Real-time AI for HEP: a few applications

Jennifer Ngadiuba (Fermilab)

51st SLAC Summer Institute
August 7-18, 2023
Stanford National Accelerator Laboratory

Fermilab

FastML Lab
Real-time AI for HEP: a few applications

Supernova detection with neutrinos in Liquid Argon experiments

Anomaly detection in LHC hardware trigger & other new approaches
Supernova detection with neutrinos in Liquid Argon experiments
Multi-messenger astronomy probes the Universe using different cosmic messengers.

- **X-rays/Gamma-rays**
- **Visible/Infrared light**
- **Cosmic event** (ex, supernovae)
- **Gravitational waves**
- **Radio telescope**
- **Neutrinos**
Multi-messenger astronomy probes the Universe using different cosmic messengers

- **neutron star merger (GW170817):**
  gravitational wave (Ligo/Virgo) + electromagnetic signal (Fermi and INTEGRAL telescopes)

- **blazar (TXS 0506+056):**
  high-energy neutrino (IceCube) + electromagnetic signals (Fermi, MAGIC and others)

Multi-messenger astronomy

Two notable examples:

- **neutron star merger (GW170817):**
  gravitational wave (Ligo/Virgo) + electromagnetic signal (Fermi and INTEGRAL telescopes)

- **blazar (TXS 0506+056):**
  high-energy neutrino (IceCube) + electromagnetic signals (Fermi, MAGIC and others)
Multi-messenger astronomy

Multi-messenger astronomy probes the Universe using different cosmic messengers

- **Two notable examples:**
  - **neutron star merger (GW170817):**
    gravitational wave (Ligo/Virgo) + electromagnetic signal (Fermi and INTEGRAL telescopes)
  - **blazar (TXS 0506+056):**
    high-energy neutrino (IceCube) + electromagnetic signals (Fermi, MAGIC and others)

- **Timing and pointing accuracy** crucial to deliver alerts for a potential cosmological event to many different instruments around the globe

- This to become challenging with increase in size and sensitivity to a larger volume of space
Multi-messenger astronomy w/ Neutrinos

- Core-collapse supernovae are a huge source of neutrinos of all flavours
  - 99% of energy released is carried away by neutrinos

- Rich information embedded in neutrino signal plus associated gravitational and electromagnetic signals
  - *supernova physics*: core-collapse mechanism, black hole formation, nucleosynthesis, …
  - *particle physics*: flavor transformation in SN core, mass ordering, BSM…

- Detection and pointing in real-time in large scale neutrino experiments is an active field of research!
Multi-messenger astronomy w/ Neutrinos

The Deep Underground Neutrino Experiment (DUNE)

- Next generation neutrinos oscillation experiment now under construction and R&D to start operations by the end of current decade

- Massive far detector 1 mile underground comprising 70k tons of LAr and advanced technology (LAr Time Projection Chambers) to record neutrino interactions with extraordinary precision
The DUNE far detector

Operating principle of a LArTPC

Electrons are produced by charged particles interacting with a large multiple-cubic meters volume of LAr.

Continuous stream of 3D images of detector volume yielding a high-resolution “video”:

- 4 modules x 150 cell volumes
- O(10) MB / frame
- O(10^5) frames / s for 2.25 ms (drift time)
- a total of ~40 Tb/s
Data reduction @ DUNE

- Trigger decision starts underground to achieve x10 data reduction before transfer to surface.
- Half of 150 cells processed in parallel in custom low power Xilinx FPGA board.
- Coarse first level of filtering on a per-cell basis (~ threshold based decision per wire).
- Second level aggregates low-level information from all cells in a single module to make a module-level trigger decision:
  - executed on CPU resources with O(s) latency.
  - Positive decision initiate readout of 2.25 ms worth of continuous data from all 150 cells.
Challenges for supernova detection

- Low-energy neutrinos from SN result in tiny signals hardly distinguishable from electronic noise or radiological background (e.g., 39Ar decays)
  - threshold based triggering suboptimal resulting in either low SN detection efficiency or too high background acceptance
  - self-triggering mode at 99% SN detection efficiency and less than one per month false positive

