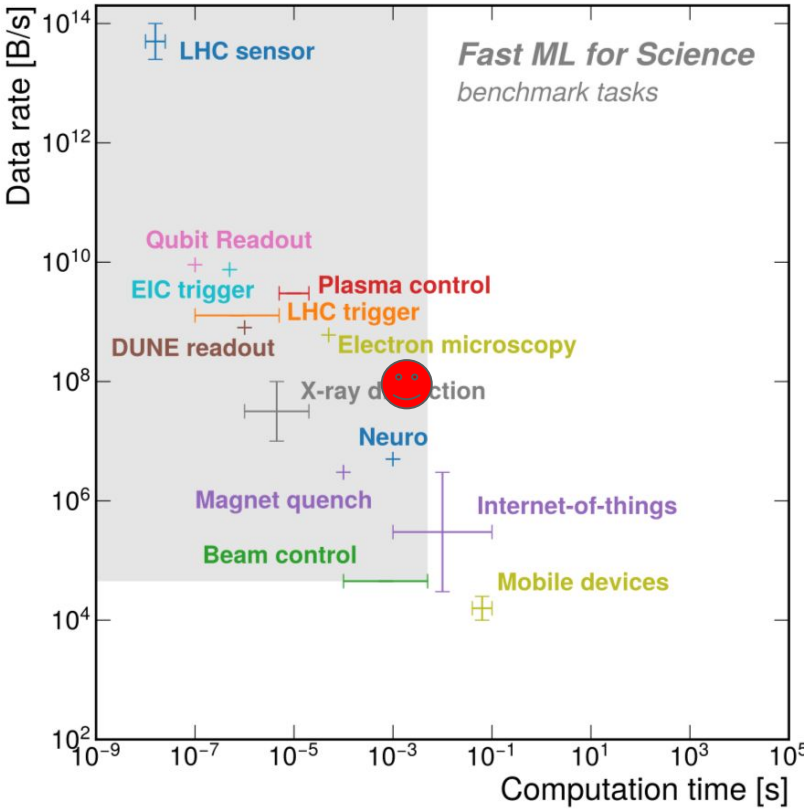
Modern Machine Learning Model Deployment on FPGA for KamLAND-Zen

Zepeng Li
University of Hawaii at Manoa

ML4FE workshop
May 19 2025

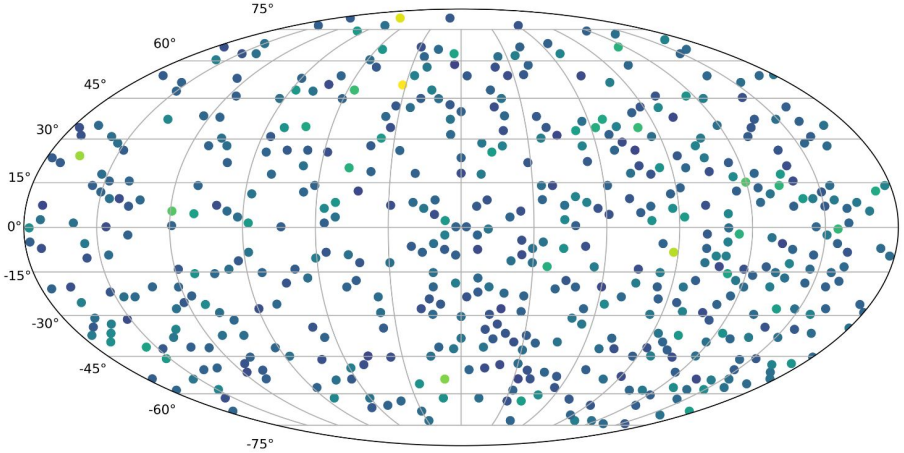
FastML for rare events search experiments

