Imperial College London

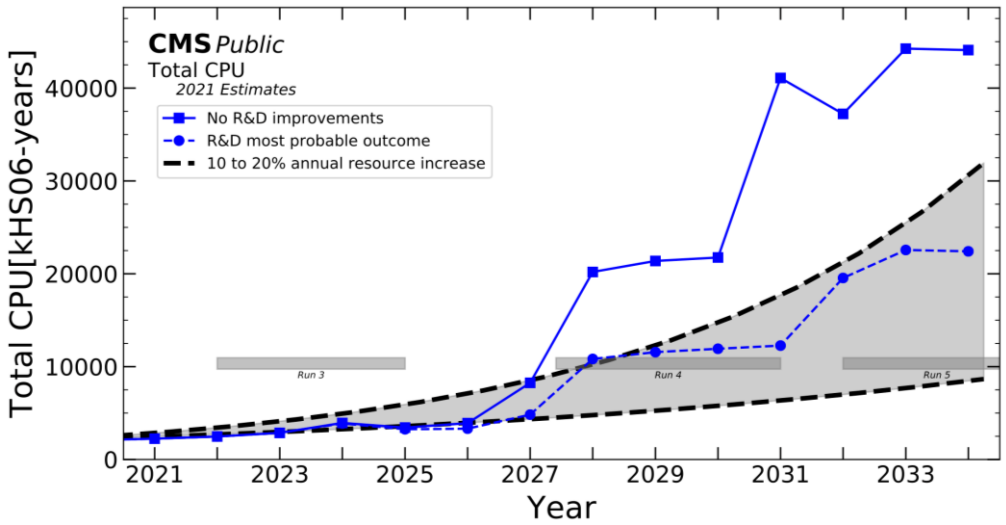
Acceleration of a CMS DNN based Algorithm

M. Barbone, W. Luk, **T. Ourida**, A. Tapper

tarik.ourida17@imperial.ac.uk

Motivation

- Data rate set to increase with High-Luminosity LHC upgrade
- Current approach not scalable to meet requirements of additional processing due to increased luminosity



https://twiki.cern.ch/twiki/bin/view/CMSPublic/CMSOfflineComputingResults

Contributions

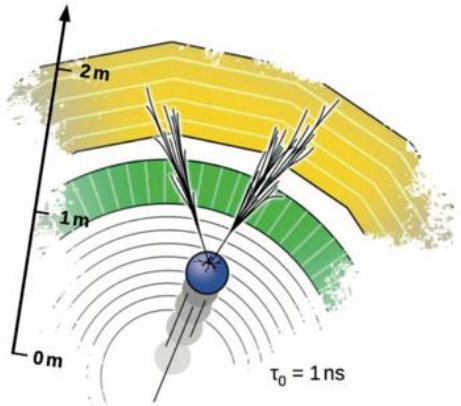
- Novel parallel architecture for Long-Lived Particle (LLP) jet tagging
  - compute: both space and time
  - accelerated processing: reduce cost of storage for post-processing

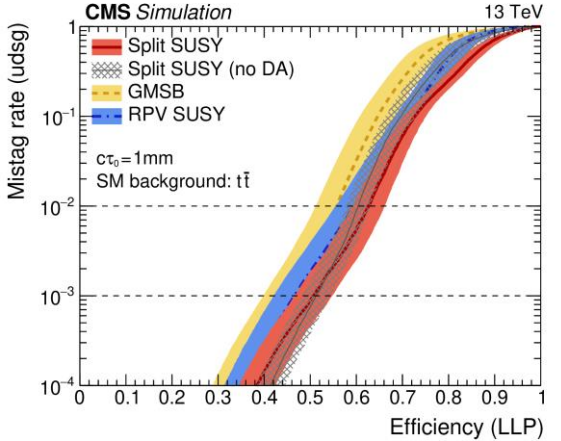
- Explore an FPGA architecture using a highly computationally intensive algorithm which could benefit from acceleration

