Fast AI/ML in Hardware for Particle Physics

Julia Gonski

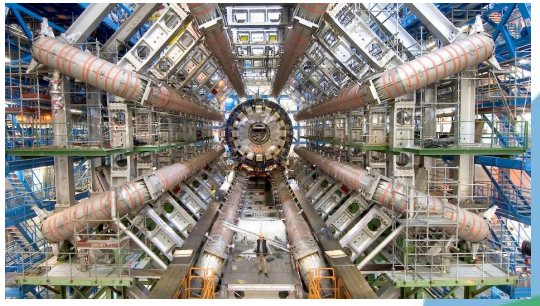
22 August 2024
HEPCAT @ SLAC

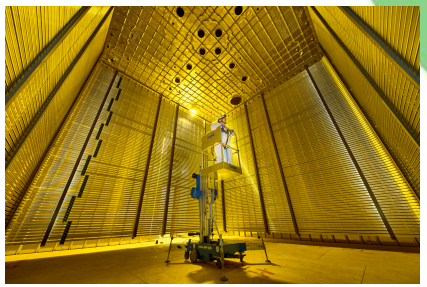
Outline

SLAC

- Motivation for ML in HEP electronics
- Fast ML Principles & Tools
  - Computing hardware
  - Quantization
  - Pruning
- Examples
  - Anomaly triggers on FPGAs
  - Front-end ML on ASICs