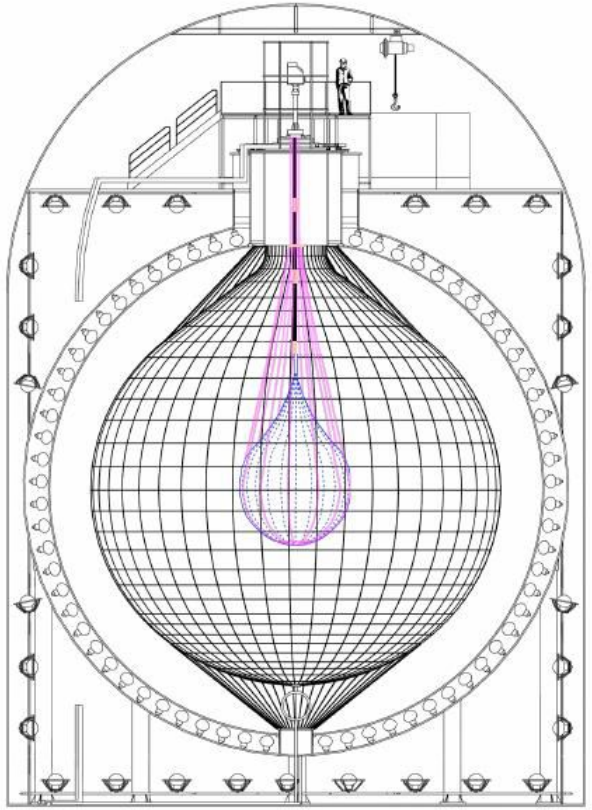


- Rare event search experiments: dark matter, neutrino, and neutrinoless double beta decay.
- Experiments built with low background materials and underground.
- FastML could help to extract rare events and suppress background!
- Less stringent latency requirement and possibility for complicated network.

KamLAND-Zen detector

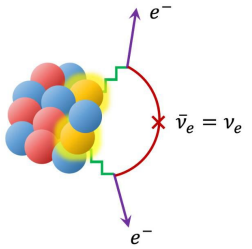
- Particles interact in the liquid scintillator and deposit energy. Energy is converted into light and detected by photo-multipliers.
- Energy resolution: 6.7%/sqrt(E (MeV))
- Vertex resolution: ~13.7 cm

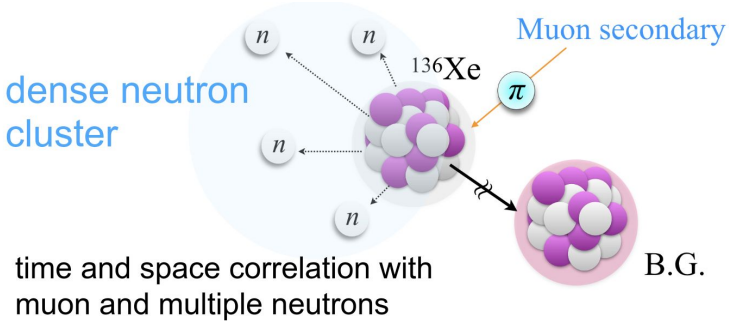


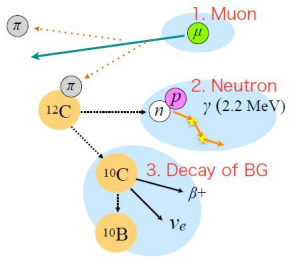


Machine learning based background rejection in KamLAND-Zen

- Signal is 2.46 MeV electron events
- Primary backgrounds:
  - 2vbb decays
  - Long-lived cosmic muon spallation
- Minor backgrounds
  - Radioactive background
  - Solar neutrinos
  - Short-lived cosmic muon spallation

