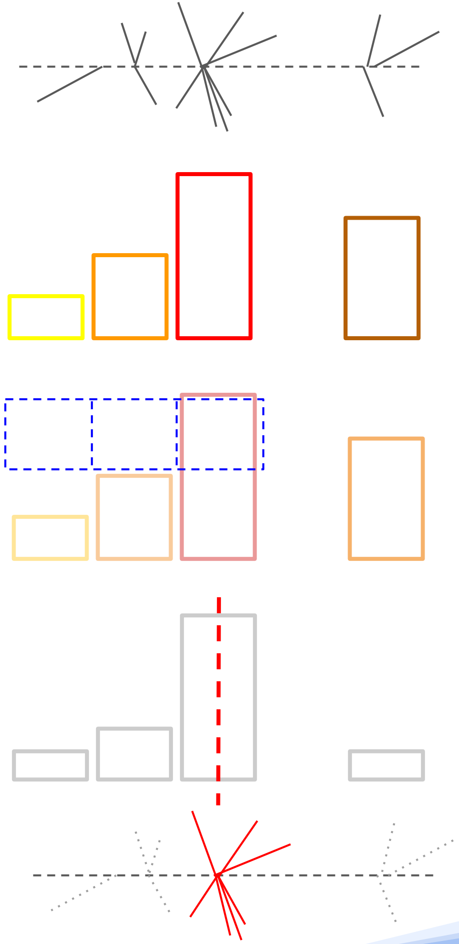
$p_T$ Weighting

Weighted Histogram

3-Bin Convolution

Peak Finder

$z_0$ window



**End to End Neural Network**

DNN multiple track features (η,BDT,$p_T$)

c.brown19@imperial.ac.uk

# Vertex Finding Concept

**Baseline**

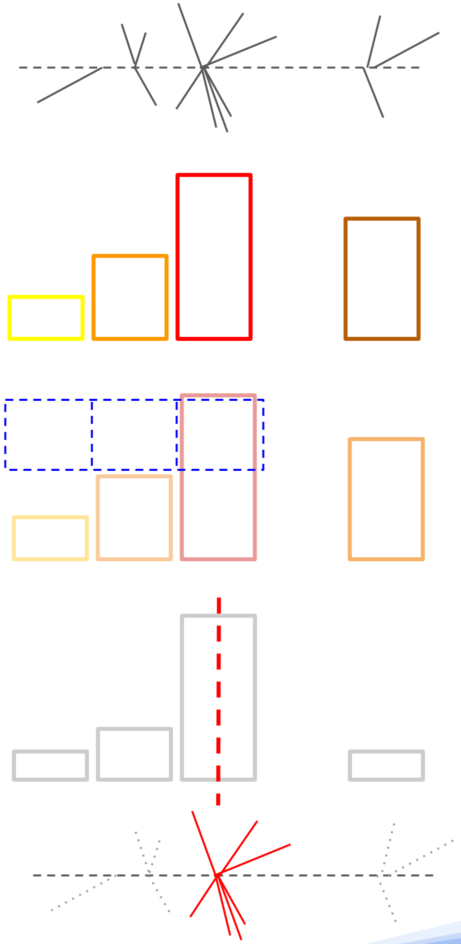
$p_T$ Weighting

Weighted Histogram

3-Bin Convolution

Peak Finder

$z_0$ window



**End to End Neural Network**

DNN multiple track features ($\eta$,BDT,$p_T$)

Weighted Histogram

c.brown19@imperial.ac.uk

# Vertex Finding Concept

**Baseline**

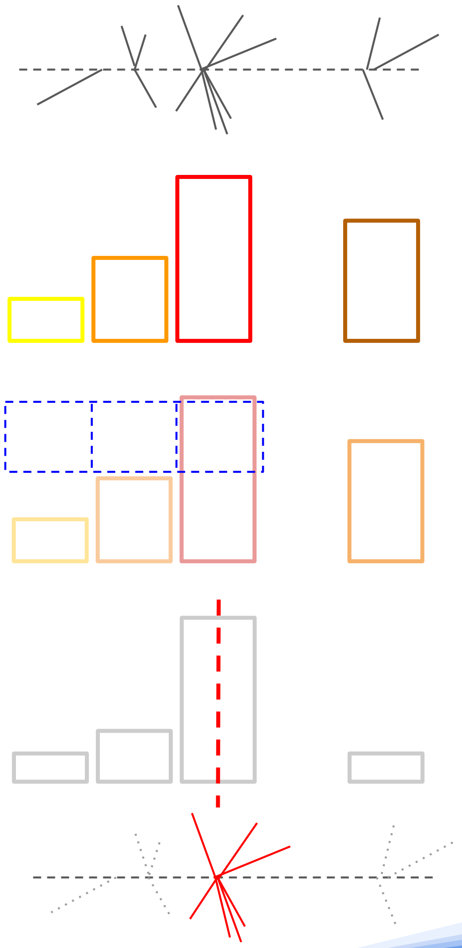
$p_T$ Weighting

Weighted Histogram

3-Bin Convolution

Peak Finder

$z_0$ window



**End to End Neural Network**

DNN multiple track features (η,BDT,$p_T$)

Weighted Histogram

Multilayered CNN

c.brown19@imperial.ac.uk

# Vertex Finding Concept

**Baseline**

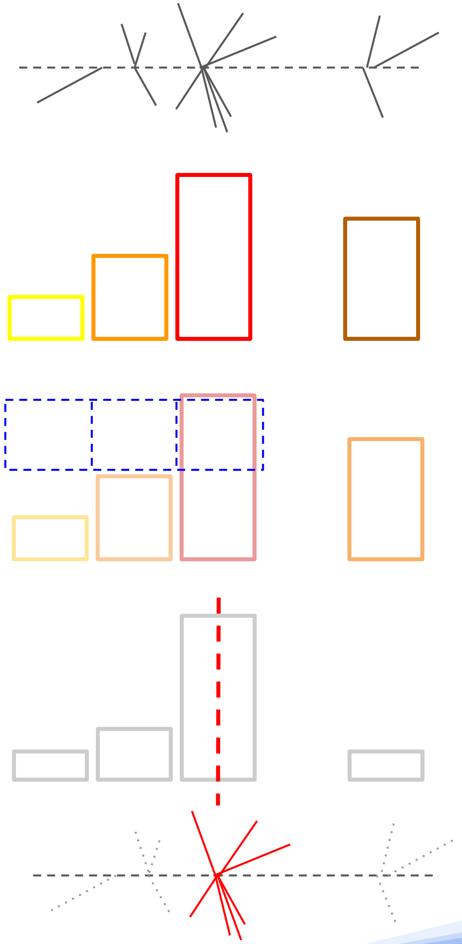
$p_T$ Weighting

Weighted Histogram

3-Bin Convolution

Peak Finder

$z_0$ window



**End to End Neural Network**

DNN multiple track features ($\eta$,BDT,$p_T$)

Weighted Histogram

Multilayered CNN

Peak Finder

# Vertex Finding Concept

**Baseline**

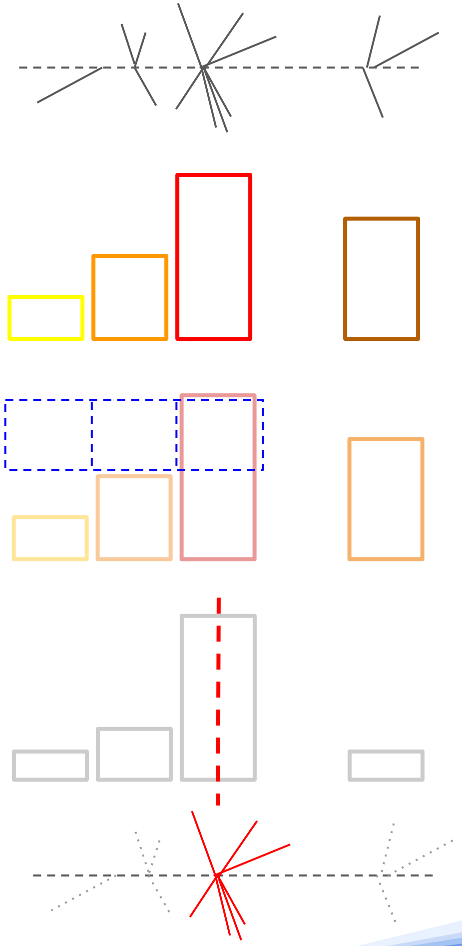
$p_T$ Weighting

Weighted Histogram

3-Bin Convolution

Peak Finder

$z_0$ window



**End to End Neural Network**

DNN multiple track features ($\eta$,BDT,$p_T$)