Imperial College London

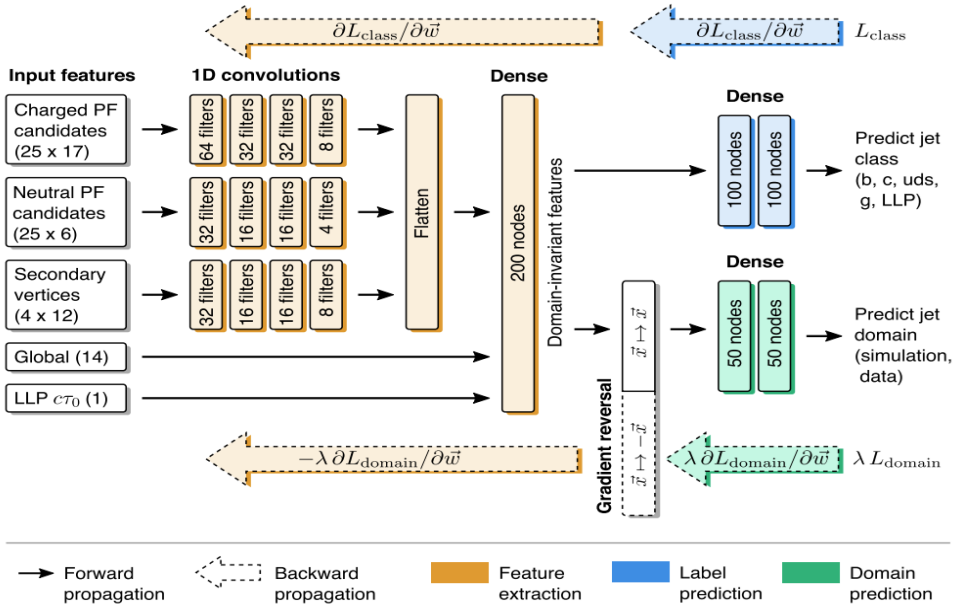
Long-Lived Particle Tagger

- CMS Collaboration, *A deep neural network to search for new long-lived particles decaying to jets*, Mach. Learn.: Sci. Technol. 1 (2020) 035012 (arXiv:1912.12238)
- Many models of new physics predict existence of exotic long-lived particles
- Lifetime frontier yet to be fully explored at the LHC
- Deep Neural Network trained on Monte Carlo to classify jets into Standard Model (light, b/c-quark gluon) and exotic long-lived particles
- Trained for lifetimes cτ ∈ [10 μm,10 m] with lifetime as a parameter – 1mm benchmark →



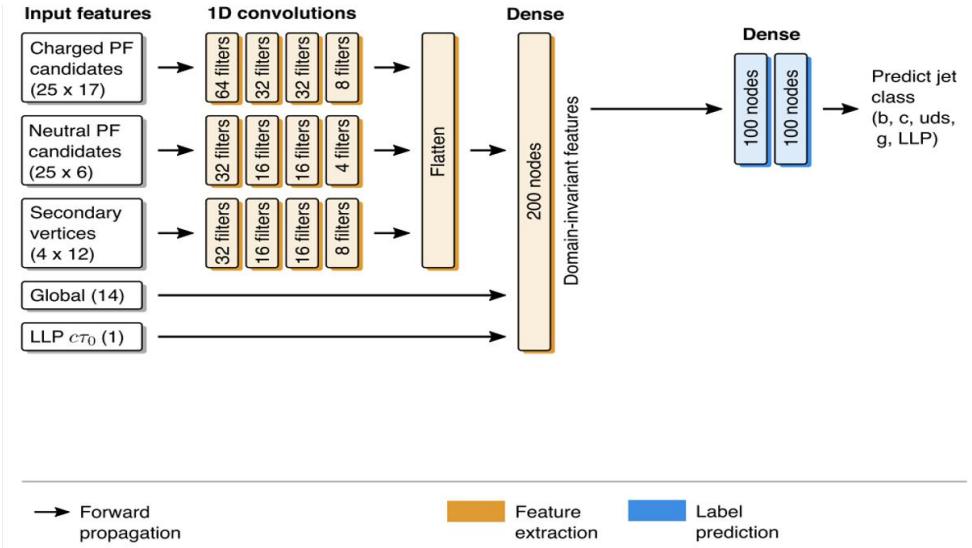


LLP Jet Tagging Algorithm



- Multiclass classification
- Jet class trained on 638 input features per labelled jet
  - Charged and neutral particles, properties of secondary vertices and global jet variables …
- Unsupervised domain adaptation via backpropagation
  - Trained on data and Monte Carlo in control region
- Architecture progressively compresses and extracts most discriminating features
- Resample jet p_T and η to have same distribution for all classes – discrimination only via correlations with other features

