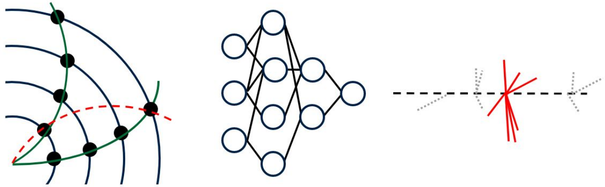
Neural Network-Based Primary Vertex Reconstruction with FPGAs for the Upgrade of the CMS Level-1 Trigger System

Christopher Brown, Benjamin Radburn-Smith, Alex Tapper (Imperial College London)
Matthias Komm (DESY)
Vladimir Loncar, Maurizio Pierini, Sioni Summers, Marcel Rod (CERN)

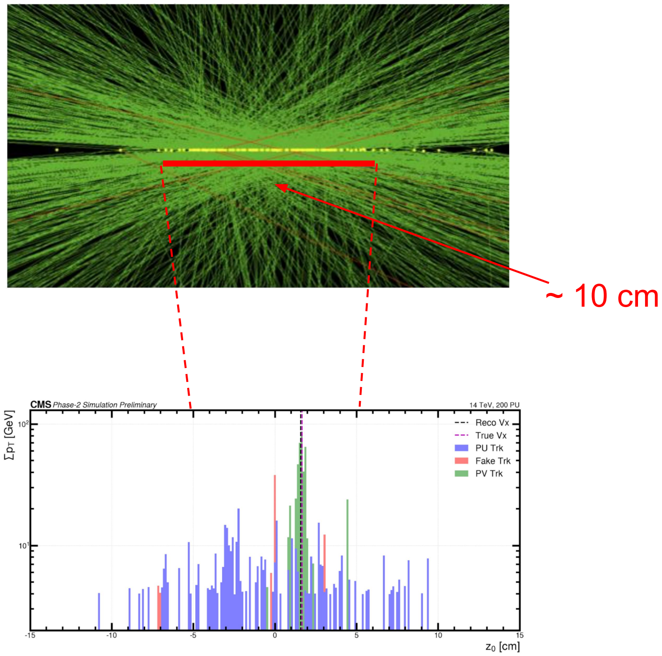
6th April 2022



Introduction

- HL-LHC, increased number of simultaneous proton-proton interactions per bunch crossing. Good for rare physics searches, bad for current era triggering
- Tracks for the first time at L1 trigger
- Tracks to locate primary vertex (the proton-proton collision with the highest $\sum p_T^2$)
- Associate tracks and other trigger objects to vertex, reducing impact of pileup on downstream algorithms (e.g. PUPPI) -> maintain sensitivity

High pile up HL-LHC

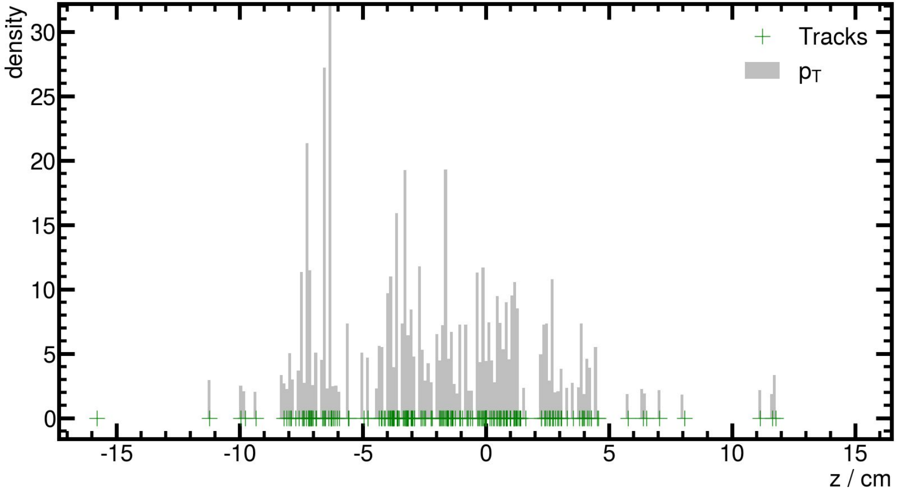


Finding primary vertex essential for reducing impact of pile up on L1 triggering

Baseline Vertex Finding Chain

Track Finding

Produces tracks > 2 GeV,
~100s per event with PU200



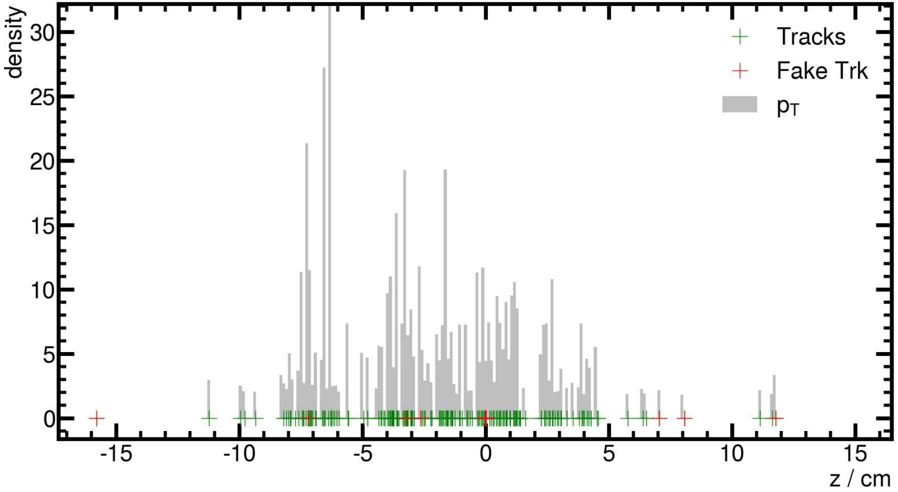
Baseline Vertex Finding Chain

Track Finding

Produces tracks > 2 GeV, ~100s per event with PU200

Track Quality

Based on χ^2 parameters from track finding, simple cuts



Baseline Vertex Finding Chain

Track Finding

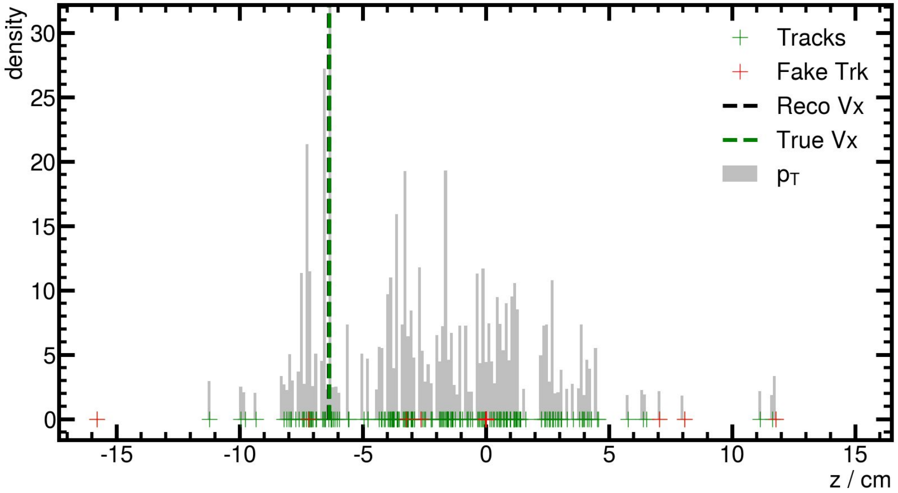
Produces tracks > 2 GeV, ~100s per event with PU200

