

Next Steps for BIB Mitigation with AI: Ideas from Chicago



Daniel Abadjiev, Karri DiPetrillo,
Zoe Fulton, Eliza Howard, Ryan
Roberts, Megan Wells



Overview

- Wary of over-optimizing for physics performance when detector layout and import details like readout hardware are fluid.
- Goals are to explore what is possible in the range of feasible technology scenarios.
 - Can we anticipate the major challenges to performing on-detector BIB rejection?
 - Can we define the envelope of what's possible to achieve?
 - Can we leverage this to inform the work of the broader MuColl detector group?
- Also plan to keep some level of activity in lab testing of Smartpixels prototype
 - not directly connected to MuColl, but always nice to work with real devices.

SEU Studies

- Continued studies of dependence on architecture and quantization
- Dependence on the metrics and loss functions
- Also want to study separate models trained for each module – each module needs to hit a target data rate in the expected radiation environment at its position using the same hardware
- Training time mitigation with regularization

Investigate Clusters from MAIA Reconstruction

- Would remove several simplifying assumptions made in the current dataset
 - Would include realistic threshold(s)
 - Would include overlapping clusters, clusters on the edge of modules.
- But we would lose the detailed simulation of cluster formation. Requires validation first! And we may find that we need a hybrid approach.
- Could think about clustering algorithms and ROI finding (Do algorithms currently used for SmartPixels make sense for MuC?)
- Could imagine exploring more diverse signals
 - Signal overlapping with BIB (should be common)
 - Signal overlapping with signal (i.e. jets)
 - Signal from displaced decays

Considering different technologies

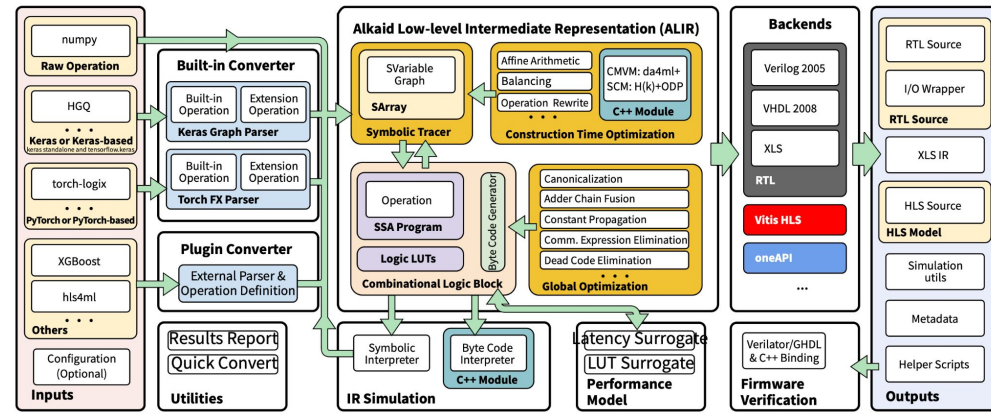
- ML techniques
 - New architectures?
 - HGQ quantization?
 - Regularization?
 - Training per module? (Or is it better to train 1 model with module ID as an input?)
- Detector
 - Impact of using 3D sensors
 - Used in innermost layers of Phase II detectors for rad hardness
 - Different drift patterns and sensor dimensions may affect cluster development
 - May require generating new E field maps for pixelav

Future Smartpixels at a Muon Collider

- This is probably redundant oops
- There are several directions to make a proof-of-concept less conceptual, many depend on exact tracker technology, layout, and design
 - Radiation robustness - reconfigurability, SEU study, smartpixels in a test beam
 - MAIA detector integration - MAIA digitization, BIB overlay, multiple tracker layers (with whichever layout is chosen), integrate a parameterized smartpixels cut in MAIA software
 - Compare datasets - pixelav smartpixels dataset vs. MAIA software generated dataset vs Angira's dataset, explore different digitization schemes
 - Physics performance - explore different signals, clusters from highly collimated jets, LLPs
 - Integrate algorithm with ROI finder, address question of overlapping clusters
 - Explore sensor technologies - 3d sensors, MAPS sensor
- This presentation: alternative ML architectures with the same framework, same dataset

HGQ + Alkaid

- HGQ can reduce the hardware resource usage of large models to help them fit on chip,
 - Reconfigurability with optimally quantized weights should be explored
- HGQ doesn't consider FFs since they are not easily predictable, but they will impact ASIC resource usage, optimizes EBOPs which measures LUT usage
- [Alkaid](#) replaces hls4ml workflow, converts numpy frontend to HDL code
 - Has a surrogate model to predict FPGA usage and use that for training optimization
 - Many optimizations: da4ml, H(K)+ODP, affine arithmetic, ternary adder fusion, LUT packing
 - “Theoretically supports ASIC” but it sounded like that hasn't been tested
- Can try out HGQ+Alkaid and see how well it works