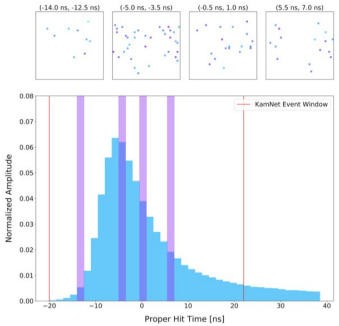


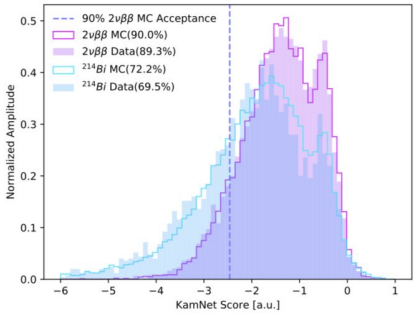


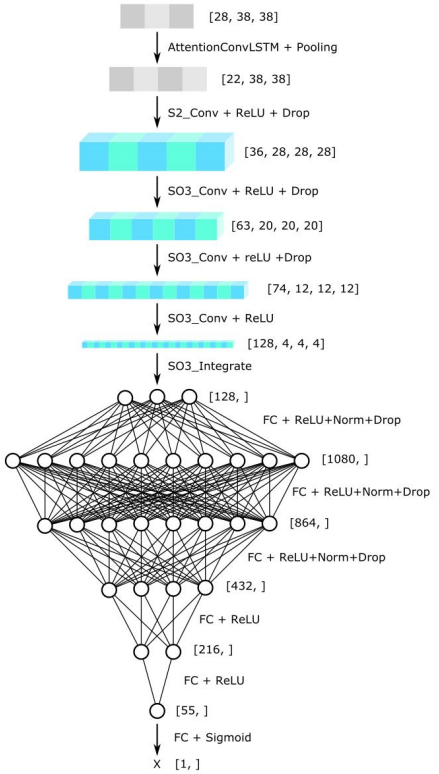


Machine learning based background rejection in KamLAND-Zen

- A novel deep learning model to distinguish backgrounds and signals
- KamNet takes a time-series of 2-D hit maps and returns a single-valued KamNetScore
- Convolutional-LSTM (Long-Short Term Memory) Layer with attention module
  - Learns to identify and focus in on important sections of the event
- Spherical Convolution
  - Utilizes spherical symmetry to learn complex features

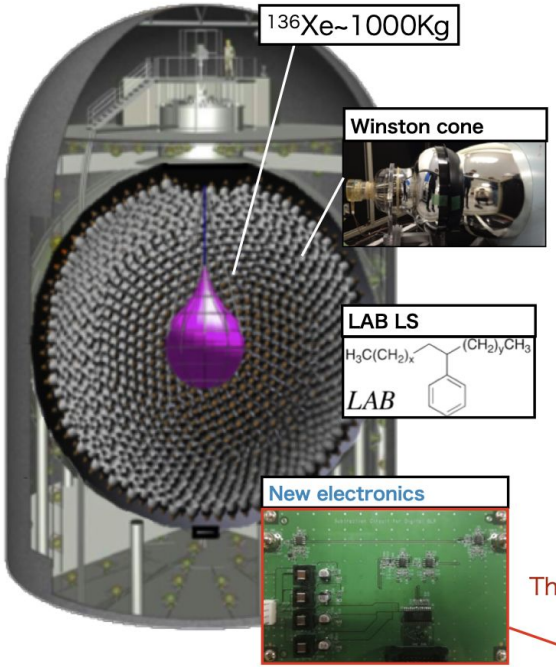




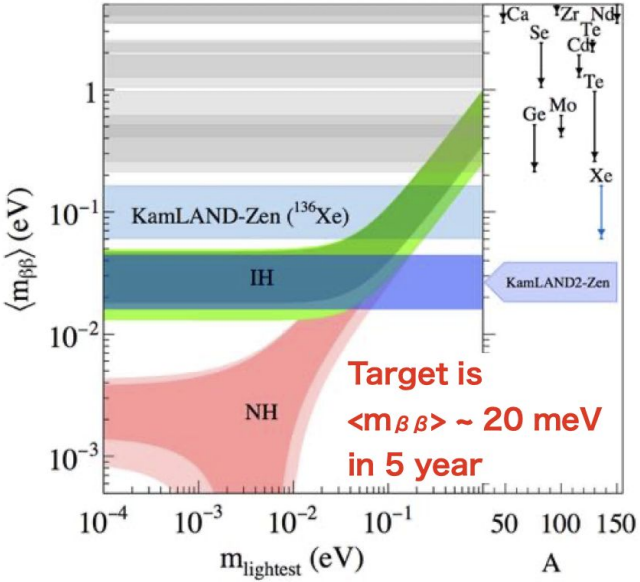


KamLAND2-Zen

From H. Ozaki



This talk



We aim to increase light collection by more than 5 times!

