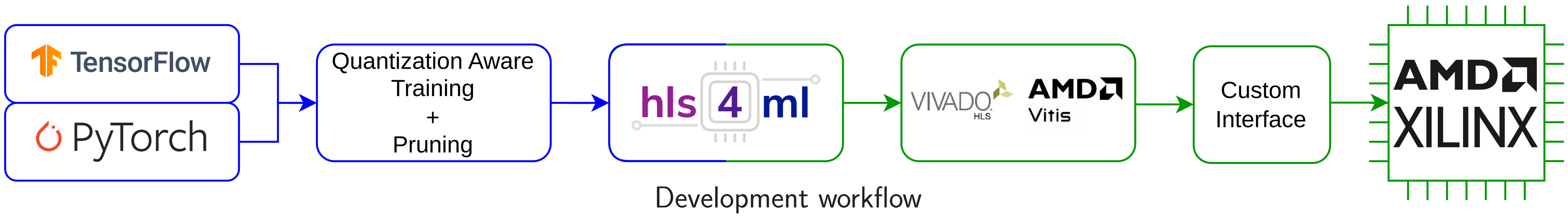
Design and implementation of Neural Network based conditions for the CMS Level-1 Global Trigger upgrade for the HL-LHC

Gabriele Bortolato[1,2], Maria Cepeda[3], Jaana Heikkilä[4], Benjamin Huber[1,5], Elias Leutgeb[1,5], Dinyar Rabady[1], Hannes Sakulin[1] on behalf of the CMS Collaboration

[1]CERN, [2]Universitá degli Studi di Padova, [3]CIEMAT, [4] Universität Zürich, [5] Technische Universität Wien

Overview

At the CMS experiment, a two-layer trigger system is used to decide which collision events to store for later analysis. To ensure the physics performance is maintained or even improved under the new high-luminosity conditions during Phase-2 operation, the CMS Level-1 Trigger is being entirely redesigned. Besides cut-based triggers, the Global Trigger will also apply novel machine-learning-based conditions on trigger objects identified by the upstream systems. These triggers rely on the full event topology to trigger on previously inaccessible events.



Development workflow

Neural Network development workflow

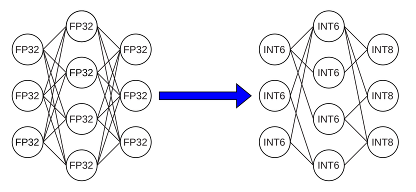
Step 1: Model definition

Model definition and training with the commonly used frameworks.

TensorFlow PyTorch

Step 2: Optimizations

Hyperparameter quantization, connection pruning and knowledge distillation.



Step 3: FPGA porting

Python model translation to HLS and finally to FPGA language [1]

hls 4 ml AMD Vitis

Step 4: Interfaces — deployment

Interface layers to adjust input and output data format. Deployment using Vivado on the target FPGA.

AMD Vivado

From high level (Python) to hardware level (VHDL/Verilog) language to FPGA fabric.

Anomaly detection vs. signature based models

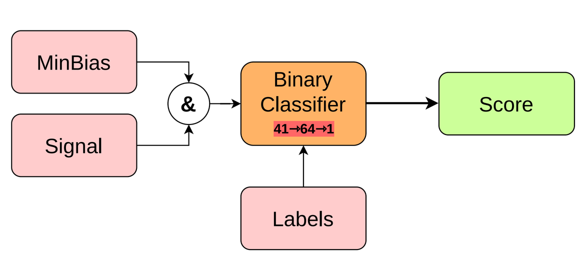
Two different flavours of neural networks are considered: deep binary classifiers and deep auto-encoders. The first is designed to distinguish a specific signal signature, while the second aims to characterize as much as possible the background and identify anything that does not resemble it marking it as anomalous.