The Frontiers of High Energy Physics



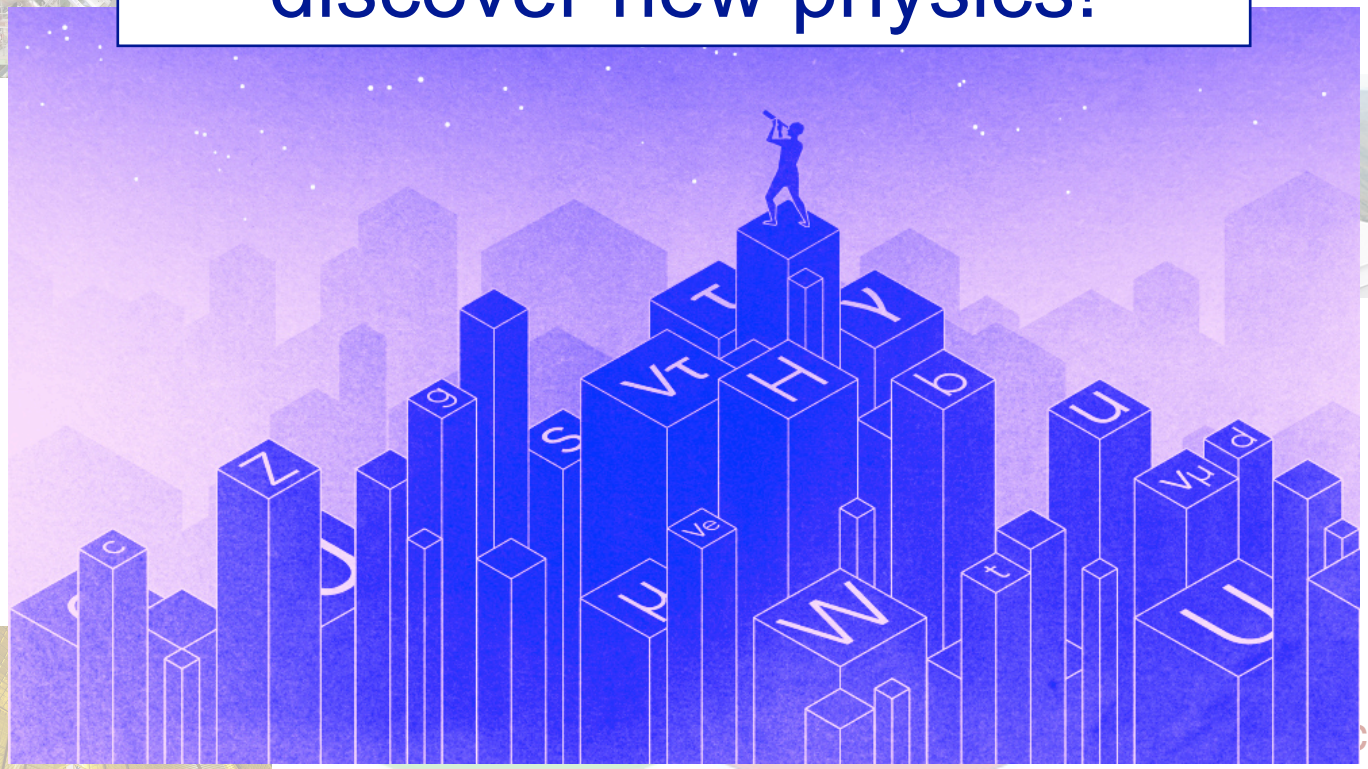


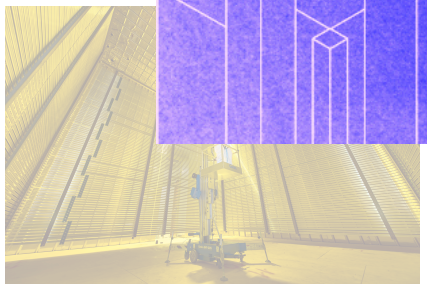
+ theory & computing, advanced technology, & accelerators

The Frontiers of High Energy Physics

SLAC

Maximize chances to discover new physics!

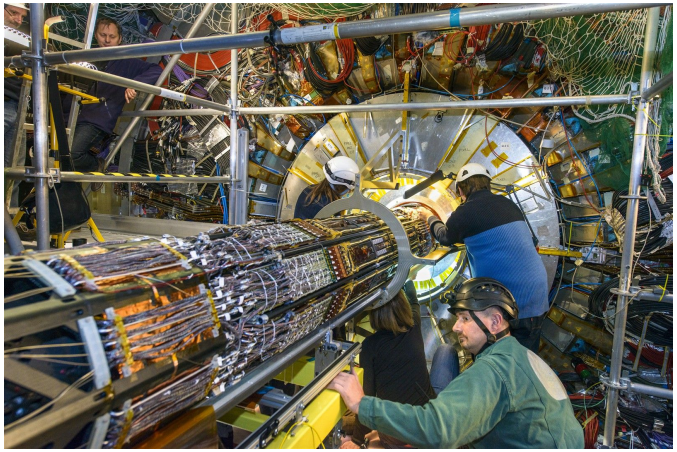


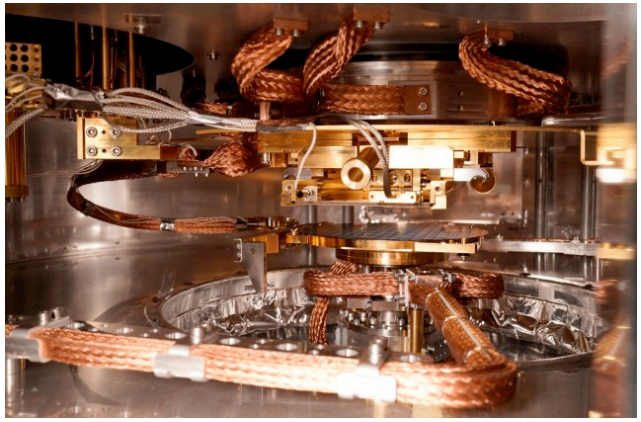


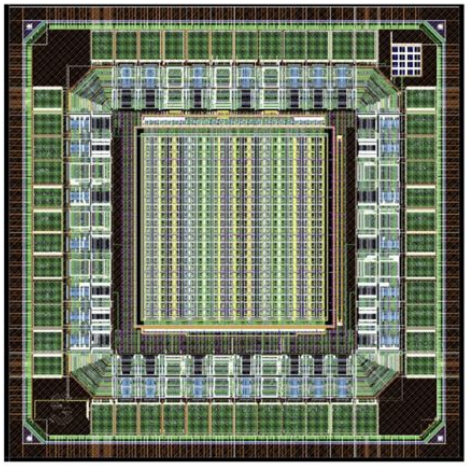
computing,
advanced technology,
& accelerators

HEP Environments

- Environments of high energy physics experiments are **“extreme”**
  - Very high radiation doses
  - Extreme temperatures (cryogenic)
  - Very high data rates/density
  - Spatial constraints (no room for cooling)
  - Very low latencies
- Many COTS hardware options don’t work for us; requires **custom design**