Track Quality

Based on χ^2 parameters from track finding, simple cuts

Vertex Finding

FastHisto, histogram all tracks in z_0 weighted by p_T, find 3 consecutive bins with highest p_T



Baseline Vertex Finding Chain

Track Finding

Produces tracks > 2 GeV, ~100s per event with PU200

Track Quality

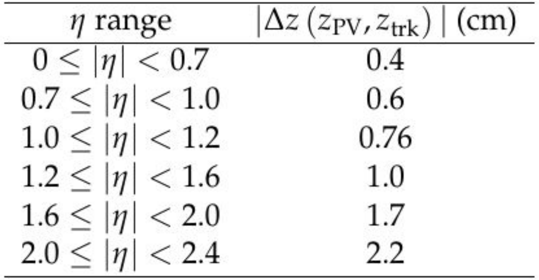
Based on χ^2 parameters from track finding, simple cuts

Vertex Finding

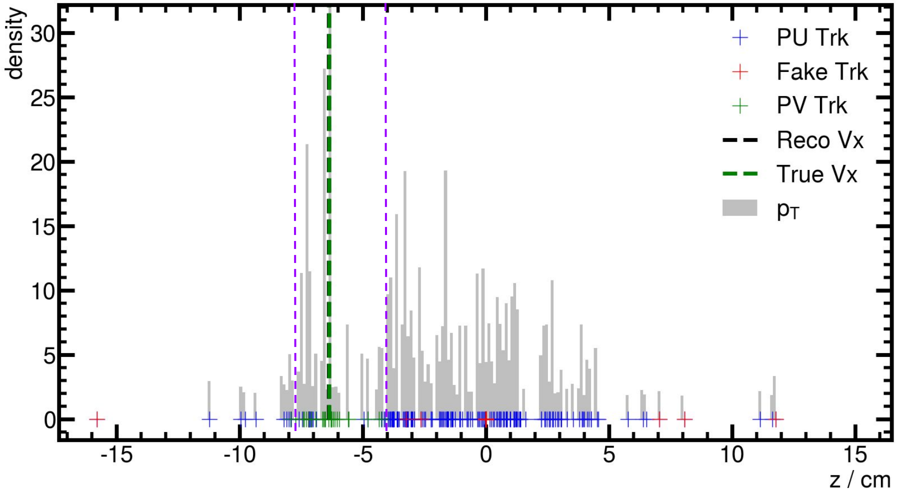
FastHisto, histogram all tracks in z_0 weighted by p_T, find 3 consecutive bins with highest p_T

Track to Vertex Association

Fixed window in z_0 or multiple windows based on track η

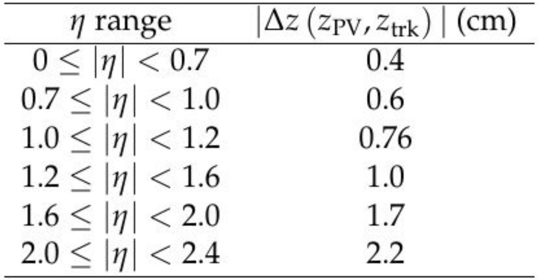


η range	$\lvert\Delta z\,(z_{\mathrm{PV}}, z_{\mathrm{trk}})\rvert$ (cm)
$0 \leq \lvert\eta\rvert < 0.7$	0.4
$0.7 \leq \lvert\eta\rvert < 1.0$	0.6
$1.0 \leq \lvert\eta\rvert < 1.2$	0.76
$1.2 \leq \lvert\eta\rvert < 1.6$	1.0
$1.6 \leq \lvert\eta\rvert < 2.0$	1.7
$2.0 \leq \lvert\eta\rvert < 2.4$	2.2

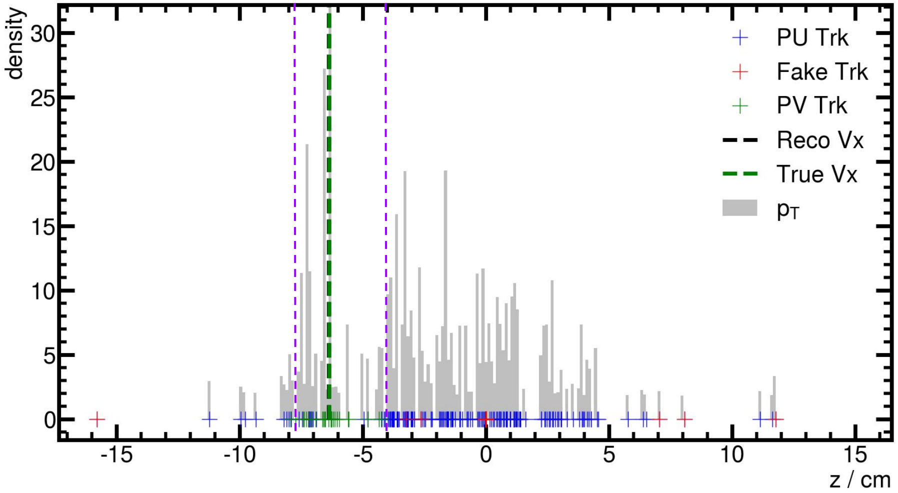


Baseline Vertex Finding Chain

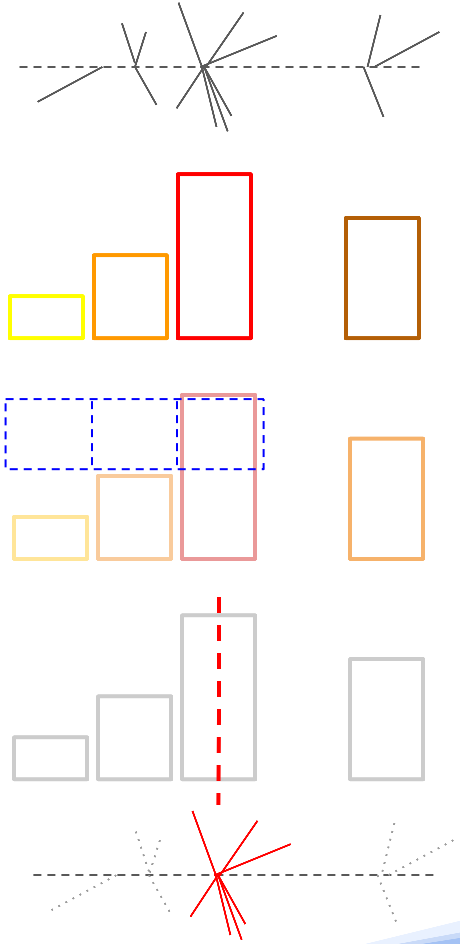
Track Finding	Produces tracks > 2 GeV, ~100s per event with PU200
Track Quality	Based on χ^2 parameters from track finding, simple cuts
Vertex Finding	FastHisto, histogram all tracks in z_0 weighted by p_T, find 3 consecutive bins with highest p_T
Track to Vertex Association	Fixed window in z_0 or multiple windows based on track η
Track E^T_{Miss} PF/PUPPI etc.	Downstream Algorithms