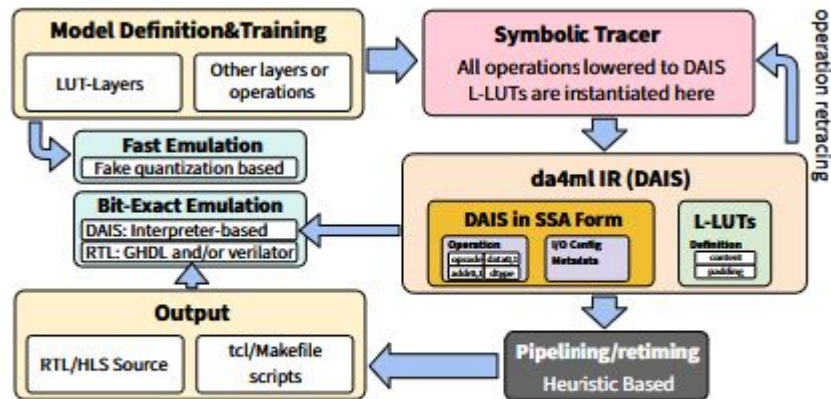


HGQ-LUT

- Extension to HGQ that directly targets LUT primitives during training
 - Instead of MLP activation, think of that layer as a “LUT layer”
- Allegedly faster than previous LUT-aware training
- Works with Alkaid framework for hardware RTL conversion
- Has been used to fit transformers on FPGA for jet tagging
- Zhiqiang Que reached out in smartpixels channel offering to help us get this running

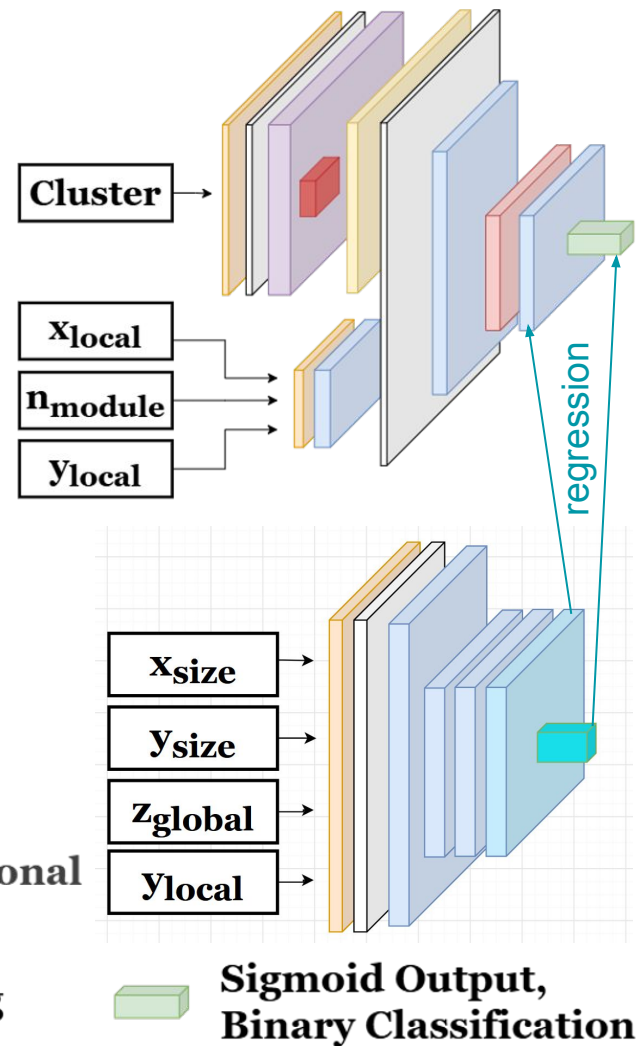
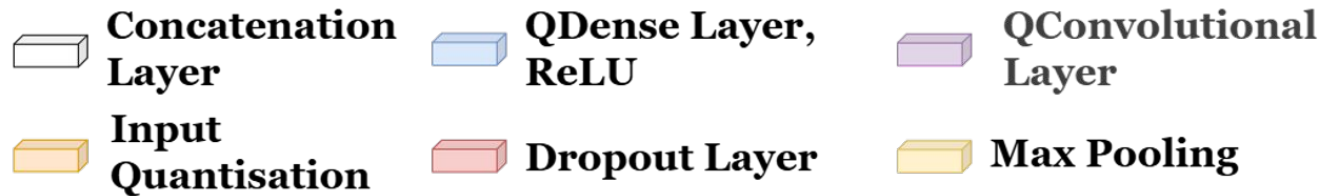
$$a_i^{(l)} = \sum_{j=1}^N \text{L-LUT}_{i,j}(a_j^{(l-1)})$$

$$a_i^{(l)} = \phi^{(l)} \left(\sum_{j=1}^N w_{ij}^{(l)} a_j^{(l-1)} + b_i^{(l)} \right)$$