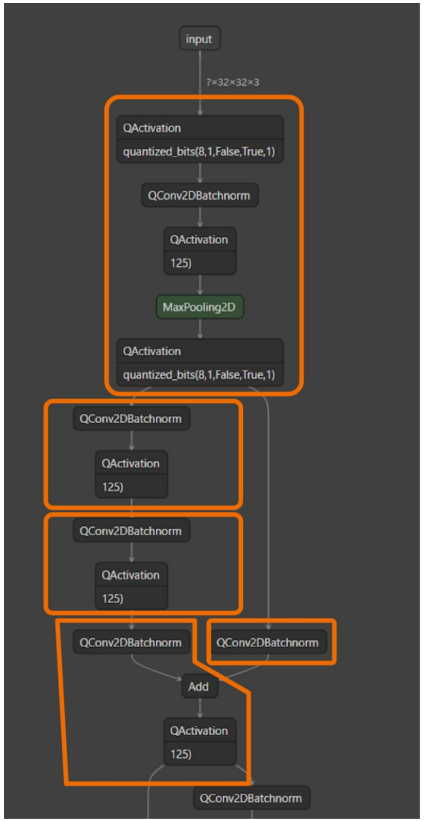
σ (2.6MeV) = 4% → 2%

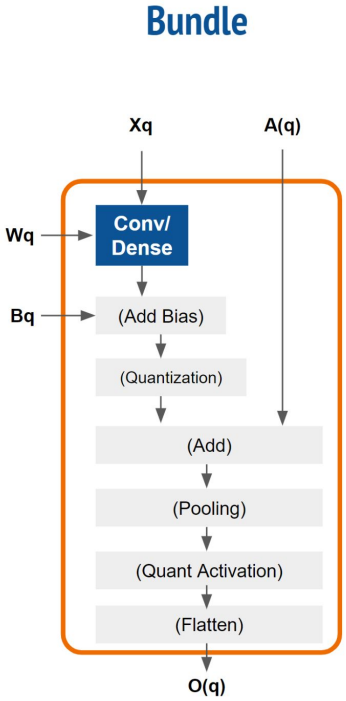
We could explore modern ML algorithms deployed on advanced FPGA in the front-end for real-time trigger/data processing.

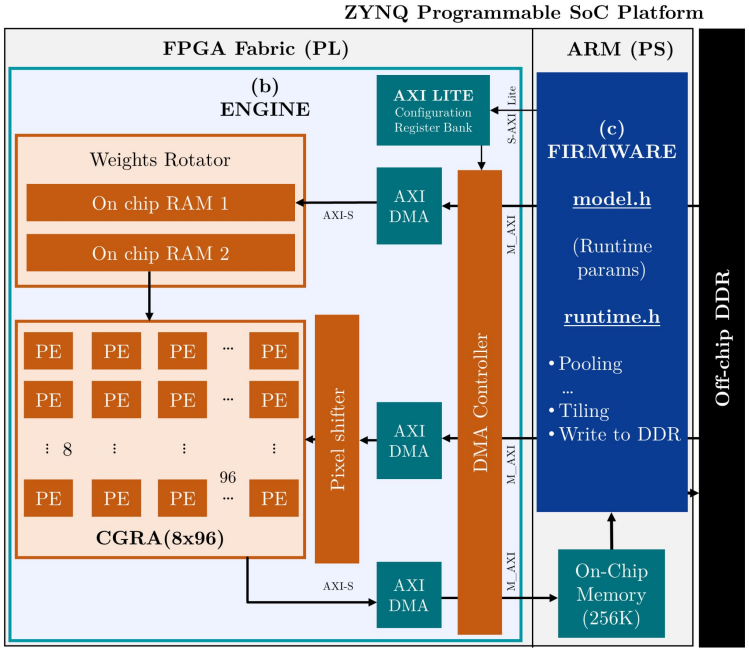
Other options(Scintillation film, Imaging detector, pressurized xenon ..) in development

We are developing new electronics with wide dynamic range!