η range	$\lvert\Delta z\,(z_{\mathrm{PV}}, z_{\mathrm{trk}})\rvert$ (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

Argmax

Cut-Based

Vertex Finding Concept

Baseline

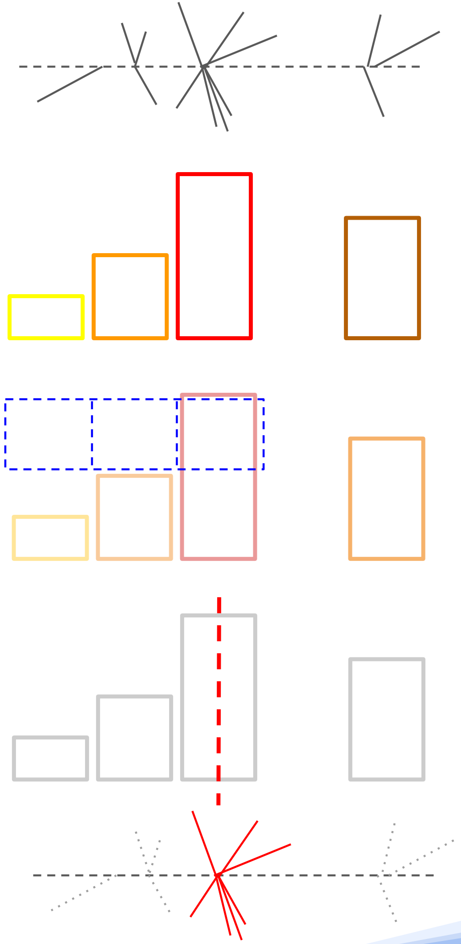
p_T Weighting

Weighted Histogram

3-Bin Convolution

Argmax

Cut-Based



End to End Neural Network

DNN multiple track features

Vertex Finding Concept

Baseline

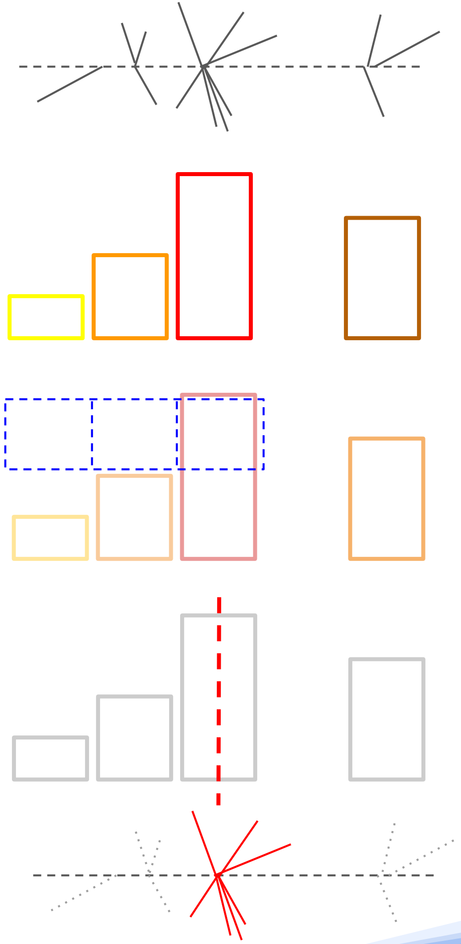
p_T Weighting

Weighted Histogram

3-Bin Convolution

Argmax

Cut-Based



End to End Neural Network

DNN multiple track features

Weighted Histogram

Vertex Finding Concept

Baseline

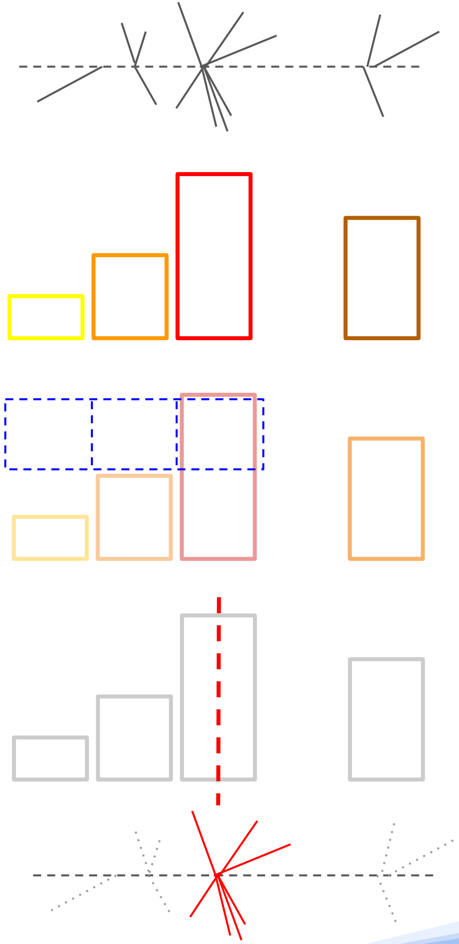
p_T Weighting

Weighted Histogram

3-Bin Convolution

Argmax

Cut-Based



End to End Neural Network

DNN multiple track features

Weighted Histogram

Multilayered CNN

c.brown19@imperial.ac.uk