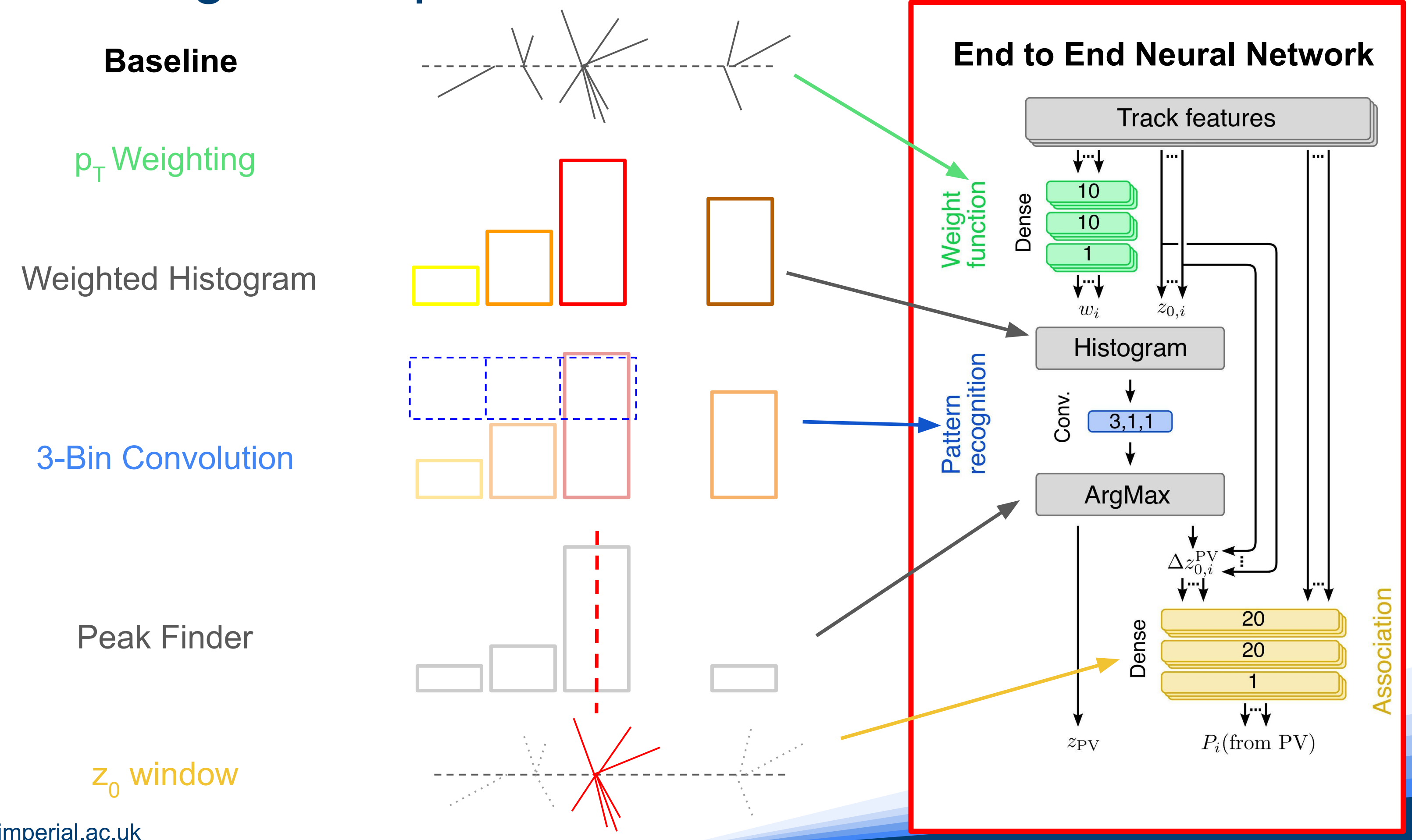
Weighted Histogram

Multilayered CNN

Peak Finder

DNN with $z_0$ distance, track features and latent features

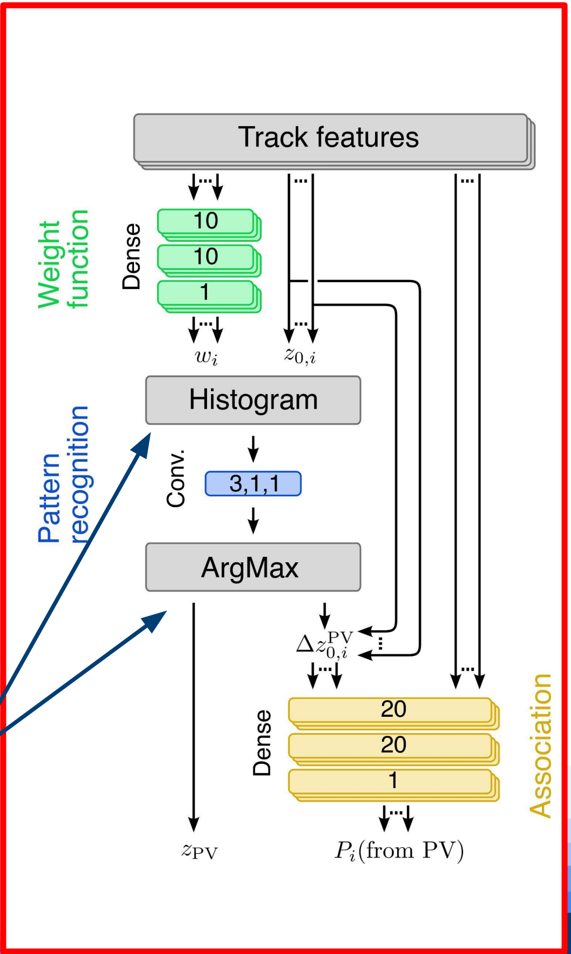
# Vertex Finding Concept



# End to End Neural Networks for Vertex Finding

- Network trained with **2 part loss function** -> Event level PV **regression**, track level PV track **classification**
- End-to-end -> track to vertex association optimised, influences vertex regression
- **1000** parameter network, all parts trained in 1 cycle
- Robust to changes in track finding
- Additional **vertex quality**