Knowledge Distillation

- A knowledgeable “teacher” model (CNN, transformer) learns to represent data in low-dimensional output layers, which are used for classification
- A student regresses to learn these output layer representations rather than directly training to do classification
- Knowledge is “distilled” from the teacher to the student
 - Often, the student will learn better by regressing to the teacher’s output than by just training to do binary classification
- Could distill model 3’s classification to a model 1

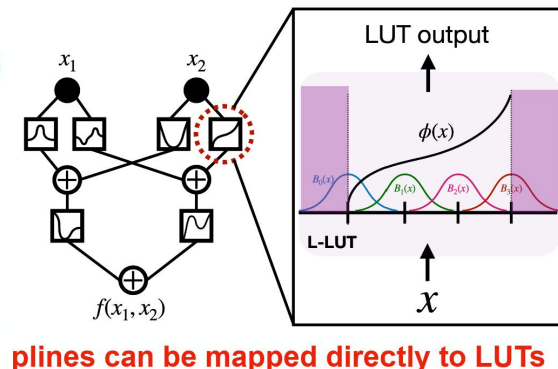
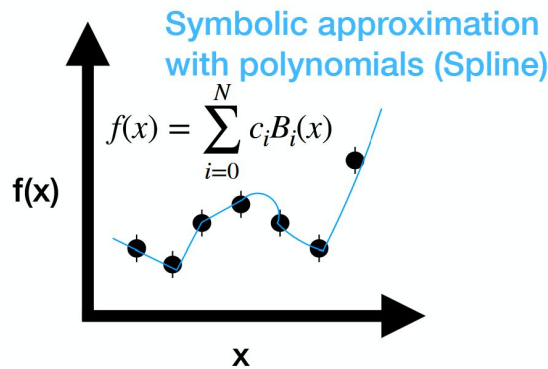
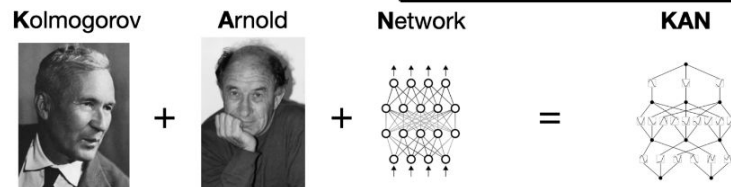
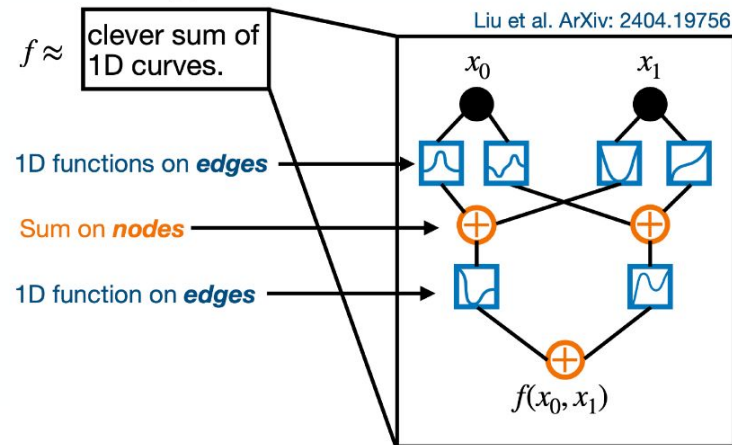


LNN

- Lots of work from Princeton on applying to Logic gate Neural Networks to smartpixels problem
- Can we apply them to the muon collider smartpixels dataset?
- Have LNNs been explored in context of knowledge distillation?

Kolmogorov Arnold Networks

- KANs use sum of splines instead of traditional weight + bias + ReLU
- Since the splines can be mapped directly to LUTs, large improvements to hardware efficiency by using KANs
- We could try replacing one of our MLPs with a KAN???
- Will the benefit from mapping to LUTs be as beneficial for ASIC as it is for FPGA??
- Is this resource benefit any better than an LNN?



Spiking Neural Networks

- Ehh I'm not convinced

Efficient RNN/Transformer implemented on Hardware

- Probably still too big to fit on ASIC