Modern ML models on FPGA using CGRA4ML







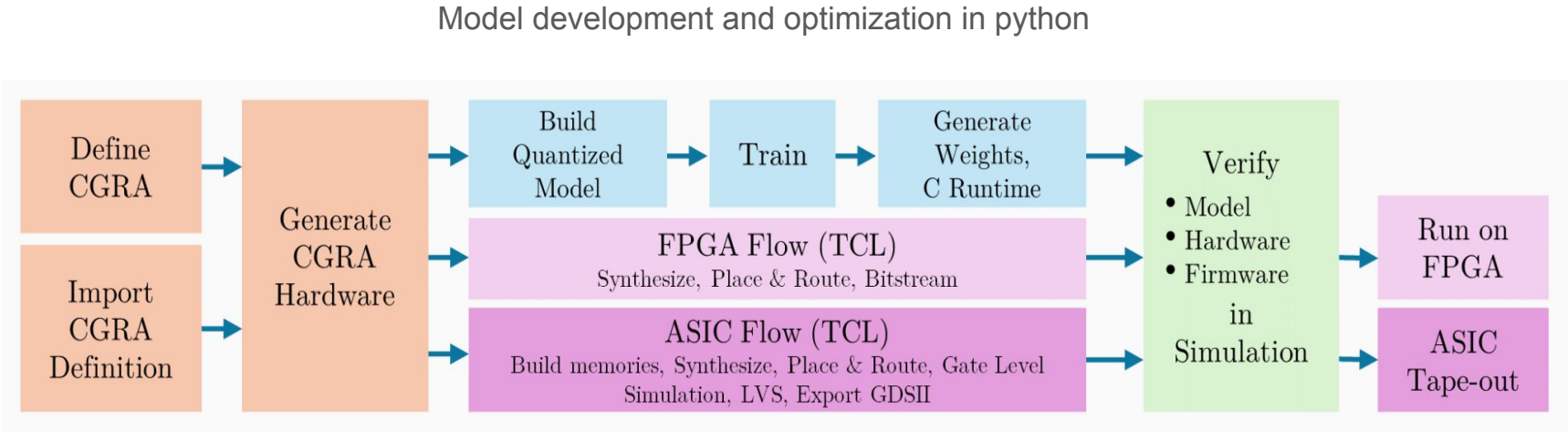
Parameterizable Coarse-Grained Reconfigurable Array

PE: processing element

Off-chip memory for parameters storage

https://arxiv.org/pdf/2408.15561

Modern ML models on FPGA using CGRA4ML



- Workflow is similar to HLS4ML: network development and optimization in python using Tensorflow/QKeras.
- CGRA is reconfigurable for different tasks and FPGAs.
- Xilinx softwares for synthesis and verification.

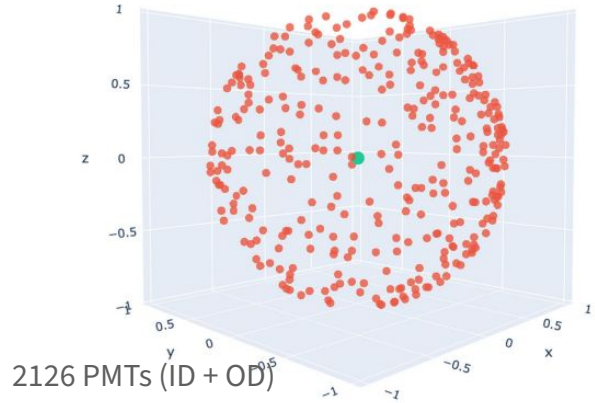
Results

Model	ResNet-50	PointNet
Bits	4	4
PEs	(7,96)	(32,32)
Frequency (MHz)	250	250
FFs	101706	69277
LUTs	82200	100076
BRAMs	6	4.5
Static Power (W)	0.700	0.700
Dynamic Power (W)	3.847	3.840
Total Power (W)	4.547	4.540
GOPs/W	37.3	56.8

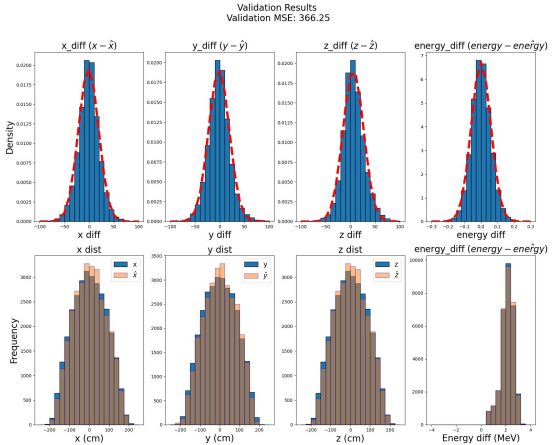
TABLE III
IMPLEMENTATION OF RESNET-50 AND POINTNET ON ZCU104 FPGA

PointNET event reconstruction

- Data can be thought of as a point cloud
  - x, y, z, t, and q of each PMT
  - Geometric semantics
  - Invariant to permutations (x, y, z encoded)
- Use the PointNET architecture (Qi et al 2017)