- Orders of magnitude more buffering and processing for a supernova burst trigger, which looks for correlated signatures in O(10) seconds!
  - aggressive background reduction needed for transfer to the surface and downstream processing
SN neutrinos interactions

• Neutrino-electron elastic scattering $\nu$-ES: $\nu_x + e^- \rightarrow \nu_x + e^-$

  - most relevant for directional information

  - the direction of the scattered electron is highly correlated to the neutrino direction

• Electron-neutrino charged-current interactions $\nu$-CC: $\nu_e + ^{40}\text{Ar} \rightarrow e^- + ^{40}\text{K}^*$

  - not interesting for supernova pointing as electron trajectory correlates only weakly with neutrino direction

  - $\sim x10$ times more frequent than $\nu$-ES $\rightarrow$ downstream processing gain if rejected in first trigger level
SN neutrinos detection & identification

- Tested performance of vanilla 2D CNN for classification task on one plane: \( \nu\text{-ES vs } \nu\text{-CC vs background} \) (radiological and electronic)

- Several approaches being studied including per-wire and per-image denoising AE
  - can be combined with YOLO-type of model to perform data reduction
  - best model to be chosen based on performance of downstream pointing algorithm

- Work in progress — first results being approved by the DUNE Collab to be presented at the Fast ML workshop in September
Fast CNNs in hls4ml

- With hls4ml CNNs can be made extremely fast for deployment on FPGA
  - optimal for underground data processing for DUNE (limited power budget)
  - alternatively standard trigger algos can run underground and ML processing on surface on GPUs also possible (trade off)

Example for 3 Conv layers on 32x32x3 input image (SVHN dataset)

~ 5 µs inference time!
Anomaly detection in the LHC hardware triggers & other new approaches
Data reduction @ LHC

99.75% events rejected!
- Hardware based
- Runs on FPGAs in real time
- $O(\mu s)$ latency

99% events rejected!
- Software based
- Runs on CPUs in real time
- $O(100 \text{ ms})$ latency

Software based
- Run on data centres (LHC grid)
- $O(s)$ computing time

Data volume
- L1 trigger
- 40 MHz

High-Level trigger
- 100 KHz

Offline reconstruction
- 1 KHz
- 1 MB/evt
- 3-30 kB/evt

Computing time

No data stored before this line

Run by individuals on the grid
Ex: the CMS trigger system in Run 3

~ 100 rule-based Global Trigger algos: designed for high acceptance of SM and BSM signals of interests

Muon detectors

- CSC TPs
- RPC Hits
- DT TPs
- HBHE TPs
- HF TPs
- ECAL TPs

- Muon Port Card
- Link Board
- CPPF
- TwinMux

- Endcap Muon Track Finder
- Overlap Muon Track Finder
- Barrel Muon Track Finder

- Layer 1 Calorimeter Trigger
- Layer 2 Calorimeter Trigger

- Global Muon Trigger
- DeMux

- 8 muons

Calorimeters

- 4.2 us

12 jets
12 e/gamma
12 taus

Energy sums

- $E_T^{miss} > 100$ (Vector sum of $p_T$ of all calorimeter deposits with $|\eta| < 5.0$)
- $H_T > 360$ (Scalar sum of $p_T$ of all jets with $|\eta| < 2.5$)
- $E_T > 2000$ (Scalar sum of $p_T$ of all calorimeter deposits with $|\eta| < 3.0$)
Ex: the CMS trigger system in Run 3

- With 40M collisions/seconds and 1000 stored, we might just be writing the wrong events

  - trigger algorithms quite model dependent (mostly high $p_T$/energy)

  - any other signature we did not think about could have easily be discarded

~ 100 rule-based Global Trigger algos: designed for high acceptance of SM and BSM signals of interests