As proof of principle four different signal signatures were considered:

- Minimum bias (as background)
- HH→2b2τ
- VBF→ $\tau\tau$
- $t\bar{t}$ decay

Binary classifier approach

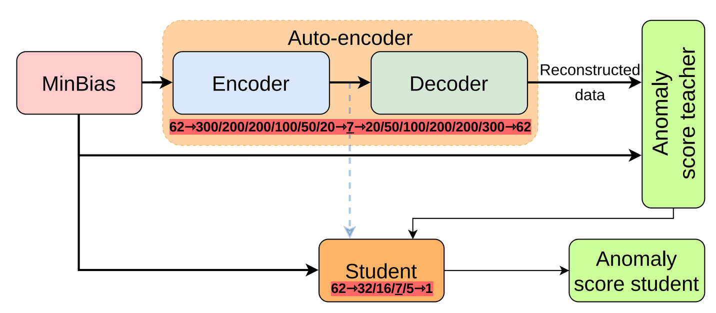
L1T Objects	Subsystem	Variables	
First 6 jets	CL2	p_T	η
First 4 electrons	CL2	p_T	η
First 4 muons	GMT	p_T	η
First 2 taus	CL2	p_T	η
Missing energy	CL2	E_T^{miss}	-



Supervised training: background and signal labels are known from the start

Auto-encoder approach

L1T Objects	Subsystem	Variables		
First 6 jets	CL2	p_T	η	ϕ
First 4 electrons	CL2	p_T	η	ϕ
First 4 muons	GMT	p_T	η	ϕ
First 2 taus	CL2	p_T	η	ϕ
Missing energy	CL2	E_T^{miss}	-	ϕ

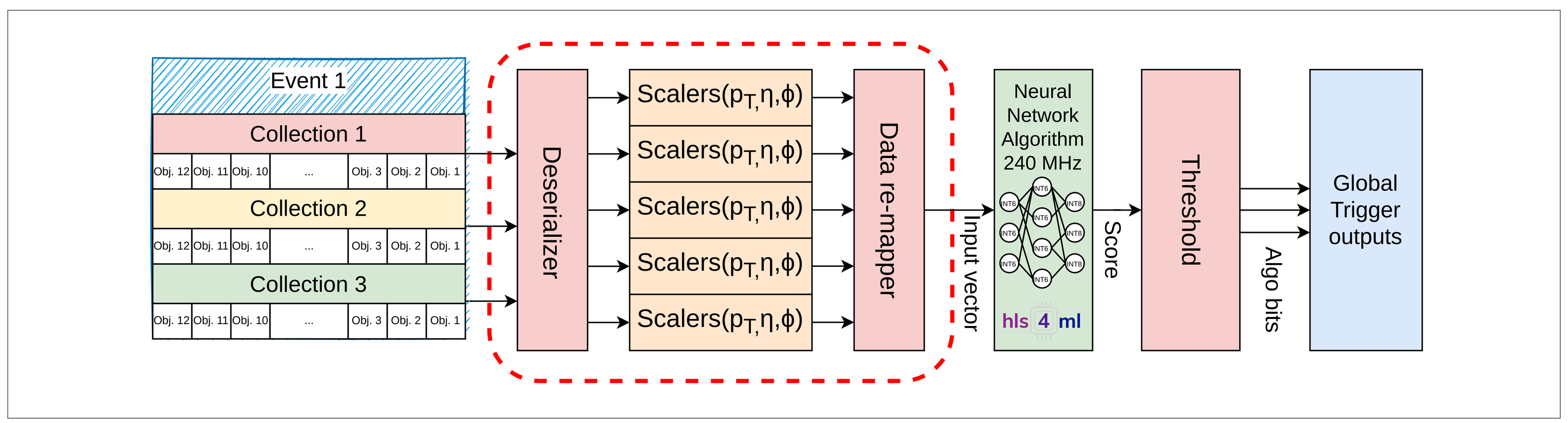


Unsupervised training + knowledge distillation: Teacher is trained with only the background, while the student uses background and random samples

Multiple optimizations take place during and after training: hyperparameter quantization, pruning of synapses, knowledge distillation (only for auto-encoder) and input selection. Each signal signature requires its own trained binary classifier model, while the auto-encoder model is trained with only the minimum bias sample and for this reason it's model independent.