Vertex Finding Concept

Baseline

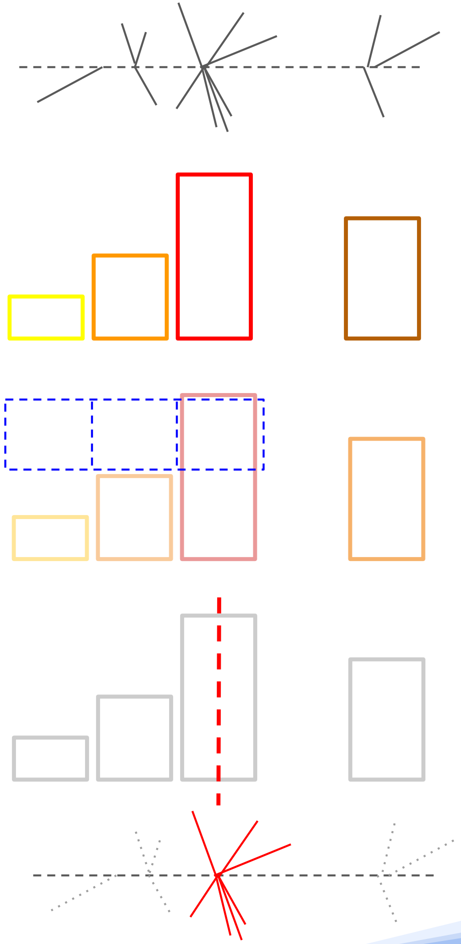
p_T Weighting

Weighted Histogram

3-Bin Convolution

Argmax

Cut-Based



End to End Neural Network

DNN multiple track features

Weighted Histogram

Multilayered CNN

Argmax

Vertex Finding Concept

Baseline

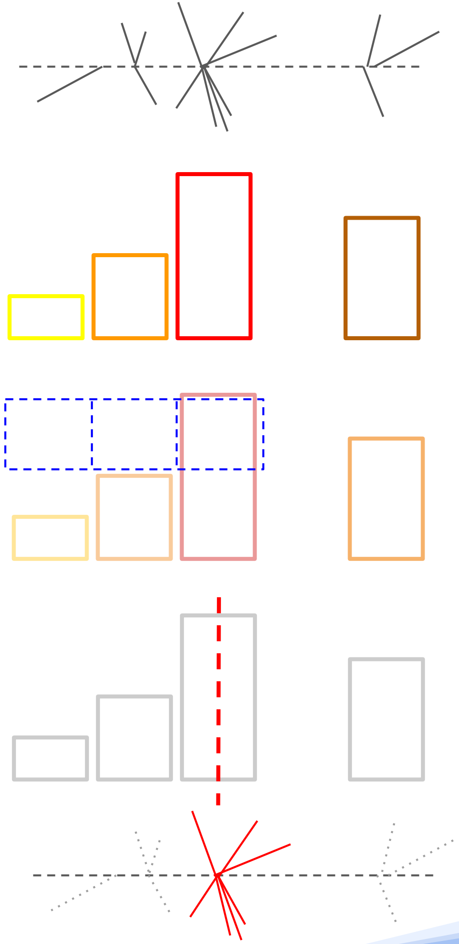
p_T Weighting

Weighted Histogram

3-Bin Convolution

Argmax

Cut-Based



End to End Neural Network

DNN multiple track features

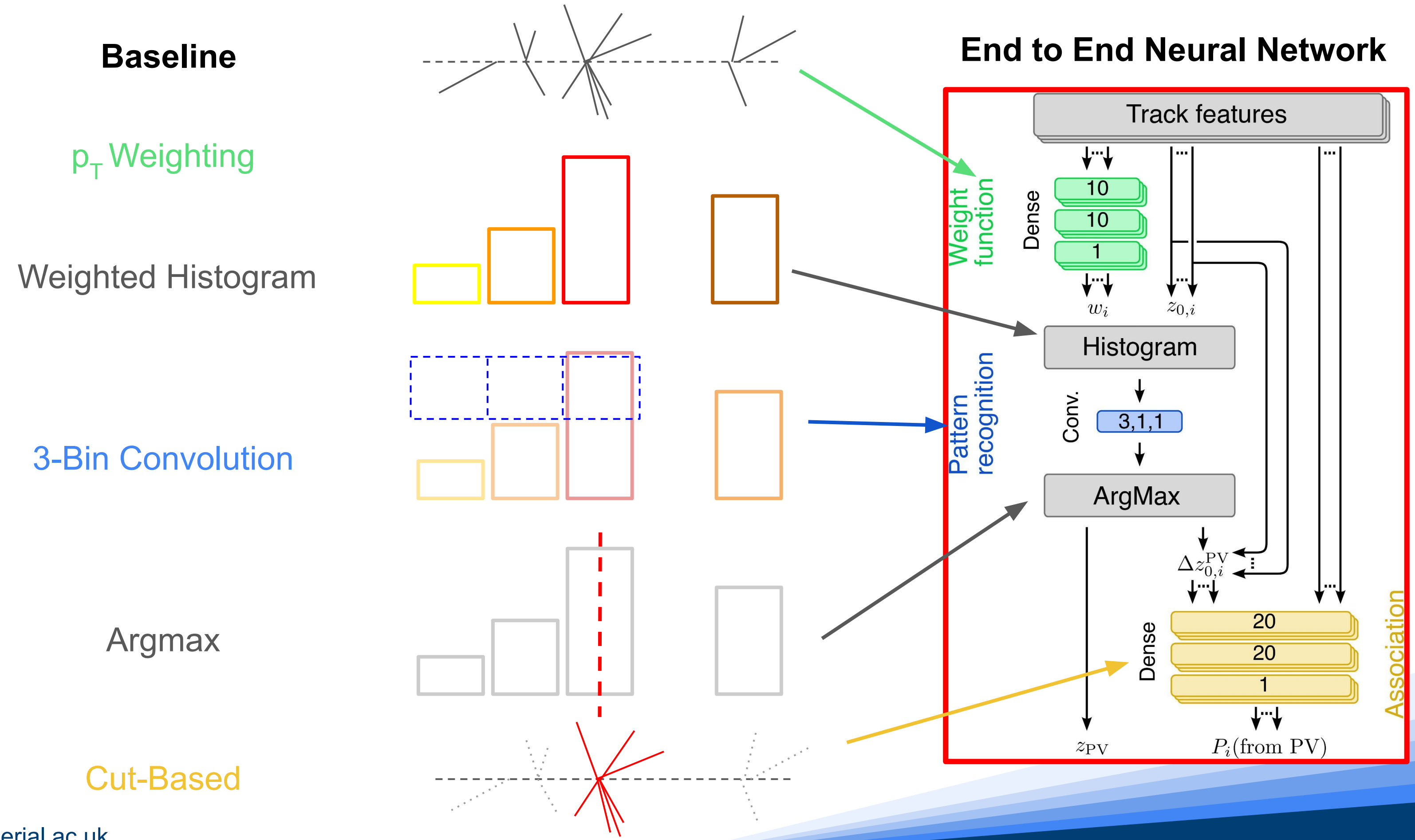
Weighted Histogram

Multilayered CNN

Argmax

DNN with z_0 distance, track features and latent features

Vertex Finding Concept



Baseline

p_T Weighting

Weighted Histogram

3-Bin Convolution

Argmax

Cut-Based

End to End Neural Network

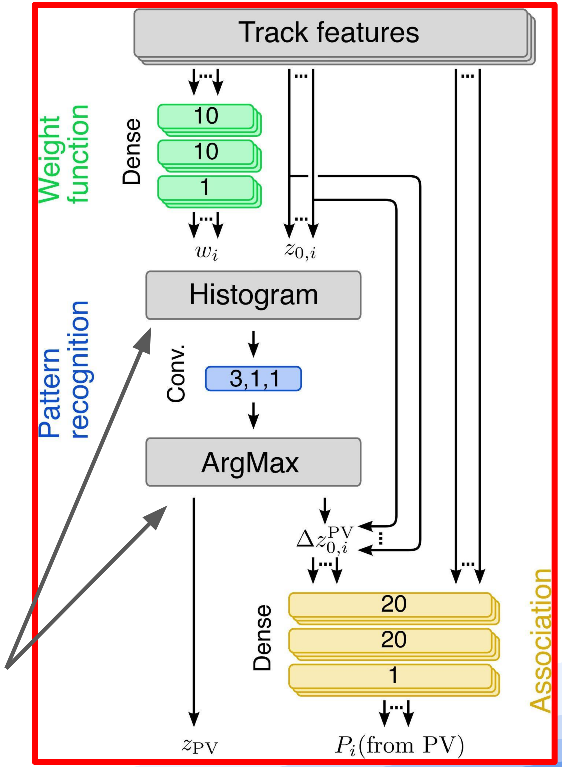
End to End Neural Networks for Vertex Finding