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Requirements (p_T, E, m_pT, and m_q in GeV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Muons</td>
<td></td>
</tr>
<tr>
<td>Single $\mu$</td>
<td>$p_T &gt; 22$ &amp; Tight quality</td>
</tr>
<tr>
<td>Double $\mu$</td>
<td>$p_T &gt; 15$ &amp; Medium quality</td>
</tr>
<tr>
<td>Double $\mu$</td>
<td>$p_T &gt; 15$ &amp; Tight quality</td>
</tr>
<tr>
<td>Double $\mu$</td>
<td>$p_T &gt; 8$ &amp; Tight quality</td>
</tr>
<tr>
<td>Double $\mu + \text{mass}$</td>
<td>$p_T &gt; 4.5$ &amp; $</td>
</tr>
<tr>
<td>Double $\mu + \Delta R$</td>
<td>$p_T &gt; 4$ &amp; Tight quality &amp; OS &amp; $\Delta R &lt; 1.2$</td>
</tr>
<tr>
<td>Double $\mu + \Delta R$</td>
<td>$p_T &gt; 0$ &amp; $</td>
</tr>
<tr>
<td>Double $\mu + \text{BX}$</td>
<td>$p_T &gt; 0$ &amp; $</td>
</tr>
<tr>
<td>Triple $\mu$</td>
<td>$p_T &gt; 5$ &amp; Medium quality</td>
</tr>
<tr>
<td>Triple $\mu$</td>
<td>$p_T &gt; 3.3$ &amp; Tight quality</td>
</tr>
<tr>
<td>Triple $\mu + \text{mass}$</td>
<td>$p_T &gt; 5, 3, 5, 2.5$ &amp; Med. qual.; two $\mu$ OS &amp; $p_T &gt; 5, 2.5$ &amp; 5 &lt; $m_{\mu}$ &lt; 17</td>
</tr>
<tr>
<td>Triple $\mu + \text{mass}$</td>
<td>Three $\mu$ any qual.; two $\mu$ &amp; $p_T &gt; 5, 3$ &amp; Tight qual. &amp; OS &amp; $m_{\mu}$ &lt; 9</td>
</tr>
<tr>
<td>Electrons / photons ($e/\gamma$)</td>
<td></td>
</tr>
<tr>
<td>Single $e/\gamma$</td>
<td>$p_T &gt; 60$</td>
</tr>
<tr>
<td>Single $e/\gamma$</td>
<td>$p_T &gt; 36$ &amp; $</td>
</tr>
<tr>
<td>Single $e/\gamma$</td>
<td>$p_T &gt; 28$ &amp; $</td>
</tr>
<tr>
<td>Double $e/\gamma$</td>
<td>$p_T &gt; 25$ &amp; $</td>
</tr>
<tr>
<td>Double $e/\gamma$</td>
<td>$p_T &gt; 22$ &amp; $</td>
</tr>
<tr>
<td>Triple $e/\gamma$</td>
<td>$p_T &gt; 18, 17, 18$ &amp; $</td>
</tr>
<tr>
<td>Triple $e/\gamma$</td>
<td>$p_T &gt; 16, 16, 16$ &amp; $</td>
</tr>
<tr>
<td>Tau leptons ($\tau$)</td>
<td></td>
</tr>
<tr>
<td>Single $\tau$</td>
<td>$p_T &gt; 120$ &amp; $</td>
</tr>
<tr>
<td>Double $\tau$</td>
<td>$p_T &gt; 32$ &amp; $</td>
</tr>
<tr>
<td>Jets</td>
<td></td>
</tr>
<tr>
<td>Single jet</td>
<td>$p_T &gt; 150$</td>
</tr>
<tr>
<td>Single jet + BX</td>
<td>$p_T &gt; 112$ &amp; $</td>
</tr>
<tr>
<td>Double jet</td>
<td>$p_T &gt; 110, 110$ &amp; two jets $p_T &gt; 35$ &amp; $m_T &gt; 620$</td>
</tr>
<tr>
<td>Double jet + mass</td>
<td>$p_T &gt; 30$ &amp; $</td>
</tr>
<tr>
<td>Triple jet</td>
<td>$p_T &gt; 95, 75, 65$; two jets $p_T &gt; 75, 65$ &amp; $</td>
</tr>
<tr>
<td>Energy sums</td>
<td></td>
</tr>
<tr>
<td>$E_T^{\text{miss}}$</td>
<td>$E_T^{\text{miss}} &gt; 100$ (Vector sum of $p_T$ of calorimeter deposits with $</td>
</tr>
<tr>
<td>$H_T$</td>
<td>$H_T &gt; 360$ (Scalar sum of $p_T$ of all jets with $p_T &gt; 30$ and $</td>
</tr>
<tr>
<td>$E_T$</td>
<td>$E_T &gt; 2000$ (Scalar sum of $p_T$ of calorimeter deposits with $</td>
</tr>
</tbody>
</table>
Ex: the CMS trigger system in Run 3