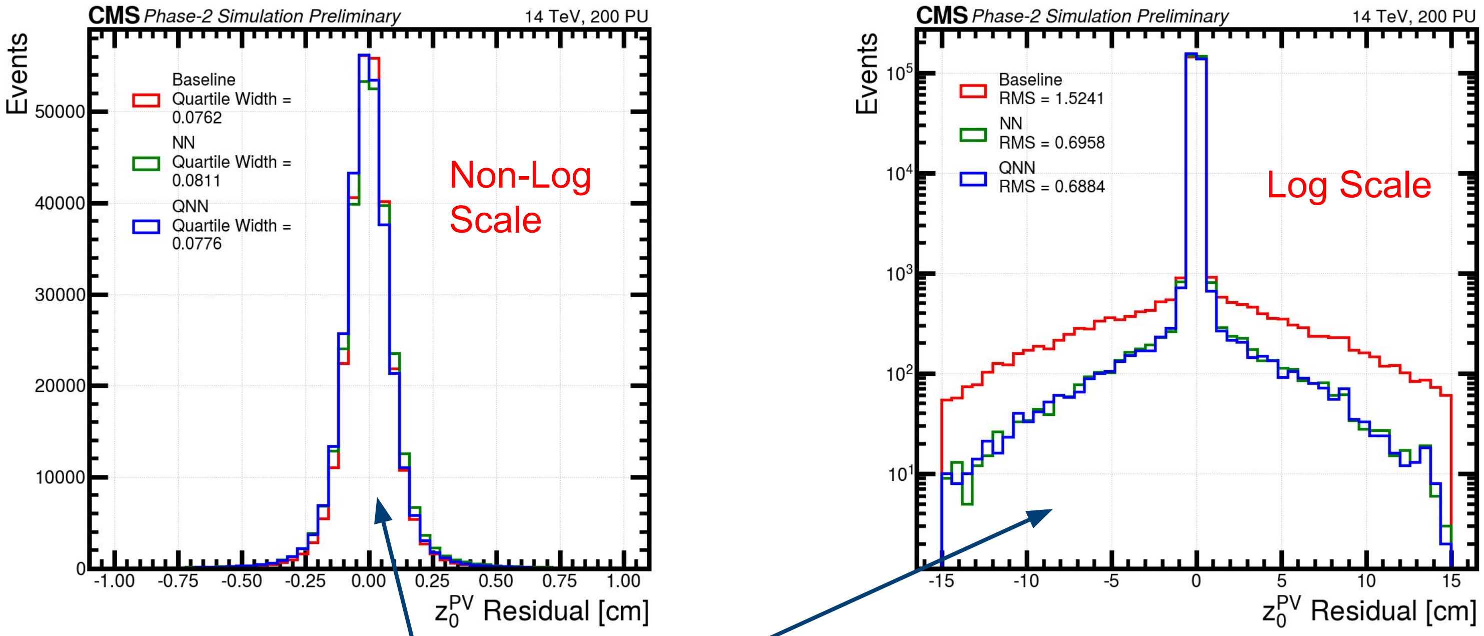
Differentiable




c.brown19@imperial.ac.uk

# Performance - Vertex Regression

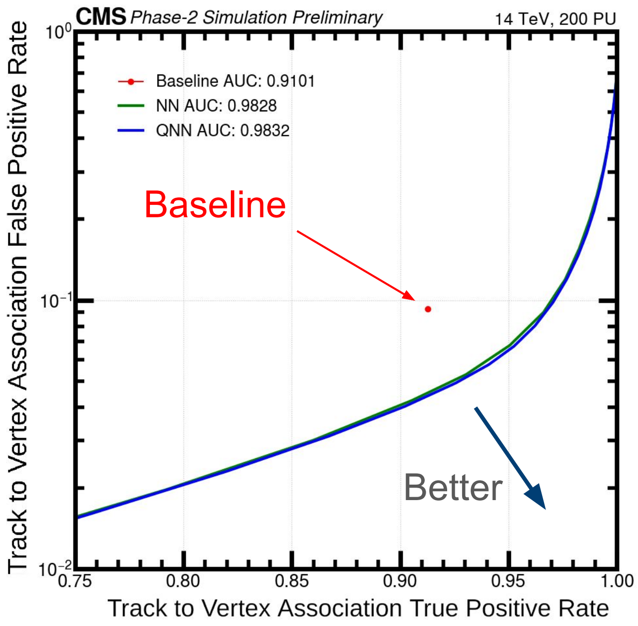
**QNN** compressed networks, see later…

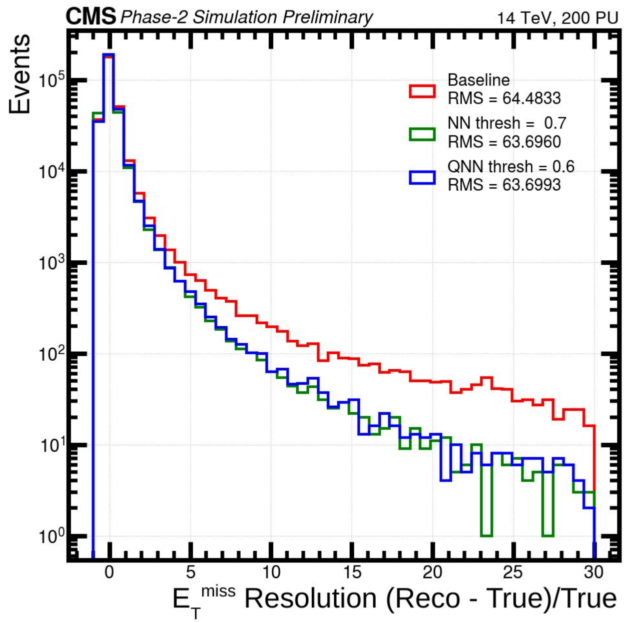



- Similar performance in core of residual
- **55% reduction** in tails of residual
- Better **identification of pileup vertices** removing high $p_T$ clusters
- Similar performance with compressed networks

c.brown19@imperial.ac.uk

# Performance - Track to Vertex Association



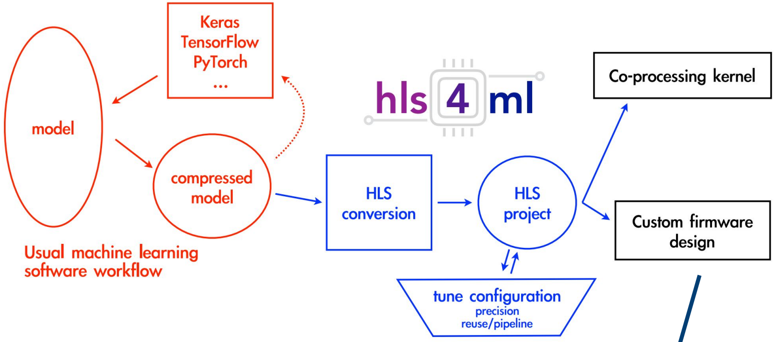

- Improvement in $E_T^{miss}$ calculation, **reduction in tails** of residual
- Returns likelihood of track belonging to vertex -> **flexible threshold** for downstream algorithms vs single window based baseline approach

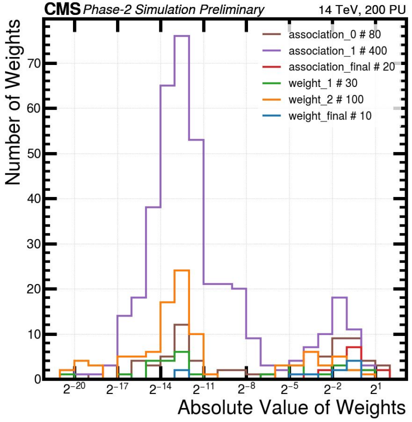
# Firmware - Network Compression

**Quantisation**:
Restrict Bitwidths
Reduce DSP usage

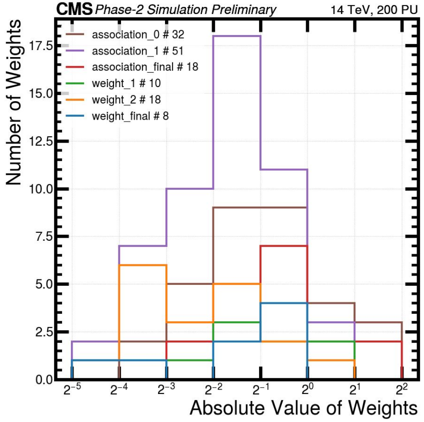
**Pruning**:
Iteratively Remove Weights
L1 Regularization

Split Model into 3 parts ->
Weight
Pattern
Association




8 training cycles



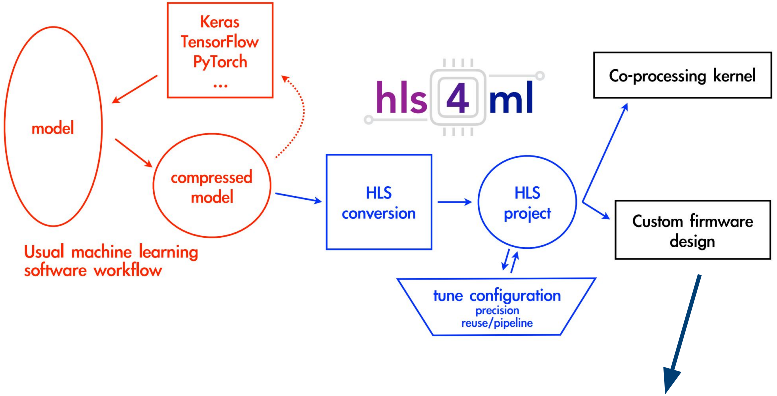

| Xilinx VU9P | Latency (ns) | DSPs % |
|---|---|---|
| NN Weight | 28 | 1.89 |
| **QPNN Weight** | **14** | **0.00** |
| NN Pattern | 42 | 3.74 |
| **QPNN Pattern** | **30** | **0.00** |
| NN Assoc. | 30 | 6.04 |
| **QPNN Assoc.** | **18** | **0.00** |

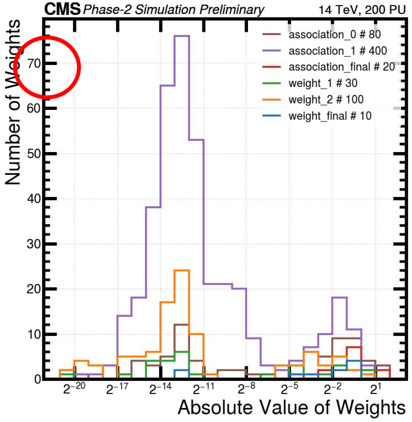
# Firmware - Network Compression

**Quantisation**:
Restrict Bitwidths
Reduce DSP usage

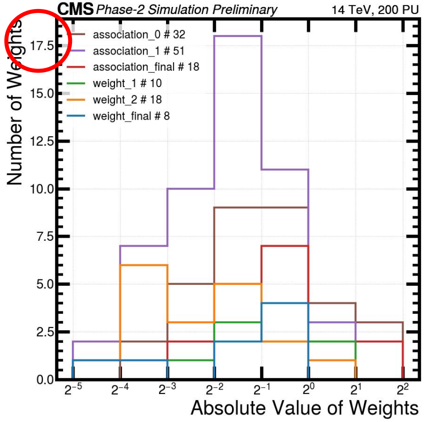
**Pruning**:
Iteratively Remove Weights
L1 Regularization

Split Model
into 3 parts ->
Weight
Pattern
Association




8 training cycles




| Xilinx VU9P | Latency (ns) | DSPs % |
|---|---|---|
| NN Weight | 28 | 1.89 |
| **QPNN Weight** | **14** | **0.00** |
| NN Pattern | 42 | 3.74 |
| **QPNN Pattern** | **30** | **0.00** |
| NN Assoc. | 30 | 6.04 |
| **QPNN Assoc.** | **18** | **0.00** |