2126 PMTs (ID + OD)



Model Quantization in cgra4ml

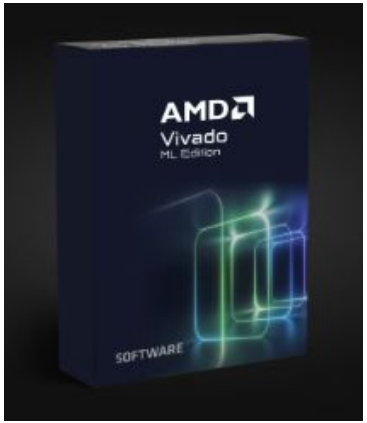
- Need to compress model
- Need to understand optimal hardware parameters
- Training QKeras establishes baseline for comparing hardware and software compression

```
class UserModel(XModel):
    def __init__(self, sys_bits, x_int_bits, *args, **kwargs):
        super(...)  x_int_bits, *args, **kwargs)
                    (variable) b0: Any
        self.b0 = XBundle(
            # core=XDense(
            #     k_int_bits=0,
            #     b_int_bits=0,
            #     units=64,
            #     act=XActivation(sys_bits=sys_bits, o_int_bits=0, type='relu', slope=0)
            # )
            core=XConvBN(
                k_int_bits=0,
                b_int_bits=0,
                filters=64,
                kernel_size=1,
                act=XActivation(sys_bits=sys_bits, o_int_bits=0, type='relu', slope=0)
            ),
        )
```

cgra port of PointNET

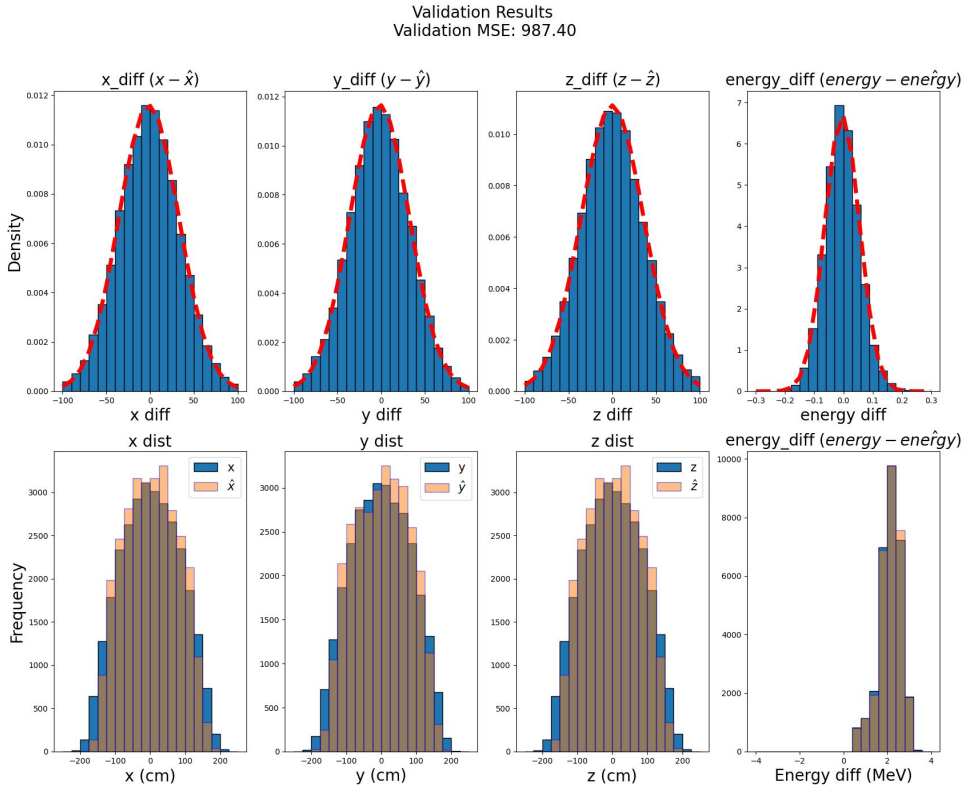
Software to Hardware: Key Tools

- cgra4ml
  - Converts Tensorflow model for vivado synthesis
  - Open-source and available on Github
- Vivado (AMD)
  - Model export verification and simulation
  - Synthesizes FPGA representation
- Vitis (AMD)
  - C-wrapper for FPGA model verification