- With 400M collisions/seconds and 1000 stored, we might just being writing the wrong events
  - trigger algorithms quite model dependent (mostly high $p_T$/energy)
  - any other signature we did not think about could have easily be discarded

Hardware trigger as the ultimate limiting factor for searches!

  - Anomaly detection can potentially mitigate this with signal-independent background rejection
  - We want to isolate any type of anomalous event (not restricted to jet substructure or dijet events) → take as input full event information as available to the global trigger
  - The hls4ml tool allows us to run ML-based AD on the FPGAs mounted on the L1T boards

~ 100 rule-based Global Trigger algs: designed for high acceptance of SM and BSM signals of interests

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Requirements (p$<em>T$, $E_T$, $m</em>{\mu\mu}$, and $m_\mu$ in GeV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Muons</td>
<td>$p_T &gt; 22$ &amp; Tight quality</td>
</tr>
<tr>
<td>Double $\mu$</td>
<td>$p_T &gt; 15, 7$ &amp; Medium quality</td>
</tr>
<tr>
<td>Double $\mu$</td>
<td>$p_T &gt; 15.5$ &amp; Tight quality</td>
</tr>
<tr>
<td>Double $\mu$</td>
<td>$p_T &gt; 8, 8$ &amp; Tight quality</td>
</tr>
<tr>
<td>Double $\mu +$</td>
<td>$p_T &gt; 4.5$ &amp; $</td>
</tr>
<tr>
<td>Double $\mu +$</td>
<td>$p_T &gt; 4$ &amp; Tight quality &amp; OS &amp; $\Delta R &lt; 1.2$</td>
</tr>
<tr>
<td>Double $\mu +$</td>
<td>$p_T &gt; 0$ &amp; $</td>
</tr>
<tr>
<td>Double $\mu +$</td>
<td>$p_T &gt; 0$ &amp; $</td>
</tr>
<tr>
<td>Triple $\mu$</td>
<td>$p_T &gt; 5, 3, 3$ &amp; Medium quality</td>
</tr>
<tr>
<td>Triple $\mu +$</td>
<td>$p_T &gt; 3, 3, 3$ &amp; Tight quality</td>
</tr>
<tr>
<td>Triple $\mu +$</td>
<td>$p_T &gt; 5, 3, 5, 2$ &amp; Medium Quality; two $\mu$ OS &amp; $p_T &gt; 5, 2, 5$ &amp; $5 &lt; m_{\mu\mu} &lt; 17$</td>
</tr>
<tr>
<td>Triple $\mu +$</td>
<td>Three $\mu$ any qual; two $\mu$ &amp; $p_T &gt; 5, 3$ &amp; Tight qual. &amp; OS &amp; $m_{\mu\mu} &lt; 9$</td>
</tr>
<tr>
<td>Electrons/photon (e/γ)</td>
<td>$p_T &gt; 60$</td>
</tr>
<tr>
<td>Single e/γ</td>
<td>$p_T &gt; 36$ &amp; $</td>
</tr>
<tr>
<td>Single e/γ</td>
<td>$p_T &gt; 28$ &amp; $</td>
</tr>
<tr>
<td>Double e/γ</td>
<td>$p_T &gt; 25, 12$ &amp; $</td>
</tr>
<tr>
<td>Double e/γ</td>
<td>$p_T &gt; 22, 12$ &amp; $</td>
</tr>
<tr>
<td>Triple e/γ</td>
<td>$p_T &gt; 18, 17, 8$ &amp; $</td>
</tr>
<tr>
<td>Triple e/γ</td>
<td>$p_T &gt; 16, 16, 16$ &amp; $</td>
</tr>
<tr>
<td>Tau leptons (τ)</td>
<td>$p_T &gt; 120$ &amp; $</td>
</tr>
<tr>
<td>Single τ</td>
<td>$p_T &gt; 32$ &amp; $</td>
</tr>
<tr>
<td>Jets</td>
<td>$p_T &gt; 180$</td>
</tr>
<tr>
<td>Single jet</td>
<td>$p_T &gt; 43$ &amp; $</td>
</tr>
<tr>
<td>Single jet</td>
<td>$p_T &gt; 150$ &amp; $</td>
</tr>
<tr>
<td>Double jet</td>
<td>$p_T &gt; 112$ &amp; $</td>
</tr>
<tr>
<td>Double jet + mass</td>
<td>$p_T &gt; 110, 3$ &amp; two jets $p_T &gt; 35$ &amp; $m_{\mu\mu} &gt; 620$</td>
</tr>
<tr>
<td>Double jet + mass</td>
<td>$p_T &gt; 30$ &amp; $</td>
</tr>
<tr>
<td>Triple jet</td>
<td>$p_T &gt; 95, 75, 65$; two jets $p_T &gt; 75, 65$ &amp; $</td>
</tr>
<tr>
<td>Energy sums</td>
<td>$E_{\text{EM}} &gt; 100$ (Vector sum of $p_T$ of calorimeter deposits with $</td>
</tr>
<tr>
<td>$H_T$</td>
<td>$H_T &gt; 360$ (Scalar sum of $p_T$ of all jets with $p_T &gt; 30$ and $</td>
</tr>
<tr>
<td>$E_T$</td>
<td>$E_T &gt; 2000$ (Scalar sum of $p_T$ of calorimeter deposits with $</td>
</tr>
</tbody>
</table>
Autoencoders in a nutshell