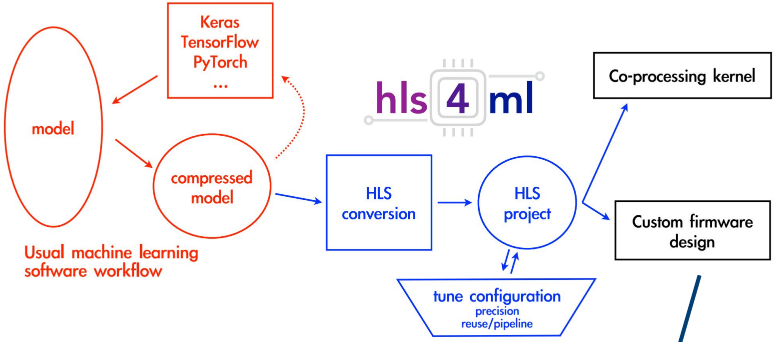
c.brown19@imperial.ac.uk

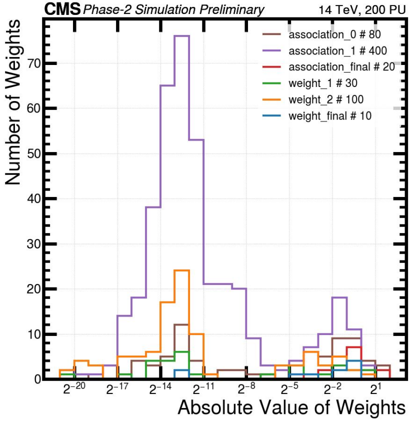
# Firmware - Network Compression

**Quantisation**:
Restrict Bitwidths
Reduce DSP usage

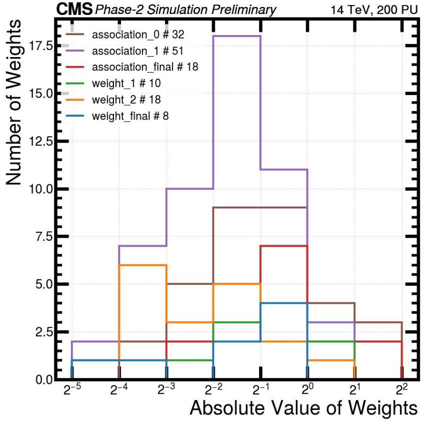
**Pruning**:
Iteratively Remove Weights
L1 Regularization

Split Model
into 3 parts ->
Weight
Pattern
Association




8 training cycles




| Xilinx VU9P | Latency (ns) | DSPs % |
|---|---|---|
| NN Weight | 28 | 1.89 |
| **QPNN Weight** | **14** | **0.00** |
| NN Pattern | 42 | 3.74 |
| **QPNN Pattern** | **30** | **0.00** |
| NN Assoc. | 30 | 6.04 |
| **QPNN Assoc.** | **18** | **0.00** |

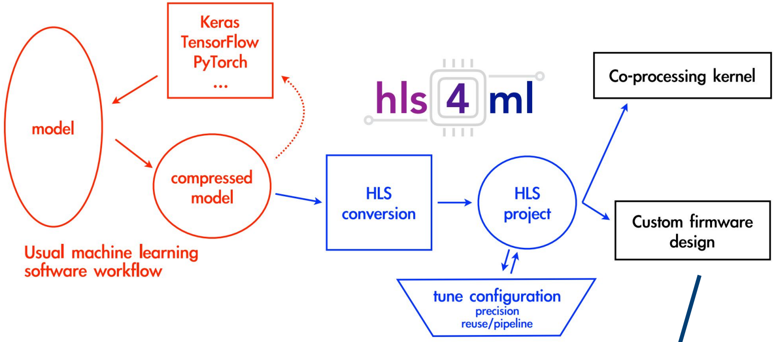
c.brown19@imperial.ac.uk

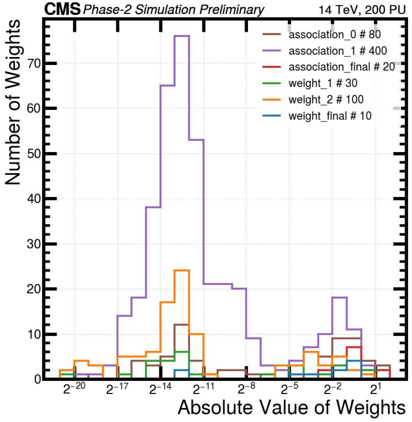
# Firmware - Network Compression

**Quantisation**:
Restrict Bitwidths
Reduce DSP usage

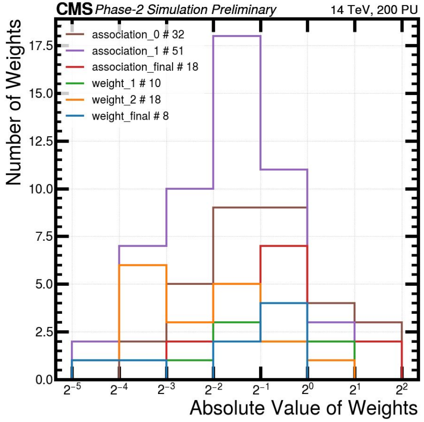
**Pruning**:
Iteratively Remove Weights
L1 Regularization

Split Model
into 3 parts ->
Weight
Pattern
Association




8 training
cycles

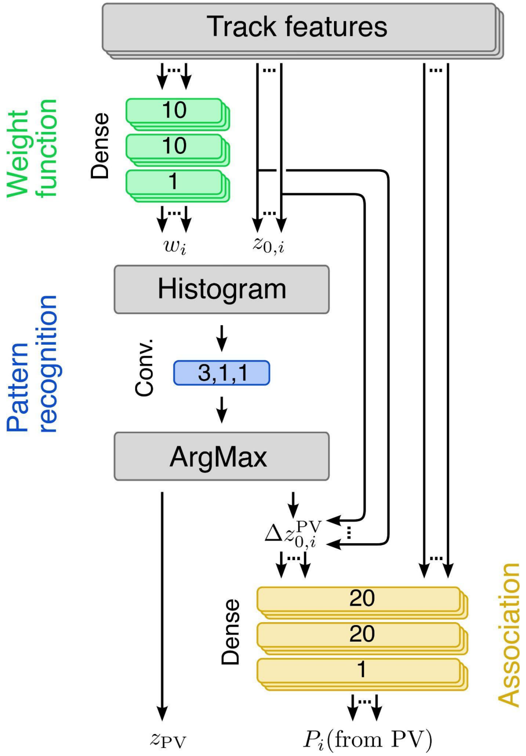



| Xilinx VU9P | Latency (ns) | DSPs % |
|---|---|---|
| NN Weight | 28 | 1.89 |
| **QPNN Weight** | **14** | **0.00** |
| NN Pattern | 42 | 3.74 |
| **QPNN Pattern** | **30** | **0.00** |
| NN Assoc. | 30 | 6.04 |
| **QPNN Assoc.** | **18** | **0.00** |

c.brown19@imperial.ac.uk

# Implementation

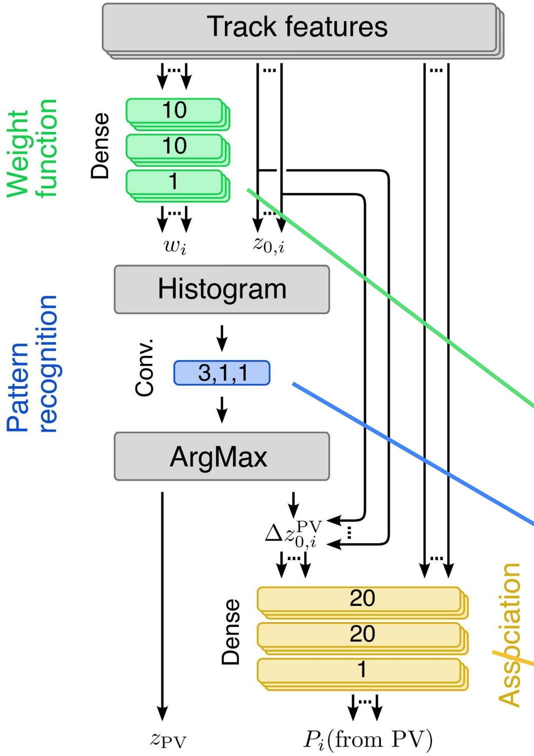
- Insert networks within existing baseline firmware
- Overall **top entities** controlling input output signals of networks
- Targeted ⅓ **Xilinx VU9P** running at **360 MHz**
- **108 ns** total algorithm latency (2x baseline approach, still faster than required latency to be passed downstream)



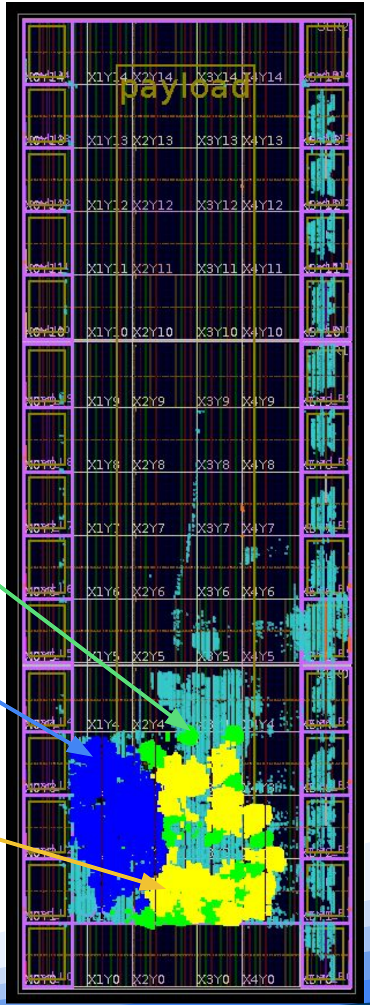

c.brown19@imperial.ac.uk

# Implementation

- Insert networks within existing baseline firmware
- Overall **top entities** controlling input output signals of networks
- Targeted ⅓ **Xilinx VU9P** running at **360 MHz**
- **108 ns** total algorithm latency (2x baseline approach, still faster than required latency to be passed downstream)