Images courtesy of AMD

PointNET cgra Port Accuracy Results



Reconstruction Results Summarized

Experiment	X Error (cm)	Y Error (cm)	Z Error (cm)	E Error (MeV)
Traditional KLZ	17	17	17	N/A
QKeras	20	21	21	0.06
cgra4ml	34	34	36	0.06

- The pointnet on FPGA achieves vertex reconstruction accuracy slightly worse than offline reconstruction.
- Accuracy is good enough for a position-aware trigger in KamLAND-Zen.

RFSoC4x2

- ZYNQ Ultrascale+ FPGA in lab
- AMD Kit

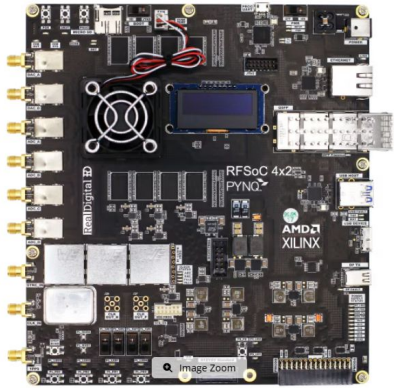
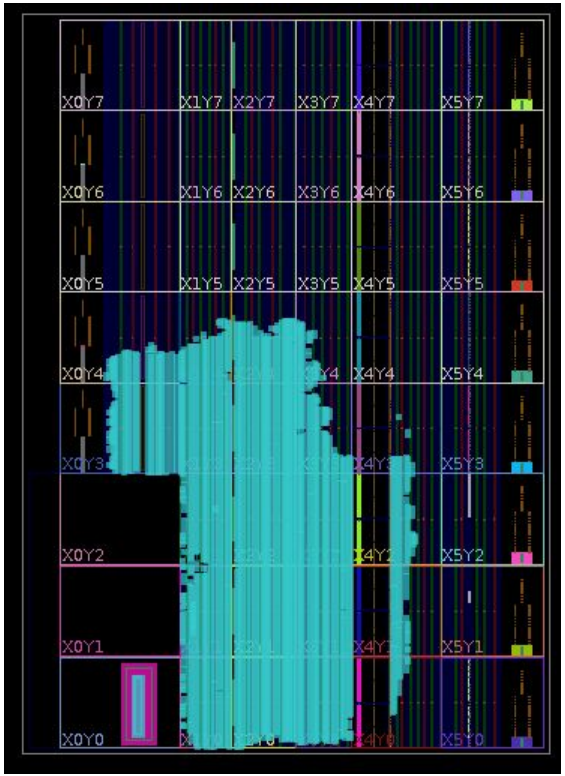