Custom interface to the Phase-2 Global Trigger framework

Serial data from upstream systems is streamed at 480 MHz in collections of 12 objects. These data need to be deserialized, re-scaled and re-mapped in order to be fed into the NN module resulting in one wide bit-vector every 25 ns. NN block runs at 240 MHz, which is a good compromise between register usage and latency.



The input interface module is entirely written in VHDL and it's model specific, e.g. bitwidth, number of inputs and re-scale parameters.

Model Evaluation

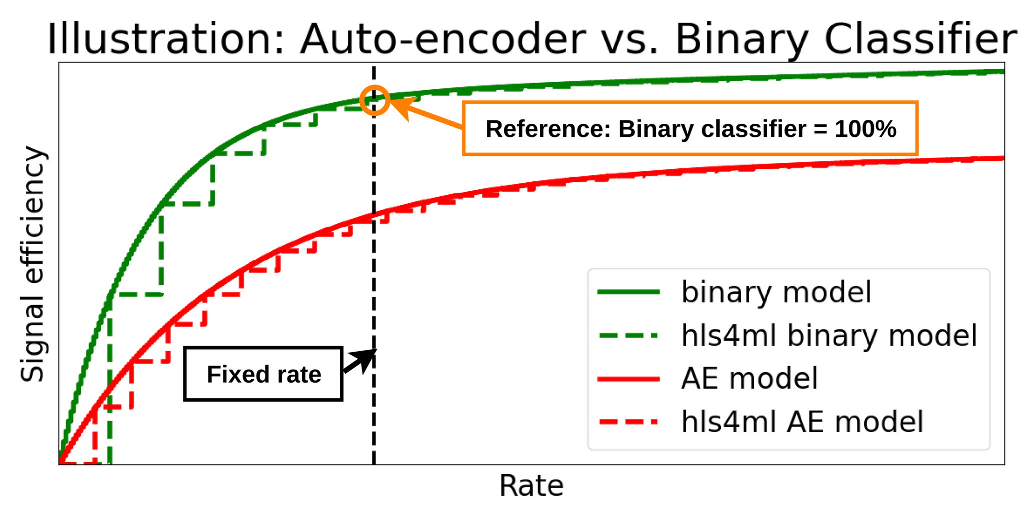


Illustration: the binary classifier efficiency at a given rate is taken as reference, while the auto-encoder efficiency is expressed relative to it.

Model	Framework	Prune	Quant[1]	LUT[k]	FF[k]	DSP	Lat [ns]	Eff/Eff$_{BinaryBaseline}$ HH	$t\bar{t}$	VBF
Baseline AE	TensorFlow	0%	FP32	-	-	-	-	70.6%	60.7%	36.7%
hls4ml AE	hls4ml	50%	<8,1/2>[2]	42	15	301	70.8	70.5%	60.7%	36.7%
Baseline HH	TensorFlow	0%	FP32	-	-	-	-	100.0%	-	-
hls4ml HH	hls4ml	50%	<6/8,1/4>[2]	4.6	2.3	19	33.3	98.3%	-	-
Baseline $t\bar{t}$	TensorFlow	0%	FP32	-	-	-	-	-	100.0%	-
hls4ml $t\bar{t}$	hls4ml	50%	<6/8,1/4>[2]	5.4	2.4	20	33.3	-	98.9%	-
Baseline VBF	TensorFlow	0%	FP32	-	-	-	-	-	-	100.0%
hls4ml VBF	hls4ml	50%	<6/8,1/4>[2]	7.7	3.4	45	33.3	-	-	95.0%

[1]In terms of <total,integer> bit width; [2] Weights and biases have two different quantizations