Floor plan of VU9P chip




c.brown19@imperial.ac.uk

**Tracker Inputs**

**Track Finder**

Tracklet Road Search

Kalman Filter

Track Quality

**Global Track Trigger**

Baseline Approach

Improved Baseline

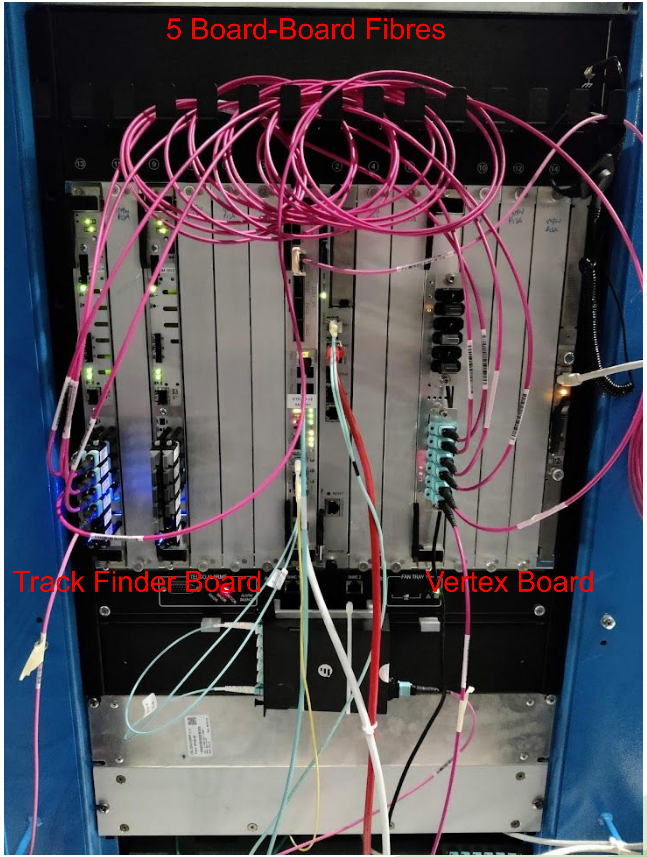
End-to-end NN approach

Firmware Implementation

**Demonstration**

# Commissioning and Testing

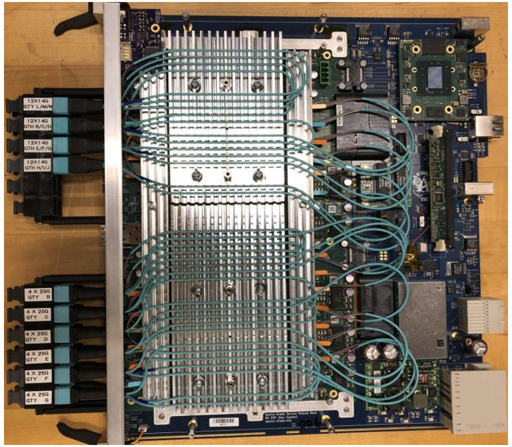
- Testing algorithms on **physical hardware** & testing communication between L1 subsystems
- Individually tested parts of Track Finder chain and Baseline Vertexing approach
- Ran **board to board tests** of Track Finder and Vertexing, can measure latency between subsystems
- **High speed fibre optics** up to 28 Gb/s



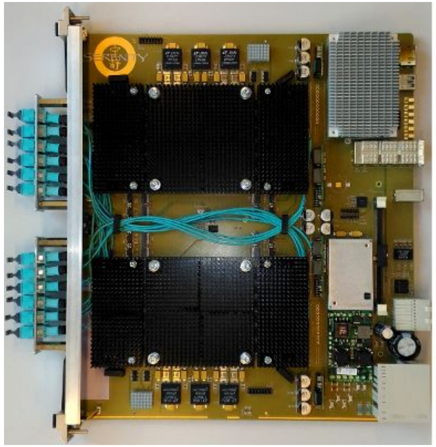

c.brown19@imperial.ac.uk

# Commissioning and Testing

- Testing algorithms on **physical hardware** & testing communication between L1 subsystems
- Individually tested parts of Track Finder chain and Baseline Vertexing approach
- Ran **board to board tests** of Track Finder and Vertexing, can measure latency between subsystems
- **High speed fibre optics** up to 28 Gb/s



Track Finder Board

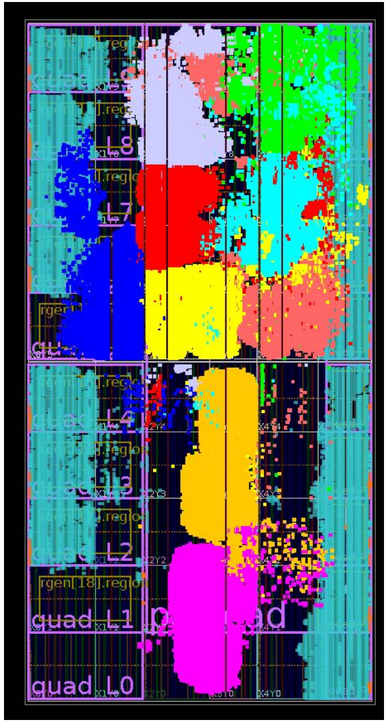


Vertex Board

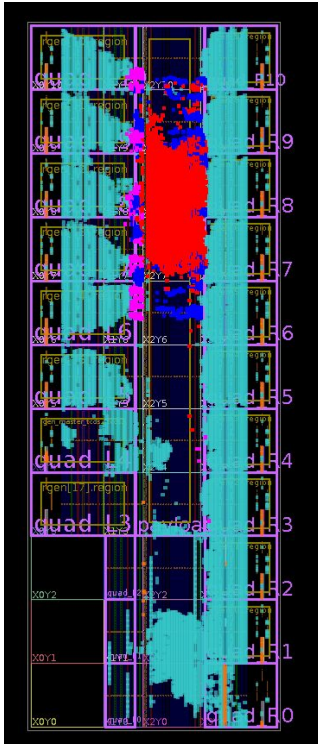
c.brown19@imperial.ac.uk

# Commissioning and Testing

- Testing algorithms on **physical hardware** & testing communication between L1 subsystems
- Individually tested parts of Track Finder chain and Baseline Vertexing approach
- Ran **board to board tests** of Track Finder and Vertexing, can measure latency between subsystems
- **High speed fibre optics** up to 28 Gb/s



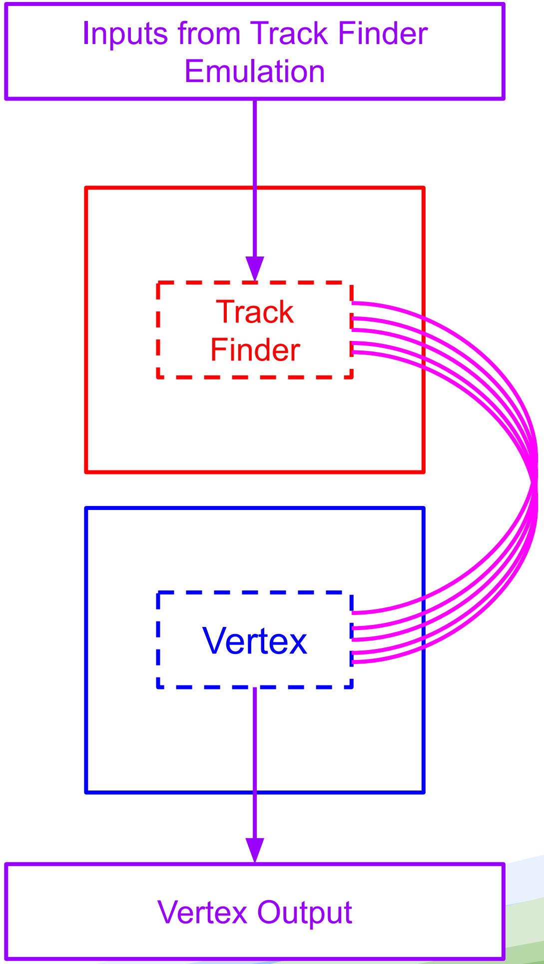
Track Finder
FPGA Floorplan



Vertex FPGA
Floorplan

# Commissioning and Testing

- Testing algorithms on **physical hardware** & testing communication between L1 subsystems
- Individually tested parts of Track Finder chain and Baseline Vertexing approach
- Ran **board to board tests** of Track Finder and Vertexing, can measure latency between subsystems
- **High speed fibre optics** up to 28 Gb/s




c.brown19@imperial.ac.uk

**Tracker Inputs**

**Track Finder**

Tracklet Road Search

Kalman Filter

Track Quality

**Global Track Trigger**

Baseline Approach

Improved Baseline

End-to-end NN approach

Firmware Implementation

**Demonstration**

## Future plans….

**Expand small scale tests to full track finding chain, displaced track finding at L1**

**End-to-end in board to board tests, vertex quality and large scale physics studies**

**Expand integration tests to larger parts of L1 trigger with multi-board tests**

**Tracker Inputs**

$p_T$ modules making online track finding possible

**Track Finder**

Tracklet Road Search

Kalman Filter

Hybrid algorithm performing online track finding within 4 μs

Track Quality

**Global Track Trigger**

Baseline Approach

Improved Baseline

End-to-end NN approach

New end-to-end neural network approach to vertex finding and association outperforming previous approaches, running on an FPGA. More info -> CMS-CR-2022-018