○ Network trained with 2 part loss function -> Event level PV regression, track level PV track classification

○ Simultaneous knowledge of both PV position and track to vertex association

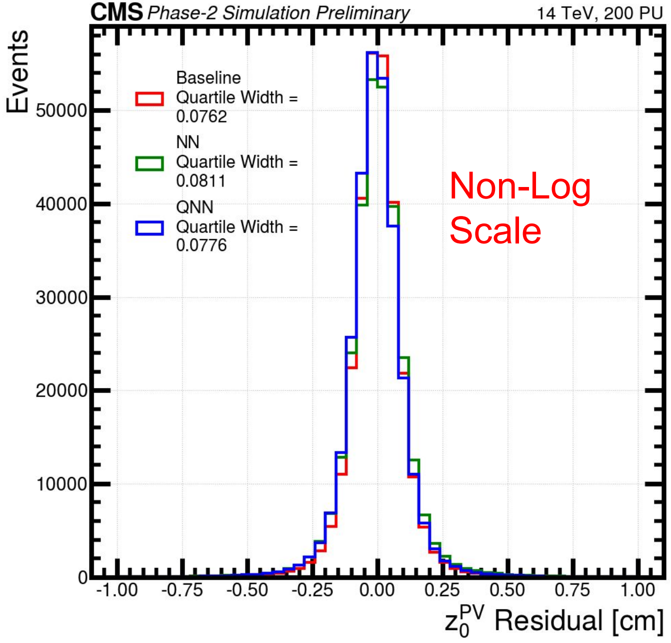
○ Robust to changes in track finding

○ Additional vertex quality

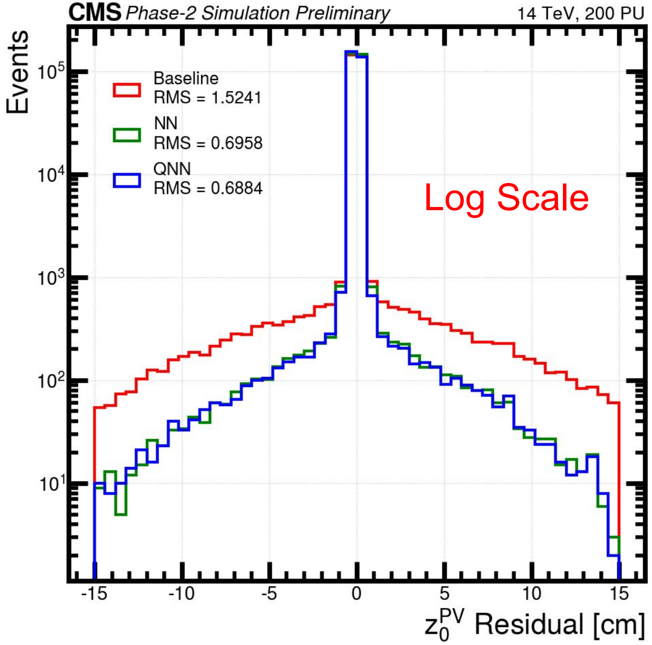


Differentiable

Performance - Vertex Regression



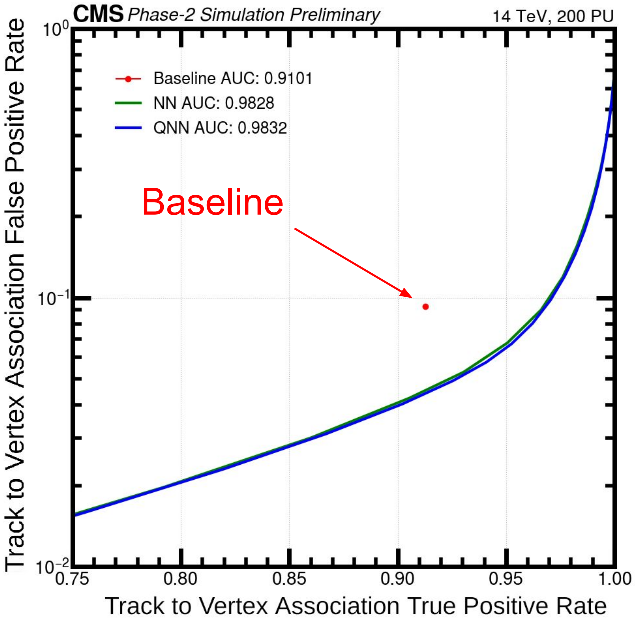
Non-Log Scale

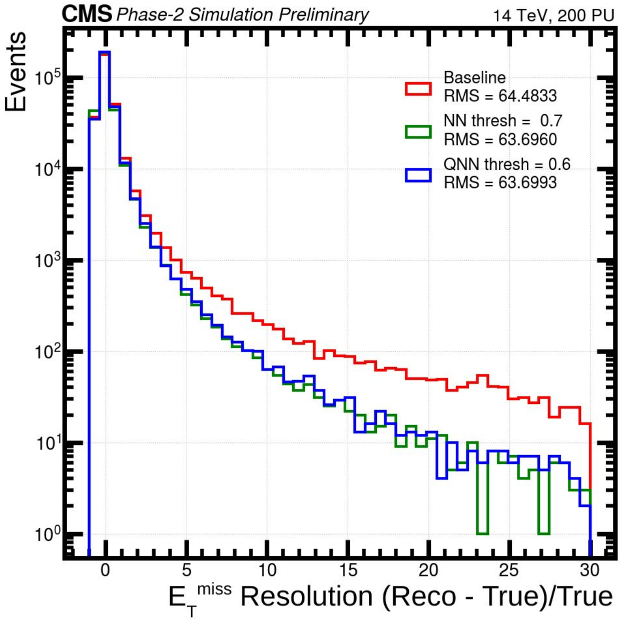


Log Scale

- 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

Performance - Track to Vertex Association



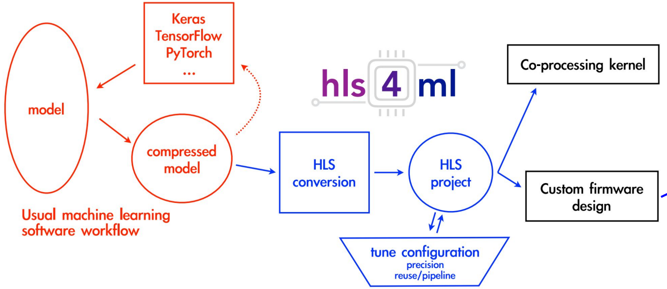


- Improvement in E_T^{miss} calculation, reduction in tails of residual
- Returns likelihood of track belonging to vertex -> flexible threshold for downstream algorithms

hls4ml - JINST 13 P07027 (2018)