- Compression-decompression algorithm that learns to describe a given dataset in terms of points in a lower-dimension latent space

- Unsupervised learning method that can be used for anomaly detection (but not only)
  - train on a reference data sample
  - when applied on new data of different kind compression-decompression might fail
  - define an anomaly score in some metric quantifying how “far” the decompressed output is from the input (e.g., the reconstruction loss)

\[ \mathcal{L}_{reco} = \|x - \hat{x}\|^2 = MSE(input, output) \]
Variational autoencoders for jets

- Instead of encoding an input as a single point, encode it as a distribution over the latent space
- The encoded distributions are chosen to be normal → encoder trained to return the mean ($\mu$) and the covariance matrix ($\sigma$) describing the multi-dim Gaussian

$$\text{Loss}_{\text{Tot}} = \text{Loss}_{\text{reco}} + \beta D_{\text{KL}}$$

$$D_{\text{KL}} = \frac{1}{k} \sum_i D_{\text{KL}}(N(\mu^i_z, \sigma^i_z) \parallel N(\mu_P, \sigma_P))$$

$$= \frac{1}{2k} \sum_{i,j} \left( \sigma^i_P \sigma^i_z \right)^2 + \left( \frac{\mu^i_P - \mu^i_z}{\sigma^i_P} \right)^2 + \ln \frac{\sigma^i_P}{\sigma^i_z} - 1$$

$$\mu_P = 0, \sigma_P = 1$$

$$\mathcal{L}_{\text{reco}} = \| x - \hat{x} \|^2 = \text{MSE}(\text{input}, \text{output})$$
Ultra-fast autoencoders

• We start from a cocktail of standard model processes that populate most of the high-energy spectrum of a standard LHC data stream after a single-lepton requirement (for simplicity)

  - simulation of a realistic unbiased 40 MHz stream would be too computing resource intensive

Sample mainly consists of \( W, Z, \text{tt} & \text{QCD} \)