Firmware Implementation

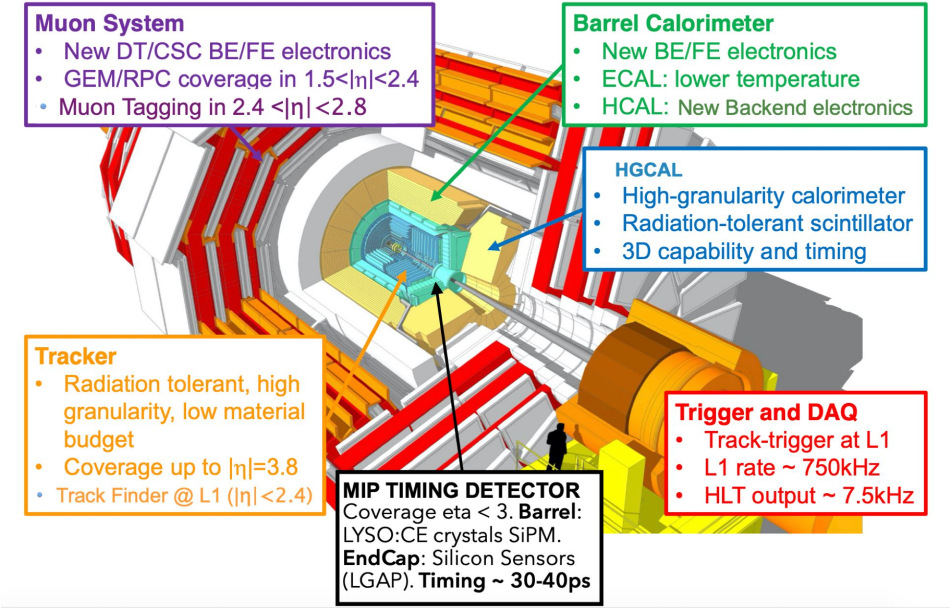
**Demonstration**

First tests of Track Finder and L1 trigger subsystems with board to board communications

# Backup

# CMS Phase-2 Upgrade

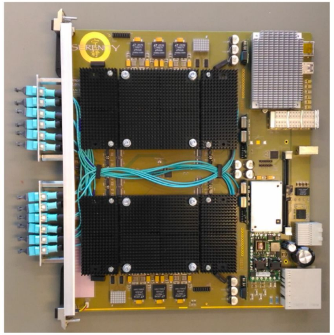
- Brand new tracker -> **radiation tolerant**, $200m^2$ of silicon, coverage up to η = 3.8
- Outer tracker for L1 trigger up to η = 2.4
- Muon systems increased η coverage and electronics
- Barrel calorimeter new electronics and lower ECAL temperature
- All new HGCAL end cap calorimetry, 4D (space-time) shower measurement
    - **High granularity** readout $1cm^2$
    - **Precision timing** < 50 ps

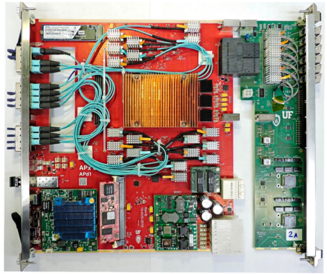


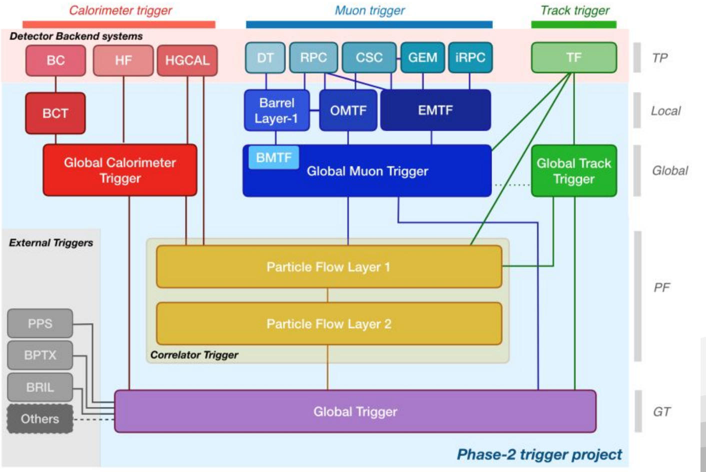

c.brown19@imperial.ac.uk

# CMS Phase-2 Upgrade - Trigger

- ATCA based cards for different trigger subsystems
- Xilinx Ultrascale+ FPGAs used throughout > 200 FPAs
- Optical link speeds up to **28 Gb/s**
- Dedicated **scouting system** at 40 MHz
- **Full event reconstruction** at L1, using **particle flow** algorithms, all sub-detector information used to reconstruct jets, missing $E_T$ and leptons
- Vertex used in **Pile Up Per Particle Identification** (PUPPI) to filter particles most likely to come from primary vertex



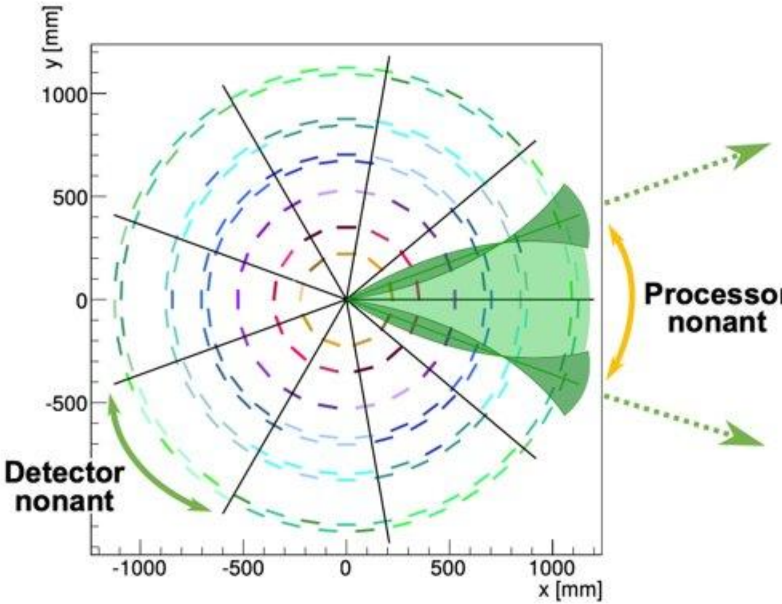





c.brown19@imperial.ac.uk

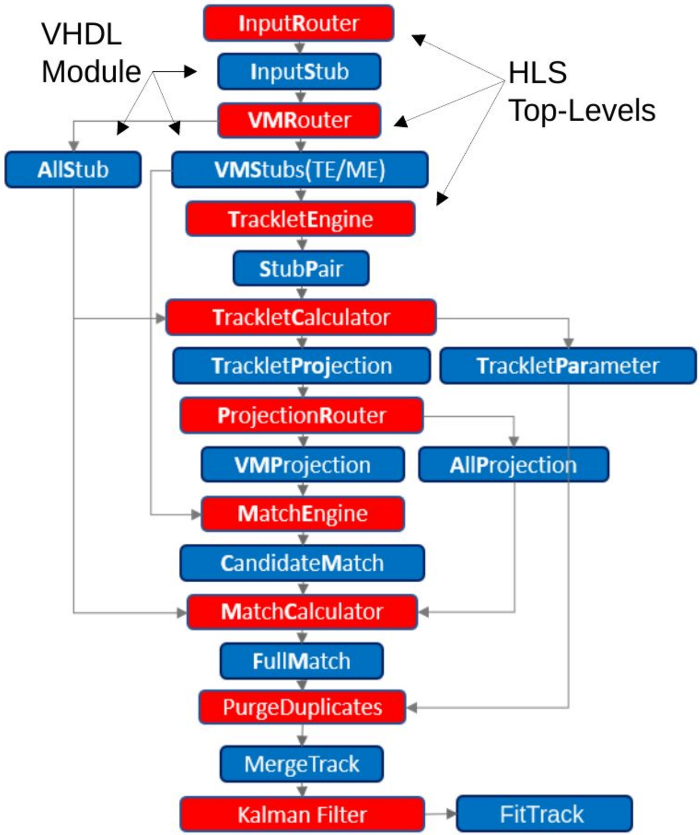
# Track Finder System

- 9 regions in ϕ
- Stubs streamed at 40 MHz to Data Trigger and Control (DTC)
- DTCs route stubs to Track Finder (TF) boards
- **18 TF boards per nonant**, processing different events
- Nonant processing occurs in **parallel**, no communication between TF boards
- Streamed to downstream trigger in **18 streams**, +/- η in 9 nonants
- All implemented on FPGAs