LLP Jet Tagging Algorithm

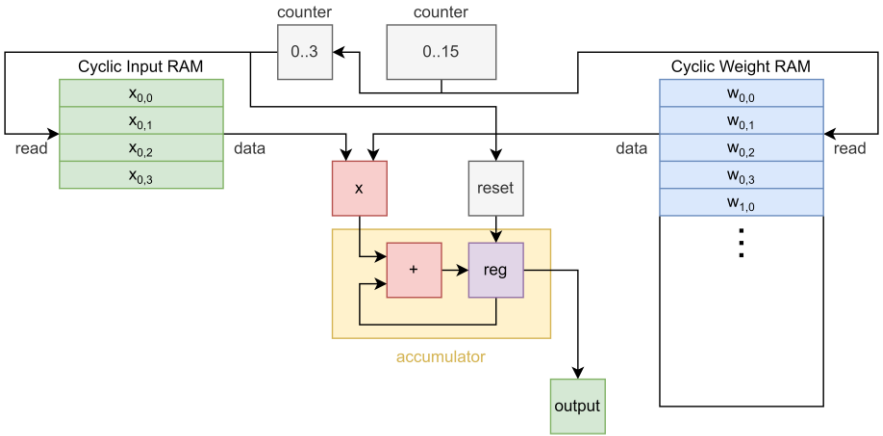


- Multiclass classification
- **Jet class trained on 638 input features per labelled jet**
  - **Charged and neutral particles, properties of secondary vertices and global jet variables …**
- Unsupervised domain adaptation via backpropagation
  - Trained on data and Monte Carlo in control region
- **Architecture progressively compresses and extracts most discriminating features**
- Resample jet p_T and η to have same distribution for all classes – discrimination only via correlations with other features

Imperial College London

Serialised Cyclic RAM Multiply Accumulate

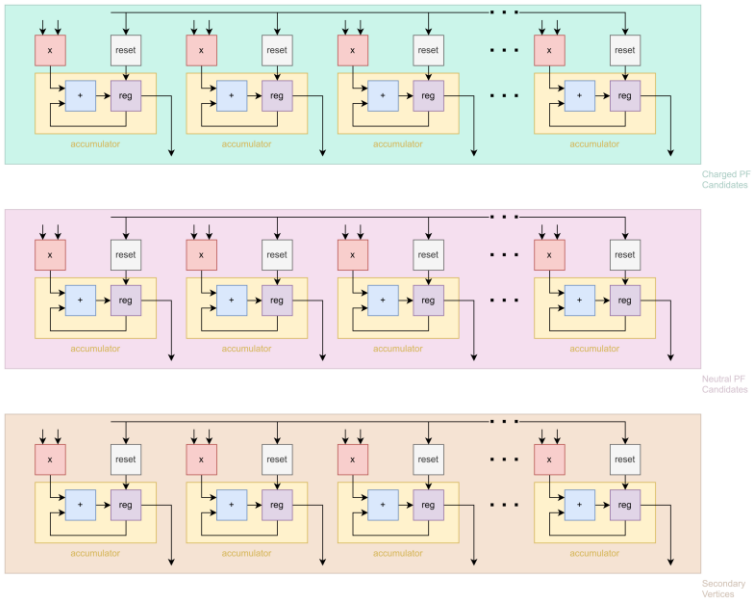
- Cyclic RAM block: RAM with read address connected to a counter
- Resource utilisation is heavily reduced as opposed to conventional convolution method
- Resource saving can be invested in instance duplication to increase throughput.