BSM signals to estimate AD algo potential

<table>
<thead>
<tr>
<th>Standard Model processes</th>
<th>Acceptance</th>
<th>L1 trigger efficiency</th>
<th>Cross section [nb]</th>
<th>Event fraction</th>
<th>Events /month</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W )</td>
<td>55.6%</td>
<td>68%</td>
<td>58</td>
<td>59.2%</td>
<td>110M</td>
</tr>
<tr>
<td>QCD</td>
<td>0.08%</td>
<td>9.6%</td>
<td>( 1.6 \cdot 10^5 )</td>
<td>33.8%</td>
<td>63M</td>
</tr>
<tr>
<td>( Z )</td>
<td>16%</td>
<td>77%</td>
<td>20</td>
<td>6.7%</td>
<td>12M</td>
</tr>
<tr>
<td>( \text{tt} )</td>
<td>37%</td>
<td>49%</td>
<td>0.7</td>
<td>0.3%</td>
<td>0.6M</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BSM benchmark processes</th>
<th>Acceptance</th>
<th>L1 trigger efficiency</th>
<th>Total efficiency 100 BSM events/month</th>
<th>Cross-section</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A \rightarrow 4\ell )</td>
<td>5%</td>
<td>98%</td>
<td>5%</td>
<td>0.44 pb</td>
</tr>
<tr>
<td>( LQ \rightarrow b\tau )</td>
<td>19%</td>
<td>62%</td>
<td>12%</td>
<td>0.17 pb</td>
</tr>
<tr>
<td>( h^0 \rightarrow \tau\tau )</td>
<td>9%</td>
<td>70%</td>
<td>6%</td>
<td>0.34 pb</td>
</tr>
<tr>
<td>( h^\pm \rightarrow \tau\nu )</td>
<td>18%</td>
<td>69%</td>
<td>12%</td>
<td>0.16 pb</td>
</tr>
</tbody>
</table>

• \( m_A = 50 \text{ GeV} \)
• \( m_{LQ} = 80 \text{ GeV} \)
• \( m_{h^0} = 60 \text{ GeV} \)
• \( m_{h^\pm} = 60 \text{ GeV} \)
Ultra-fast autoencoders

- We use a **momentum-based data representation** to mimic the typical information available in the L1 trigger
  - avoid need of computing high-level features at L1 which can be time or resource consuming
  - number of objects chosen to emulate limited L1 bandwidth
  - zero padding if less objects are found

<table>
<thead>
<tr>
<th></th>
<th>$p_T$</th>
<th>$\eta$</th>
<th>$\varphi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MET</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>4 $e/\gamma$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 $\mu$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 jets</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Ultra-fast autoencoders

• We compare different architectures: CNN vs DNN and autoencoders versus variational AE

• Need to be careful with VAE → total loss as metric somehow sub-optimal
  - random sampling not practical on FPGA hardware (need to store full distributions, keep track of seeds and sampled values, etc…)
  - trigger decision required to be reproducible (match offline emulator)
  - non-determinist decision might generally encounter resistance
Ultra-fast autoencoders

**ALTERNATIVE APPROACH:**

- Train encoder+decoder with $L_{tot} = (1 - \beta) \cdot L_{reco} + \beta \cdot D_{KL}(\mu, \sigma)$
- Define an AD figure of merit in the latent space $D_{KL}(\mu, \sigma)$ or $R_z = \sum_i (\mu_i / \sigma_i)^2$
- Advantages for L1 trigger application:
  - no sampling at inference
  - save resources and latency by not running decoder at inference

![Diagram of autoencoder architecture](image)
Ultra-fast autoencoders

- **For the DNN:** $\text{MSE}_{\text{VAE}} \approx \text{MSE}_{\text{AE}} \approx \text{D}_{\text{KL}} \rightarrow$ can run only encoder @ L1 without loss in performance

- **For the CNN:** VAE performing better than AE at low FPR but loss in performance when using $R_Z$ or $D_{\text{KL}}$ strategies

![Dense NN ROC A → 4l](image1)

![Convolutional NN ROC A → 4l](image2)
Ultra-fast autoencoders

With no signal prior, enhancement of 2-3 orders of magnitude in sample purity wrt standard cut-based trigger algorithms while keeping low output rate!