Firmware - Network Compression

Split Model
into 3 parts ->
Weight
Pattern
Association

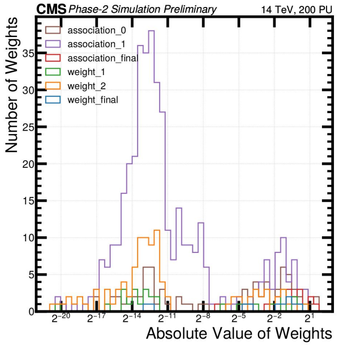


Wire up generated HDL
within existing VHDL
GTT

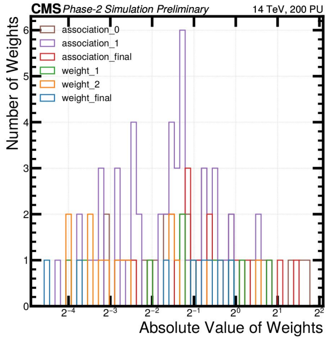
Quantisation:
Restrict Bitwidths
Reduce DSP usage

Pruning:
Iteratively Remove Weights
L1 Regularization

VU9P	Latency (ns)	Initiation Interval (ns)	LUTs %	DSPs %	BRAMs %	FFs %
NN Weight	28	2.0	0.17	1.89	0.00	0.08
QPNN Weight	**14**	**2.0**	**0.04**	**0.00**	**0.00**	**0.02**
NN Pattern	42	38	2.54	3.74	5.28	3.20
QPNN Pattern	**30**	**26**	**2.12**	**0.00**	**5.28**	**2.96**
NN Assoc.	30	2.0	0.60	6.04	0.00	0.28
QPNN Assoc.	**18**	**2.0**	**0.13**	**0.00**	**0.00**	**0.06**



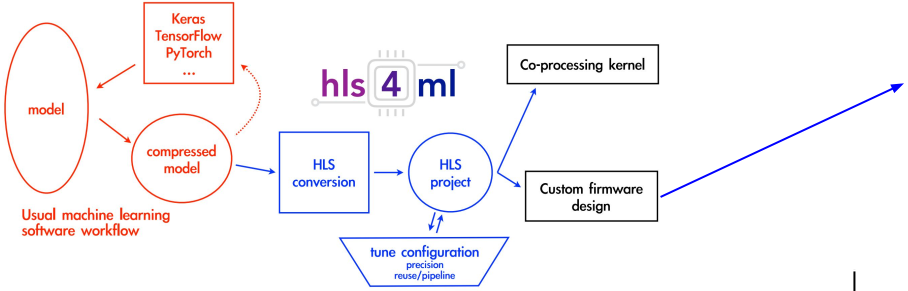
8 training cycles



hls4ml - JINST 13 P07027 (2018)

Firmware - Network Compression

Split Model
into 3 parts ->
Weight
Pattern
Association

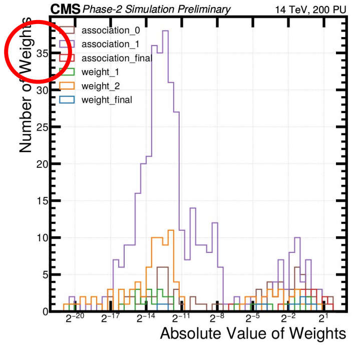


Wire up generated HDL within existing VHDL GTT

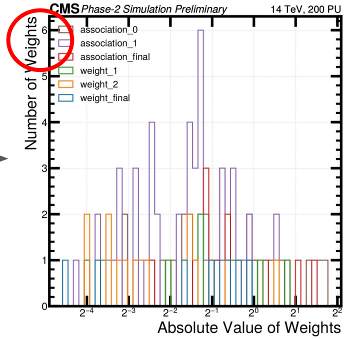
Quantisation:
Restrict Bitwidths
Reduce DSP usage

Pruning:
Iteratively Remove Weights
L1 Regularization

VU9P	Latency (ns)	Initiation Interval (ns)	LUTs %	DSPs %	BRAMs %	FFs %
NN Weight	28	2.0	0.17	1.89	0.00	0.08
QPNN Weight	**14**	**2.0**	**0.04**	**0.00**	**0.00**	**0.02**
NN Pattern	42	38	2.54	3.74	5.28	3.20
QPNN Pattern	**30**	**26**	**2.12**	**0.00**	**5.28**	**2.96**
NN Assoc.	30	2.0	0.60	6.04	0.00	0.28
QPNN Assoc.	**18**	**2.0**	**0.13**	**0.00**	**0.00**	**0.06**



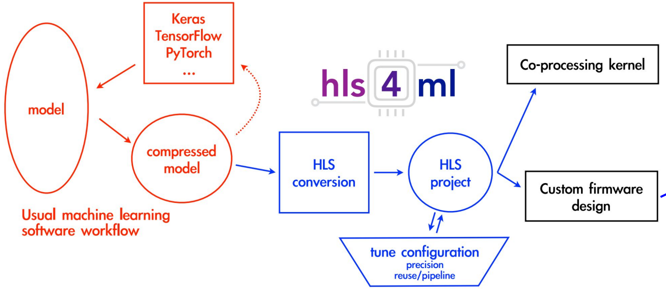
8 training cycles



hls4ml - JINST 13 P07027 (2018)

Firmware - Network Compression

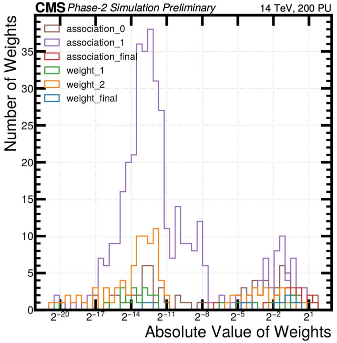
Split Model
into 3 parts ->
Weight
Pattern
Association



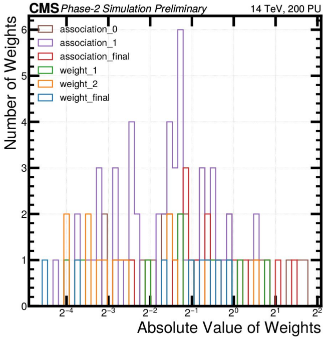
Wire up generated HDL
within existing VHDL
GTT