# Track Finding Firmware Implementation

- Each tracklet step implemented in **HLS**
  - Sub chain tested in HW
  - Barrel only chain synthesised, being optimised
- KF and final trigger output written in **VHDL**
  - Both barrel only and full config tested in HW
- **Top level VHDL** controls overall dataflow and multiple instances of various modules
- Each module individually synthesized meeting timing and matching emulators



# BDT For Track Quality

Conifer

- Trained on TTbar PU200 sample, 170K events
- Using Conifer Package -> generate HLS code
- **Tunable fixed point precision** <10,5> used
- Targeted **VU9P 240MHz**, Initiation Interval = 1 cycle

| Model | Python AUC | HLS AUC | Latency (cycles) | LUT % | FF % | DSP % |
|---|---|---|---|---|---|---|
| BDT | 0.986 | 0.981 | 3 | 0.140 | 0.027 | 0.0 |

c.brown19@imperial.ac.uk

# Vertex Finding Concept

**Extended Baseline**

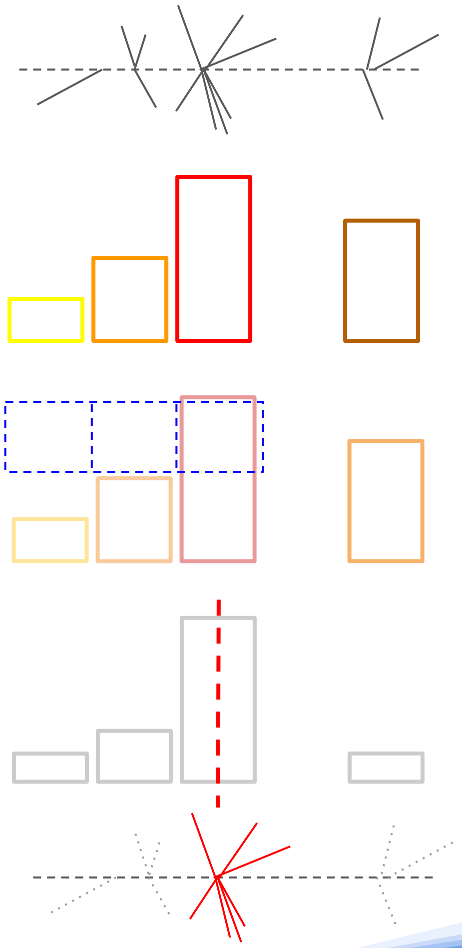
$p_T$ Weighting + $1/\eta^2$ additional weighting, approximation of curve on slide 12