Imperial College London

Instance Duplication Across Channels

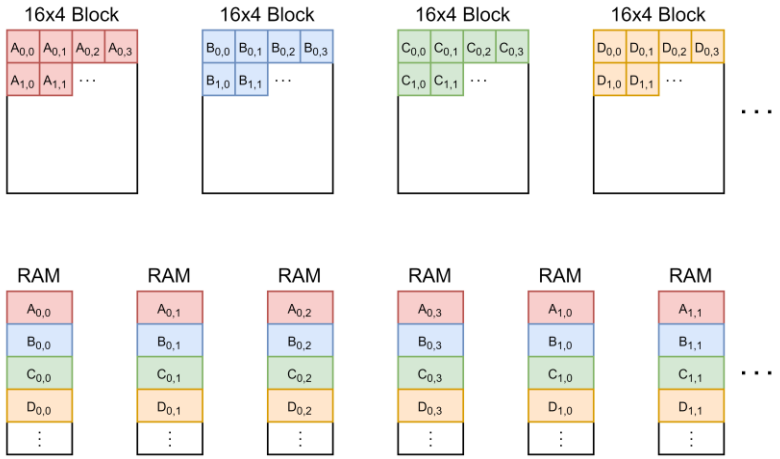
- MAC units can be duplicated across the dimension common throughout the convolution layers
- MAC units can be duplicated across input channels
- Compute in space by taking advantage of computationally independent data streams



Imperial College London

Elementwise Matrix Indexing for RAM Storage

- Matrices usually stored contiguously in memory
- Alternative storage mode: elements in same position in matrices stored in same RAM element
- Faster and simplified read mechanism compared to traditional storage mode



Dataflow and Maxeler

- Dataflow computing involves channelling data between bespoke compute units (kernels)
- Benefit of faster development as opposed to use of Verilog/VHDL and exposure to the rich ecosystem that Java provides
- MaxCompiler (Java based HLS tool) chosen for simplified pipelining and latency control
- Communication between host CPU and FPGA over PCIe interface

Results – Test Setup

- Hardware implementations running on a Xilinx VU9P @ 100MHz
- Resource utilisation comprised of usage of:
  - Look Up Tables (LUT)
  - Flip Flops (FF)
  - Digital Signal Processors (DSP)
  - Block Random Access Memory (BRAM)
- Resource utilisation: $\frac{1}{4}\left(\frac{Used\ LUT}{Total\ LUT}+\frac{Used\ FF}{Total\ FF}+\frac{Used\ DSP}{Total\ DSP}+\frac{Used\ BRAM}{Total\ BRAM}\right)\times 100\%$

Results – Serialised Cyclic RAM Multiply Accumulate

Hardware Design	Fully Unrolled	Partial Unroll	Single MAC Unit	Column MAC Units
f (MHz)	100	100	350	250
Latency (ns)	**80.00**	**210.00**	**140.00**	**24.00**
LUT (%)	26.86	13.24	1.08	17.31
FF (%)	21.61	16.88	1.07	19.22
DSP (%)	29.94	1.86	0.06	0.94
BRAM (%)	8.24	3.11	2.36	2.98
RU (%)	**21.66**	**8.77**	**1.14**	**10.11**

Results – Elementwise Matrix Indexing for RAM Storage

Hardware Design	Single Buffer	Double Buffer	Elementwise RAM Storage
f (MHz)	100	100	100
Latency (ns)	**100.00**	**80.00**	**20.00**
LUT (%)	1.65	3.38	0.71
FF (%)	6.43	13.22	2.90
DSP (%)	0.00	0.00	0.00
BRAM (%)	9.28	18.56	8.12
RU (%)	**4.34**	**8.77**	**1.14**

Comparisons to Prior Implementations

- C++ implementation ran on Intel Xeon Gold 6154 CPU @ 3.00 GHz
- Took 780ns to convolve two 16x4 input matrices
- Fastest FPGA implementation offers ~32.5x latency reduction
- FPGAs offer a viable route to online processing, especially after HL-LHC upgrade