Gain comes from AD ability to largely reduce low-energy background while retaining a sensible fraction of low-energy signals

Standard threshold based trigger cannot achieve this without significantly reducing signals as well → rely on powerful offline algorithms if signal has not been thrown out completely

FPR = $10^{-5}$ → threshold for comparing performance (~1000 events/month)
Ultra-fast autoencoders

- Perform **compression** with Tensorflow Pruning API during training, targeting 50% sparsity → baseline pruned model
  - obtain similar performance as for the unpruned model

- Perform **quantization-aware training** with QKeras while also imposing 50% compression → quantized model
  - for VAE this is done for the encoder only
  - scan range 2 to 16 integer bits and compare performance with respect to baseline pruned model in terms of AUC and TPR @ fixed FPR of $10^{-5}$
Ultra-fast autoencoders

**PLAIN CNN AUTOENCODER WITH MSE AD SCORE**

- Very stable performance versus bitwidth
- Ratios are typically above 1
  - quantized model is better than baseline
- Similar results for the DNN based architecture

**VARIATIONAL CNN AUTOENCODER WITH D_{KL} SCORE**

- Less stable performance versus bitwidth wrt plain AE
- Ratios are typically below 1
  - quantized model is worse than baseline
- Similar results for Rz strategy and DNN based architecture
Ultra-fast autoencoders

Xilinx VU9P @ 200 MHz (5 ns clock) → HL-LHC planned L1 trigger hardware!

<table>
<thead>
<tr>
<th>Model</th>
<th>DSP [%]</th>
<th>LUT [%]</th>
<th>FF [%]</th>
<th>BRAM [%]</th>
<th>Latency [ns]</th>
<th>II [ns]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN AE QAT 8 bits</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>0.5</td>
<td>130</td>
<td>5</td>
</tr>
<tr>
<td>CNN AE QAT 4 bits</td>
<td>8</td>
<td>47</td>
<td>5</td>
<td>6</td>
<td>1480</td>
<td>895</td>
</tr>
<tr>
<td>DNN VAE PTQ 8 bits</td>
<td>1</td>
<td>3</td>
<td>0.5</td>
<td>0.3</td>
<td>80</td>
<td>5</td>
</tr>
<tr>
<td>CNN VAE PTQ 8 bits</td>
<td>10</td>
<td>12</td>
<td>4</td>
<td>2</td>
<td>365</td>
<td>115</td>
</tr>
</tbody>
</table>

Xilinx Virtex7 690 @ 200 MHz (5 ns clock) → Current CMS L1 trigger hardware!

<table>
<thead>
<tr>
<th>Model</th>
<th>DSP [%]</th>
<th>LUT [%]</th>
<th>FF [%]</th>
<th>BRAM [%]</th>
<th>Latency [ns]</th>
<th>II [ns]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN VAE PTQ 8 bits</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>0.4</td>
<td>205</td>
<td>5</td>
</tr>
</tbody>
</table>

Brand new CMS results to be presented at the Fast ML Workshop!

Nature Machine Intelligence 4, 154 (2022)
Triggering on the unimagined

Process each collision every 25 ns with an AD algorithm

 Might get collected because triggered by a standard algo

IS IT ANOMALOUS?

NO

YES

Could be a hint of a so far unimagined process

Several ideas on what to do with these anomalous events:

• could aid standard signatures (eg, dijet) - supervised or unsupervised
• could drive new theory-motived searches with low or zero trigger acceptance
• upon anomaly interpretation, could seed a fully supervised search
• with the agreement of the collaborations, the output stream could be published as Open Data to foster new ideas!
Other approaches could exist for optimized **anomaly detection** in **low-latency and low-resource** experimental environments.

We have thought setup a new challenge to stimulate a community effort!

**CLICK ME!**

Find training/testing datasets and lot of info on how to estimate latency and footprint of your algorithm!

*Nature Scientific Data 9, 118 (2022)*
The High-Luminosity LHC challenge

instantaneous luminosity

x 0.75  x 1–2  x 2.5  x 5–7

LHC Run 1
$\sqrt{s} = 7$–8 TeV
30/fb

Long Shutdown 1

LHC Run 2
$\sqrt{s} = 13$ TeV
150/fb

Long Shutdown 2

LHC Run 3
$\sqrt{s} = 14$ TeV
300/fb

Long Shutdown 3

Run 4: HL-LHC
$\sqrt{s} = 14$ TeV
3000/fb

LHC TODAY

HL-LHC

NOW

2029

40 simultaneous collisions per bunch crossing

200 simultaneous collisions per bunch crossing + more granular detector!
Advantages for anomaly detection (but not only):
• use all fundamental event properties (as in Run 3)
• higher-resolution reconstruction of such properties
• particle-based raw information available