Quantisation:
Restrict Bitwidths
Reduce DSP usage

Pruning:
Iteratively Remove Weights
L1 Regularization



8 training cycles

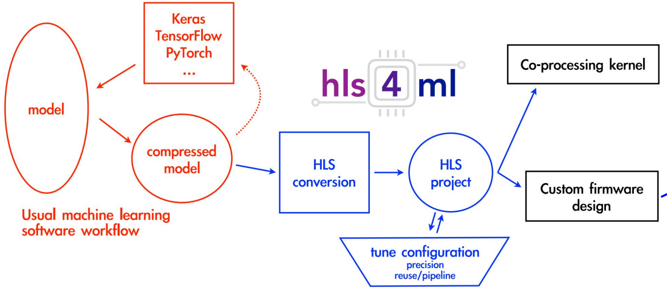


VU9P	Latency (ns)	Initiation Interval (ns)	LUTs %	DSPs %	BRAMs %	FFs %
NN Weight	28	2.0	0.17	1.89	0.00	0.08
QPNN Weight	**14**	**2.0**	**0.04**	**0.00**	**0.00**	**0.02**
NN Pattern	42	38	2.54	3.74	5.28	3.20
QPNN Pattern	**30**	**26**	**2.12**	**0.00**	**5.28**	**2.96**
NN Assoc.	30	2.0	0.60	6.04	0.00	0.28
QPNN Assoc.	**18**	**2.0**	**0.13**	**0.00**	**0.00**	**0.06**

hls4ml - JINST 13 P07027 (2018)

Firmware - Network Compression

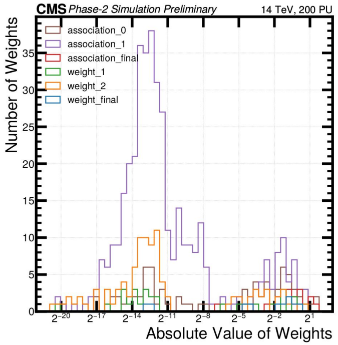
Split Model
into 3 parts ->
Weight
Pattern
Association



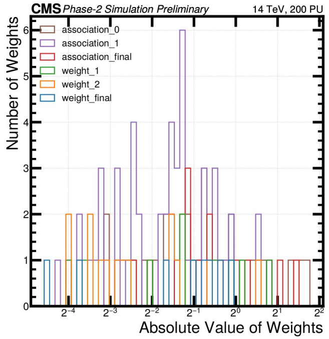
Wire up generated HDL within existing VHDL GTT

Quantisation:
Restrict Bitwidths
Reduce DSP usage

Pruning:
Iteratively Remove Weights
L1 Regularization



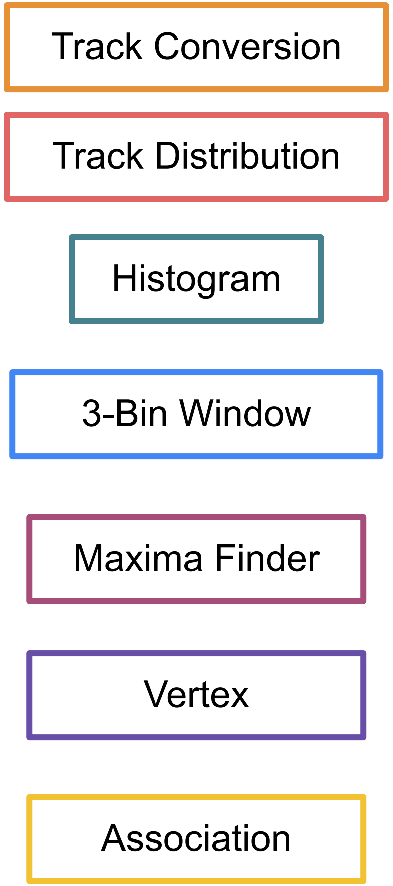
8 training cycles



VU9P	Latency (ns)	Initiation Interval (ns)	LUTs %	DSPs %	BRAMs %	FFs %
NN Weight	28	2.0	0.17	1.89	0.00	0.08
QPNN Weight	**14**	**2.0**	**0.04**	**0.00**	**0.00**	**0.02**
NN Pattern	42	38	2.54	3.74	5.28	3.20
QPNN Pattern	**30**	**26**	**2.12**	**0.00**	**5.28**	**2.96**
NN Assoc.	30	2.0	0.60	6.04	0.00	0.28
QPNN Assoc.	**18**	**2.0**	**0.13**	**0.00**	**0.00**	**0.06**

Implementation

- Take VHDL processing blocks of baseline histogramming approach
- VHDL **top entities** controlling input output signals of networks
- Targeted ⅓ **VU9P** running at **360 MHz**
- Meets timing after running networks through Vitis with better pipelining
- **108 ns** total algorithm latency



Track Conversion

Track Distribution

Histogram

3-Bin Window

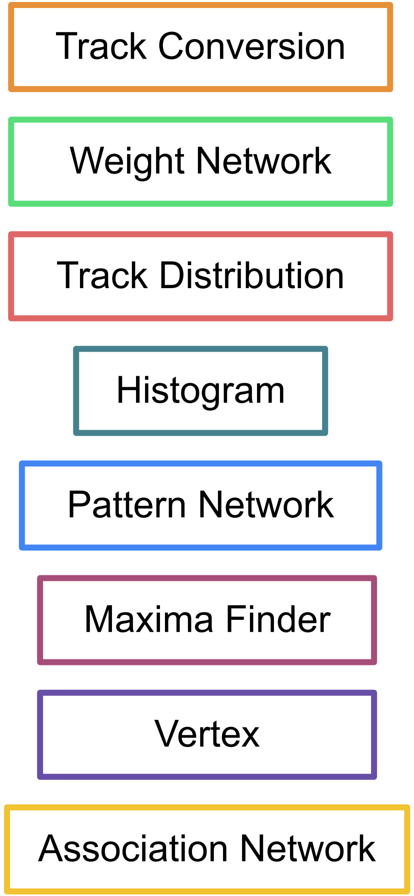
Maxima Finder

Vertex

Association

Implementation

- Take VHDL processing blocks of baseline histogramming approach
- VHDL **top entities** controlling input output signals of networks
- Targeted ⅓ **VU9P** running at **360 MHz**
- Meets timing after running networks through Vitis with better pipelining
- **108 ns** total algorithm latency