Image courtesy of AMD

https://ml4physicalsciences.github.io/2024/files/NeurIPS_ML4PS_2024_153.pdf



Vivado synthesis of model

Version	Latency (ms/batch) @ 20 runs
Trained	6980.9
Untrained	6996

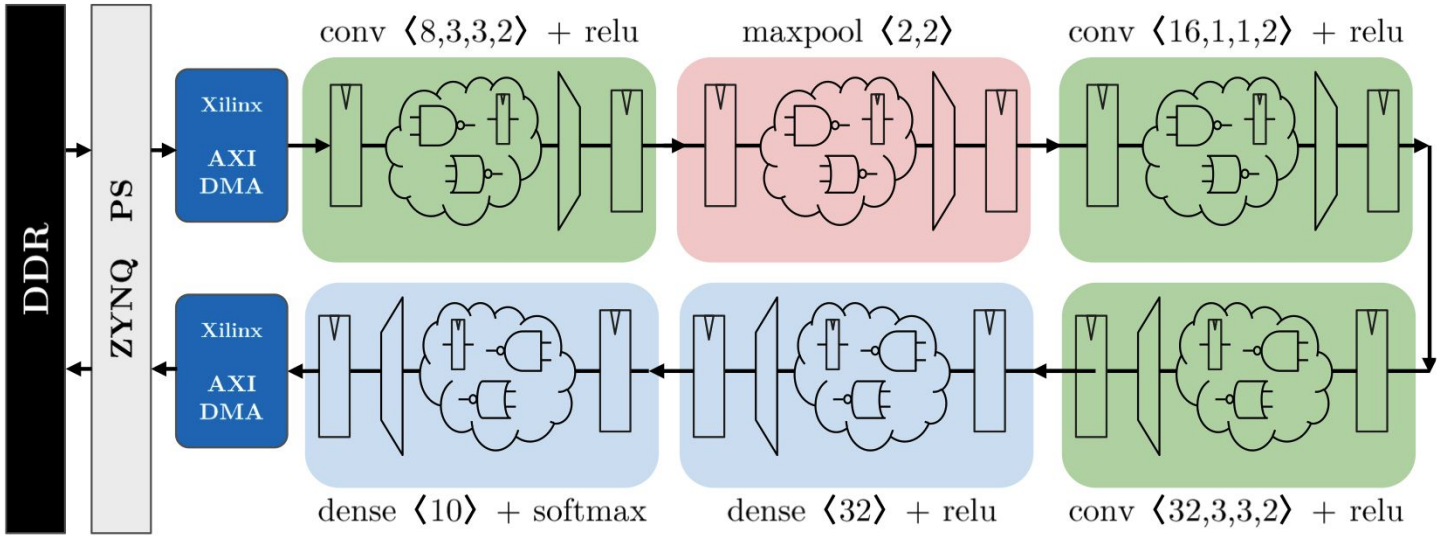
~436.3 ms
inference per event

- A simpler model (<<1 million parameters) without losing much accuracy.
- Use PMT cluster as a single point instead of a single PMT as input.
- Optimize quantization

Summary

- CGRA4ML provide a framework for modern ML model deployment on FPGA.
- PointNET is an effective way of reconstructing detector physics in KLZ.
- We can deploy PointNET onto an FPGA to make single-event inference that opens possibility of position-ware trigger.

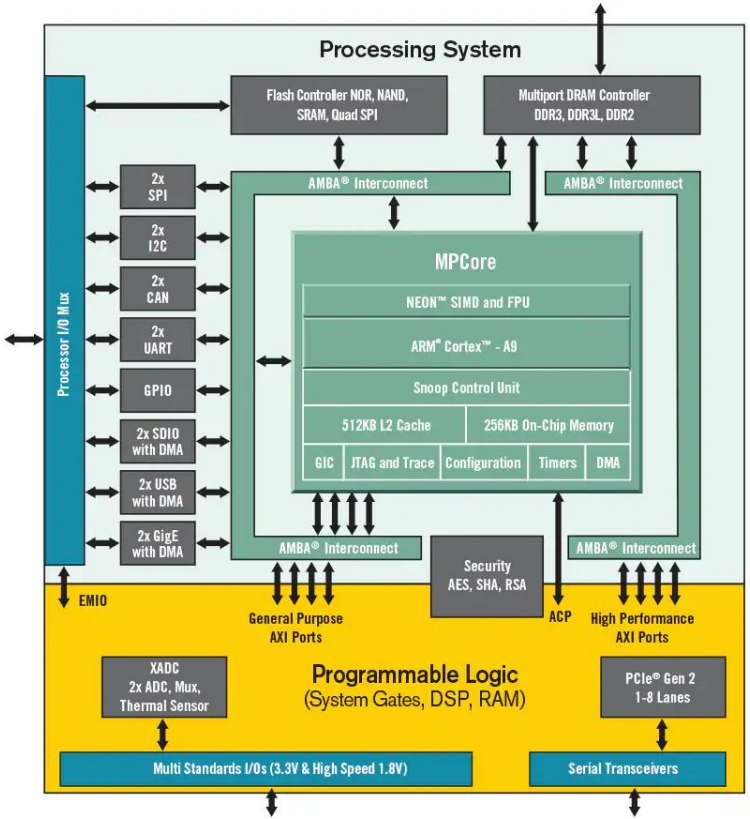
HLS4ML



Layer-by-layer implementation of neural networks in HLS4ML.

Multiplier could be reused inside a layer.

ZYNQ heterogeneous SoC



- System on Chip: Arm CPU + FPGA
- Advanced eXtensible Interface (AXI) provides for high bandwidth and low latency connections between elements.