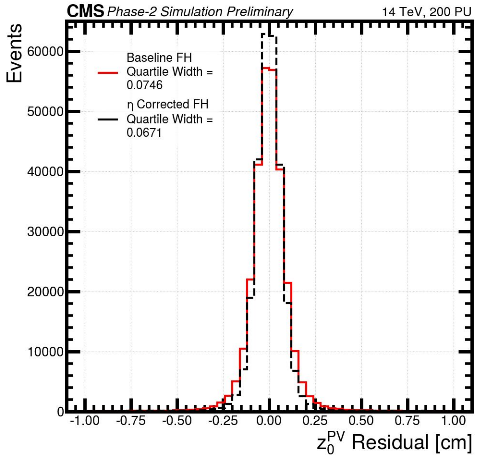
Weighted Histogram

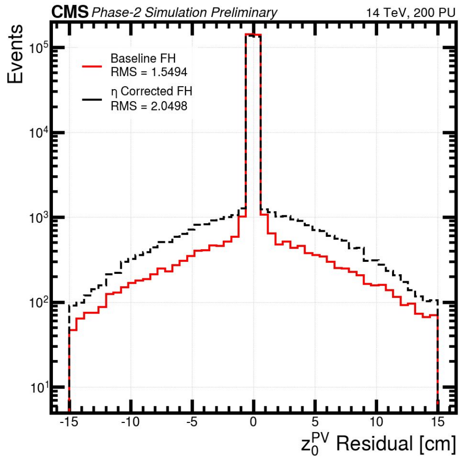
3-Bin Convolution

Peak Finder

$z_0$ window






c.brown19@imperial.ac.uk

# Vertex Finding Concept

**Extended Baseline**

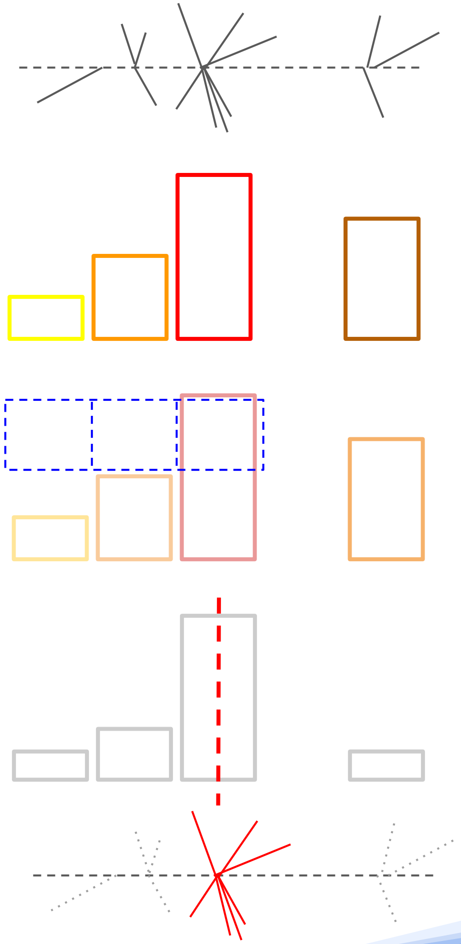
$p_T$ Weighting + $\chi^2$ cuts

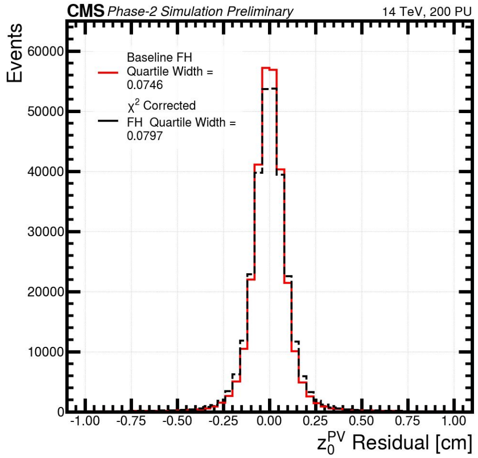
Weighted Histogram

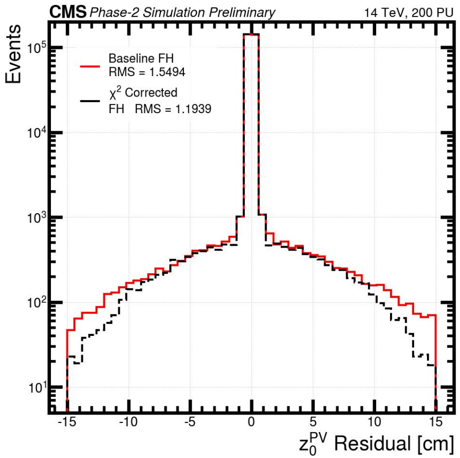
3-Bin Convolution

Peak Finder

$z_0$ window

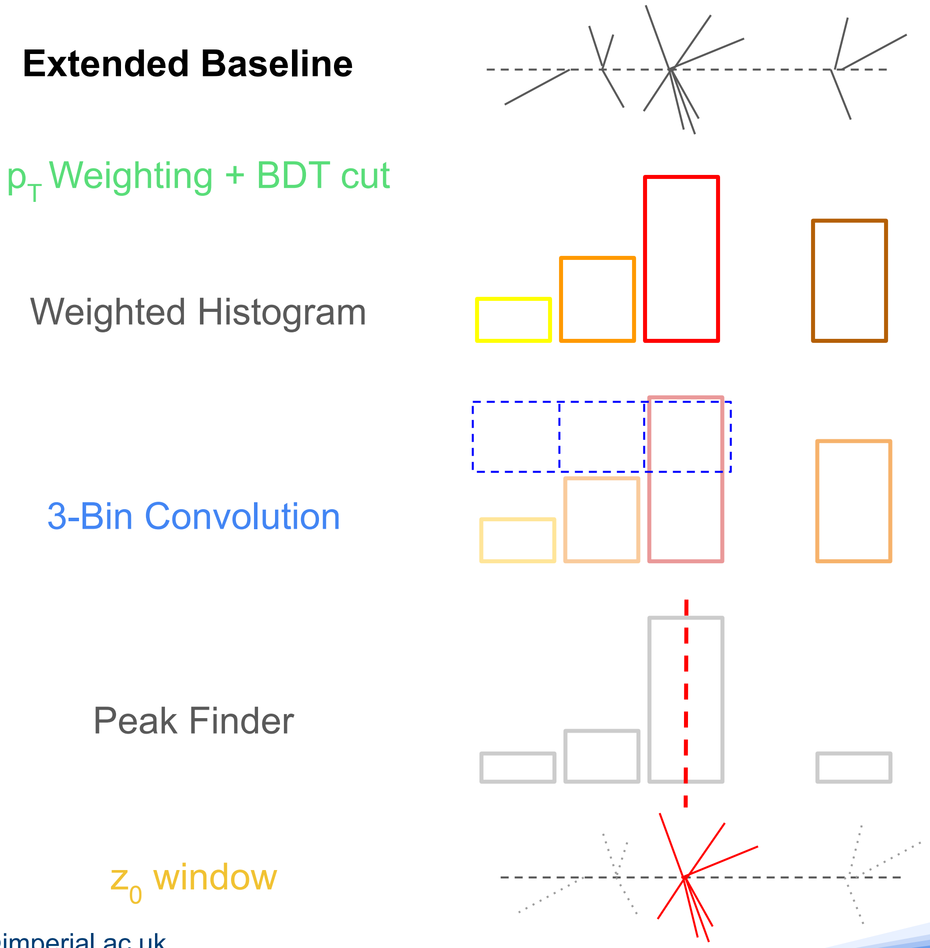


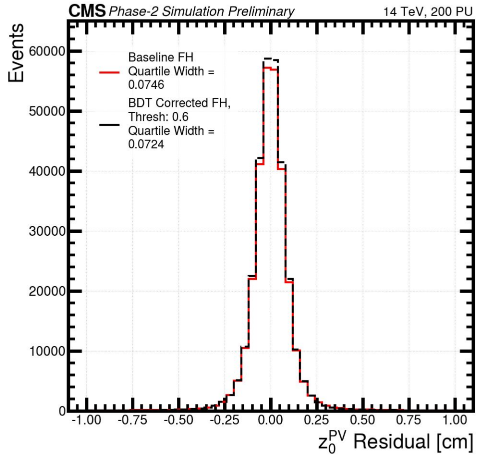


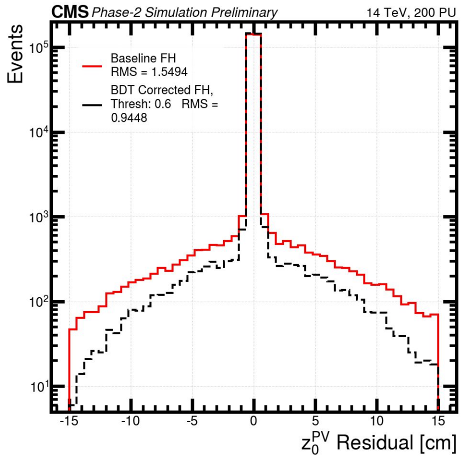




c.brown19@imperial.ac.uk

# Vertex Finding Concept

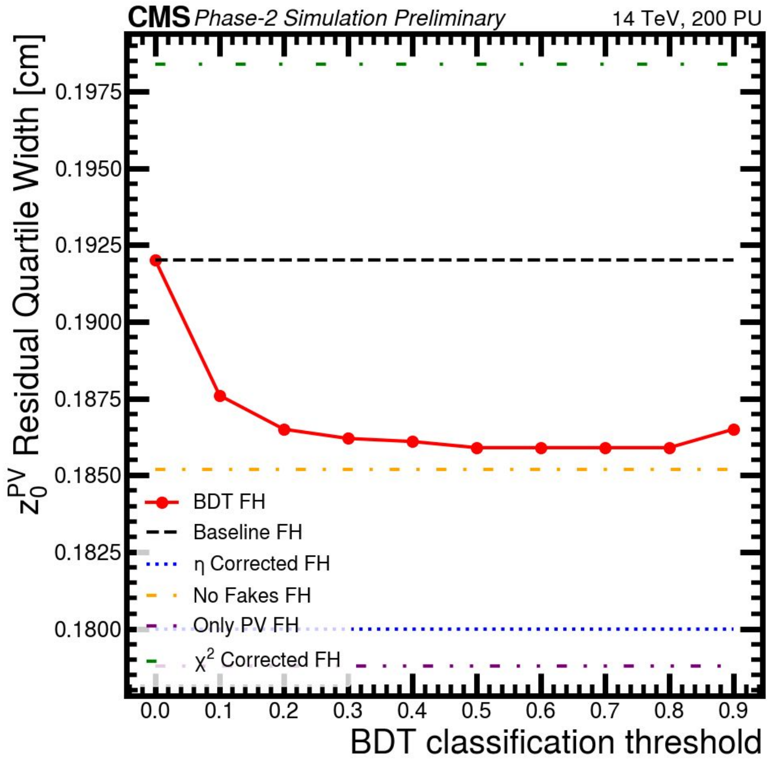


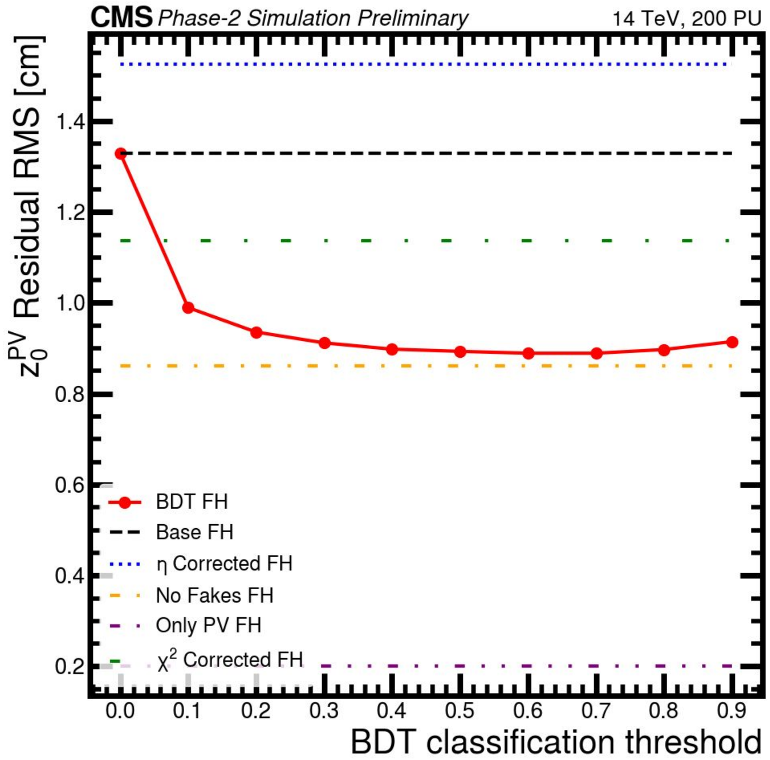




c.brown19@imperial.ac.uk

# Fast Histo Cuts




c.brown19@imperial.ac.uk

# Learning Track Weights

- Network learns ideal track weighting into histogram
- Histogram part of Network training cycle filled with:

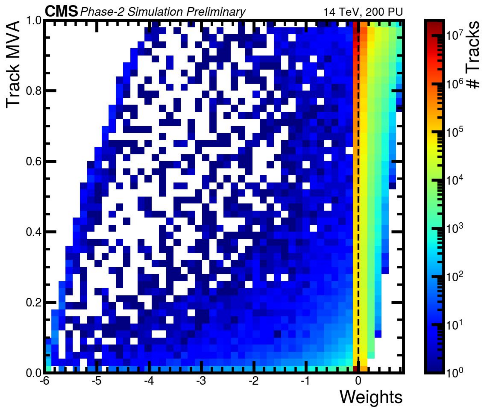
$$h_i = \sum_{j}^{\text{tracks}} \delta(j \in \text{bin } i) \times w(p_{\mathrm{T},j}, \eta_j, \chi^2_j, \ldots)$$

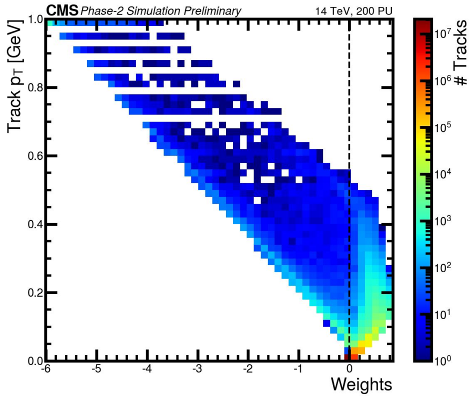
- Differentiated to give:

$$\frac{\partial h_i}{\partial \vec{w}} = \sum_{j}^{\text{tracks}} \delta(j \in \text{bin } i) \qquad \frac{\partial h_i}{\partial \vec{z}_0} = 0$$

- Passed through convolutional network and differentiable ArgMax to give peak

$$\sum_{i=0}^{N} i \frac{e^{x_i/T}}{\sum_{j=0}^{N} e^{x_j/T}}$$




c.brown19@imperial.ac.uk