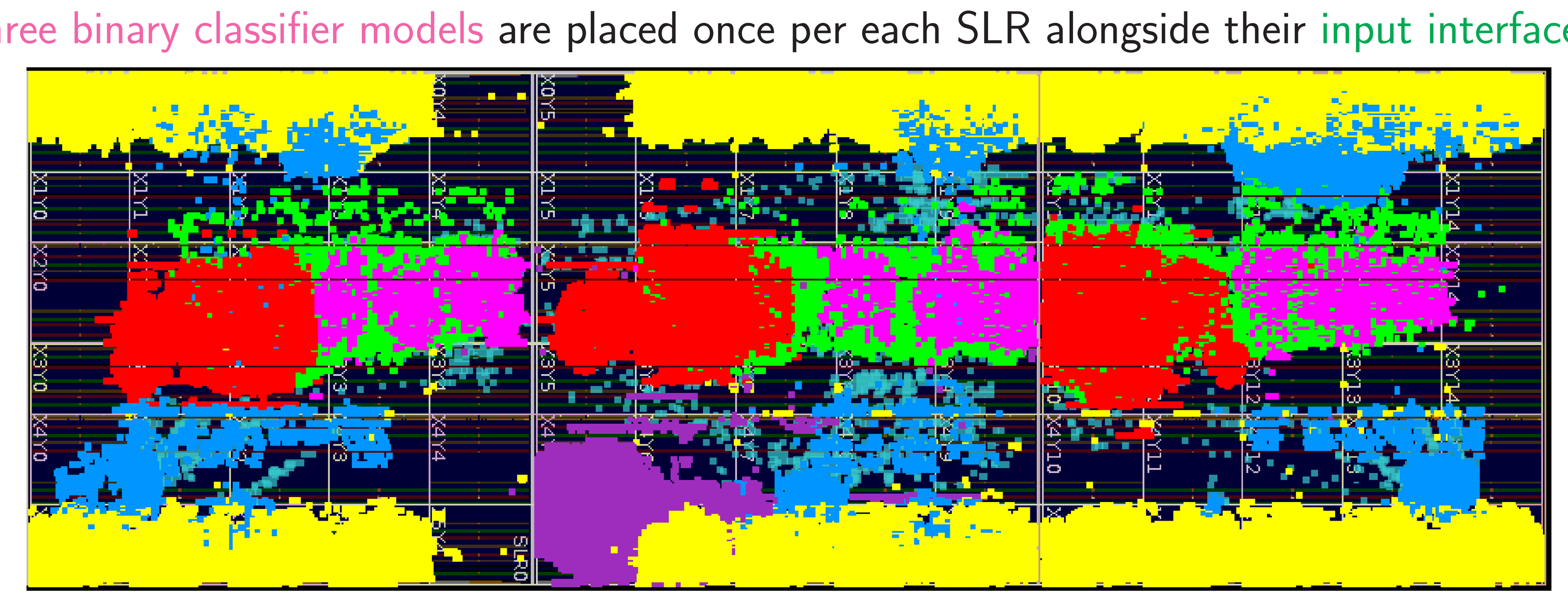
Hardware implementation

The neural network block is deployed on a Serenity [2] board equipped with a Virtex Ultrascale+ (VU9P) FPGA.

The neural-network based algorithms have been integrated in the Global Trigger (GT) pre-production firmware [3] that is based on the EMP framework [4].

Site Type	Synth	Impl
CLB LUTs	218k (22%)	320k (27%)
CLB Regs	509k (22%)	452k (19%)
BRAM	475 (22%)	723 (33%)
DSPs	150 (2%)	1290 (19%)

The GT firmware demultiplexes data received from EMP data region buffers and distributes the data collections to all SLRs. For testing purposes one anomaly detection trigger and the three binary classifier models are placed once per each SLR alongside their input interfaces.



GT demultiplexers and distribution Anomaly detection EMP TTC & DMA
Neural Network interface Binary classifiers EMP link buffers

Reference

[1] Javier Duarte et al. "*Fast inference of deep neural networks in FPGAs for particle physics*", DOI: 10.1088/1748-0221/13/07/P07027

[2] Andrew Rose et al. "*Serenity: An ATCA prototyping platform for CMS Phase-2*", DOI: 10.22323/1.343.0115

[3] Hannes Sakulin et al. "*Architecture and prototype of the CMS Global Level-1 Trigger for Phase-2*", DOI: 10.1088/1748-0221/18/01/C01034

[4] *EMP Framework* `https://serenity.web.cern.ch/serenity/emp-fwk/`

Contacts

gabriele.bortolato@cern.ch   cms-l1t-p2gt@cern.ch