Track Conversion

Weight Network

Track Distribution

Histogram

Pattern Network

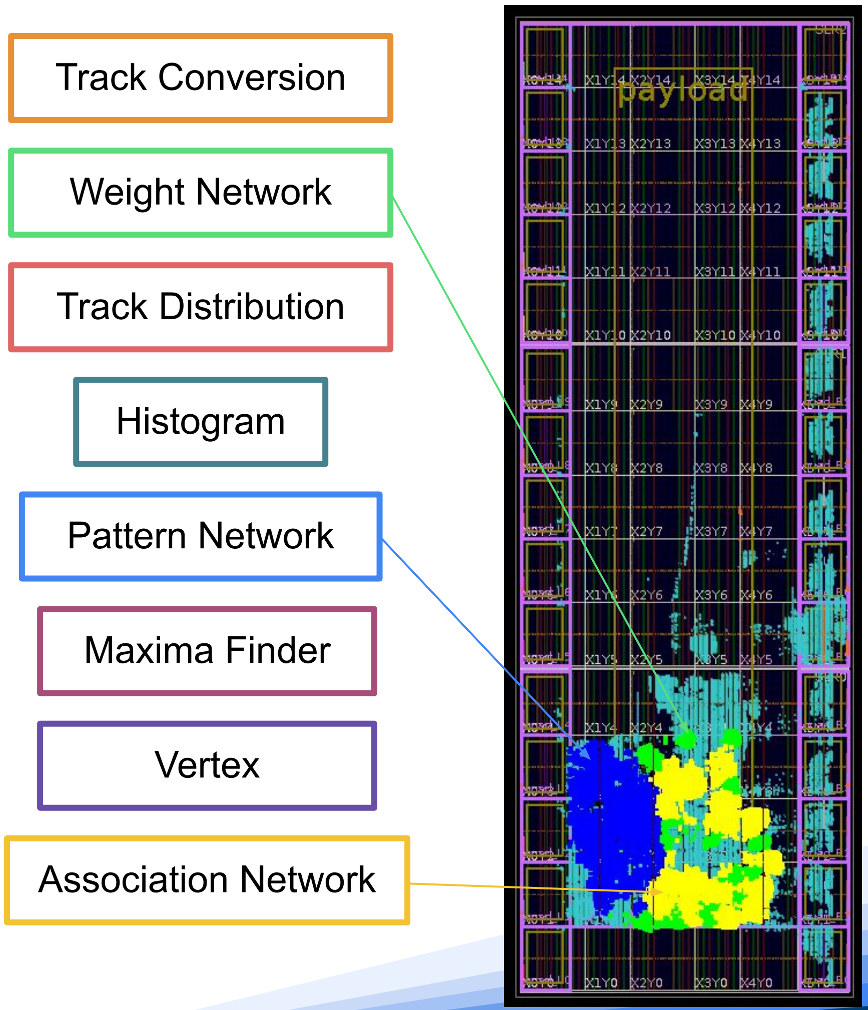
Maxima Finder

Vertex

Association Network

Implementation

- Take VHDL processing blocks of baseline histogramming approach
- VHDL **top entities** controlling input output signals of networks
- Targeted ⅓ **VU9P** running at **360 MHz**
- Meets timing after running networks through Vitis with better pipelining
- **108 ns** total algorithm latency



Track Conversion

Weight Network

Track Distribution

Histogram

Pattern Network

Maxima Finder

Vertex

Association Network

Conclusion

Baseline Approach to Vertex Finding

End to End Neural Network for Vertex Finding

- Concept
- Performance

Firmware and Network Compression

Implementation

CMS Conference Note

Future Steps

Verify in Hardware

Downstream Physics Impact

Vertex Quality Estimation

Backup

Learning Track Weights

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

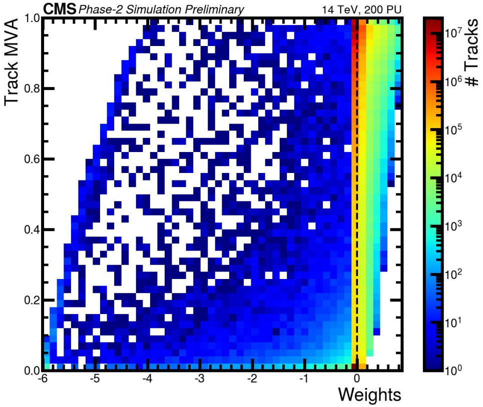
$$h_i = \sum_{j}^{\text{tracks}} \delta(j \in \text{bin } i) \times \boxed{w(p_{\mathrm{T},j}, \eta_j, \chi^2_j, \ldots)}$$

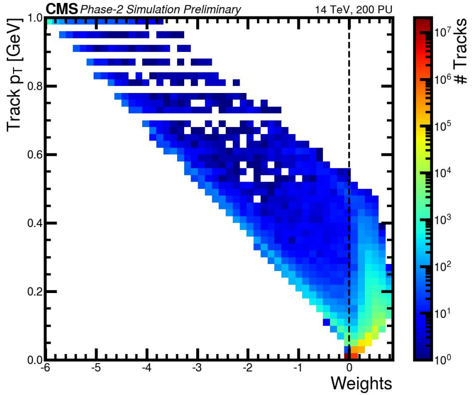
- Differentiated to give:

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

- Passed through convolutional network and differentiable ArgMax to give peak

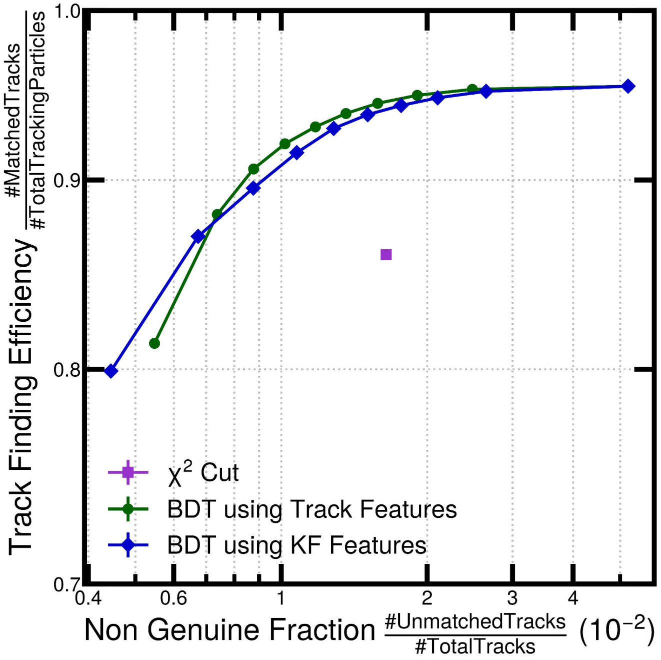
$$\sum_{i=0}^{N} i \frac{e^{x_i/T}}{\sum_{j=0}^{N} e^{x_j/T}}$$





Track Quality

- Fake rate is high ~20% at high p_T can use χ^2 cuts to reduce but big drop to tracking efficiency
- Use small BDTs to learn to classify fakes based on track fit and helix parameters
- Outperforms χ^2 cuts, high fake rejection with only small reduction to tracking efficiency
- Used as input feature to end-to-end neural network



c.brown19@imperial.ac.uk