40 MHz tracking!

Muon detectors

Calorimeters

12 μs latency will allow us to deploy bigger models!
Beyond anomaly detection

• Many developments for HL-LHC ongoing in CMS — all supervised algorithms able to harness the enhanced trigger system capabilities

• Opportunity to port compressed versions of offline ML algos to the hardware trigger for more informed background rejection
  - b-tagging (DP-2022-021)
  - tau reconstruction (CMS-TDR-021)
  - long-lived particles tagging (arXiv.2307.05152)
  - vertex finding (DP-2022-020)
  - track quality
  - electron identification
  - ...

• Opportunity to think about new approaches (e.g., continual learning — DP-2023-022) to ensure performance robustness versus quickly changing detector conditions
  - addressing in particular possible generalization loss due to compression/quantization
Continual learning

- Continual learning can be used to continuously update re-configurable ML model weights in quasi real time
  - only a small dataset needed for model update given new conditions without forgetting essential information from previous baseline training

- Eg, ML model for vertex reconstruction makes use of tracks whose reconstruction can quickly degrade because of radiation damage

- The avalanche framework for CL is used where each degraded sample is a new experience to the model and a replay buffer strategy is used
  - keeps some data from each of the previous experiences in a buffer which is then mixed in with the current experience as the model trains
From fast to ultra fast ML

ASICs typically used at the front end for sensors read out: directly embed ML in here to allow intelligent data compression at the very edge.
Example:
High-granularity calorimeter @ HL-LHC

Novel technology for CMS endcap calorimeter:
50 layers with unprecedented number of readout channels (6M)!
Example: CMS HG calorimeter

**Input**

HGCAL 8" hex module

432 silicon sensors → 48 trigger cells (TC) @ 7b per TC

(336b in total)

**Output**

“Super trigger cell” algo

3 [16 TC sum] x 16–48 bits

= 48—144 bits

(depending on the position)
Example: CMS HG calorimeter

Input
HGCAL 8" hex module
432 silicon sensors → 48 trigger cells (TC) @ 7b per TC
(336b in total)

Output:
“Super trigger cell” algo
3 [16 TC sum] x 16–48 bits
= 48—144 bits
(depending on the position)

Can we do a better job of encoding the info in those bits w/o so much loss in granularity?

Encoder on ASIC
Decoder on L1 board

Really need quantized training here to optimize information encoding

Use QKeras!
Example: CMS HG calorimeter

- Evaluate AutoEncoder performance according to image similarity

- **Energy Mover’s distance**: quantify the cost of transforming one image into another as energy x distance (lower EMD better performance)

- Use of more outputs at lower precision outperform their counterpart

- **Use hls4ml for mapping the ML model onto reconfigurable logic:**
  - extended for the ML-to-ASIC flow to support Mentor’s Catapult HLS and target the specific 65 nm LP CMOS technology

- Downstream performance driven by physics to be fully assessed with codesign tools allowing for fast feedback loop!

![EMD plot]

<table>
<thead>
<tr>
<th>TABLE III</th>
<th>KEY SIMULATION PERFORMANCE PARAMETERS OF THE DESIGN.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td>Energy/inference</td>
</tr>
<tr>
<td>50 ns</td>
<td>2.38 nJ/inf.</td>
</tr>
</tbody>
</table>
Summary & Conclusions

• In this course I have introduced the need in particle physics to bring ML closer to the data source at the edge of particle physics detectors

• We have discussed a few real-life applications at the energy and intensity frontiers where system constraints require innovation in developing new ML training paradigms, model architectures as well as tools for easy deployment

• The hls4ml tool can help physicists to bring ML to such systems in order to meet system constraints while giving you control on the optimization
  
  - tutorial: https://github.com/fastmachinelearning/hls4ml-tutorial
  
  - documentation: https://fastmachinelearning.org/hls4ml/

• There are many more applications of fast ML in science — check out the next workshop held at Imperial College this September to learn more!