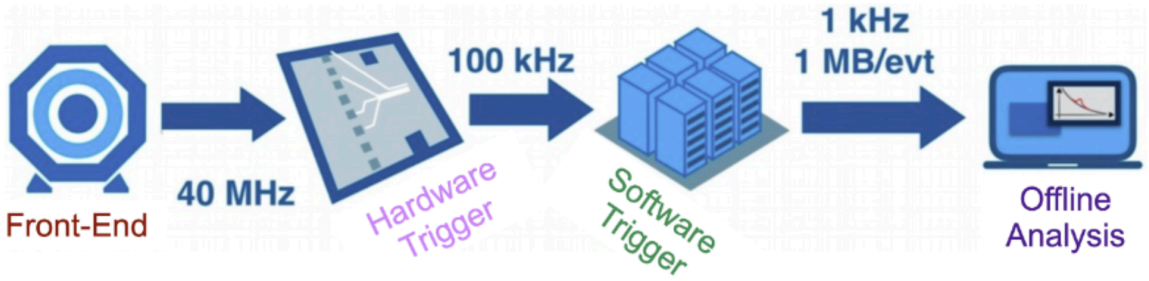




Layout of SLAC prototype for WP1.2 2022 shared submission on TowerSemi 65nm

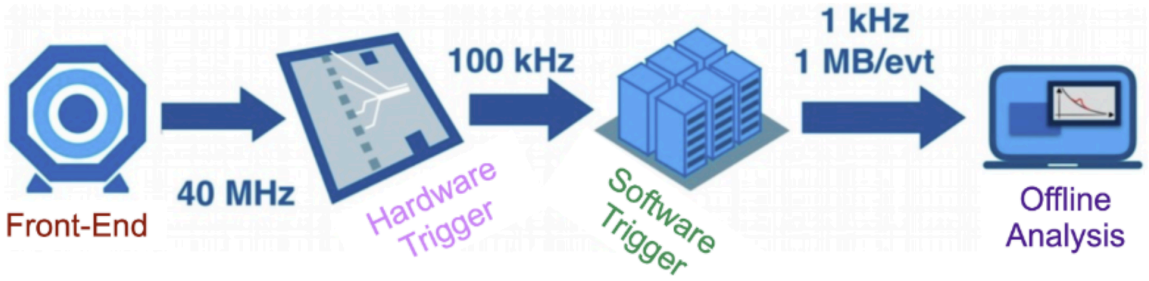
Data Acquisition in HEP

- Heterogenous data pipelines process data from acquisition at-source to offline analysis
  - At colliders: **trigger** filters events to reduce very high data rates to manageable levels

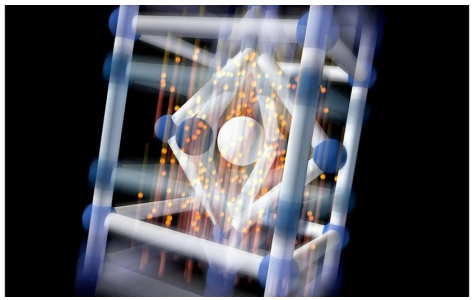


Data Acquisition in HEP

- Heterogenous data pipelines process data from acquisition at-source to offline analysis
  - At colliders: **trigger** filters events to reduce very high data rates to manageable levels



  - Not viable depending on the type of detector, e.g. processing unique images from 1 MHz X-ray pulses via LINAC Coherent Light Source (LCLS) @ SLAC



Data Acquisition in HEP

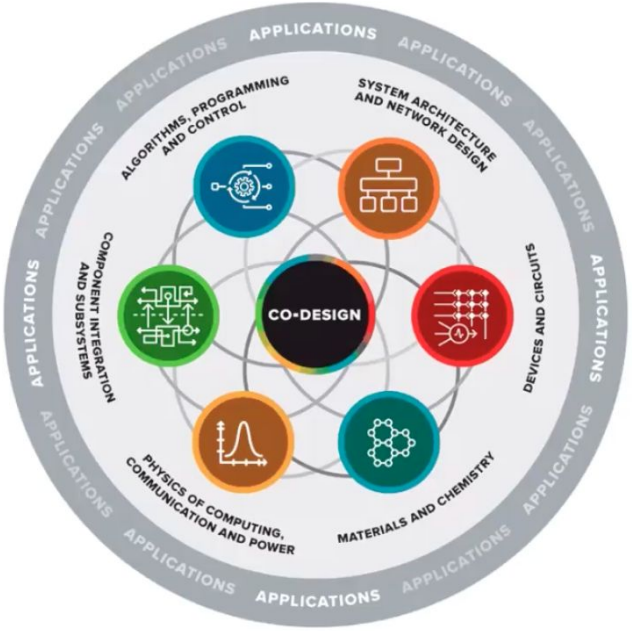
- Heterogenous data pipelines process data from acquisition at-source to offline analysis
  - At colliders: **trigger** filters events to reduce very high data rates to manageable levels
  - Not viable c… mages from 1 MHz X-ray …

Tough inference problems (classification, feature extraction, data compression, in real-time) motivate ML in DAQ electronics

Fast ML

- **"Fast ML"** = hardware acceleration of ML algorithms running in software to compile and operate in hardware platforms
  - Lower power, smaller footprint, faster inference time
  - Allows for advanced ML algorithms to run within HEP experimental data acquisition/ triggering scheme

- **Codesign**: intrinsic development loop between algorithm and hardware design



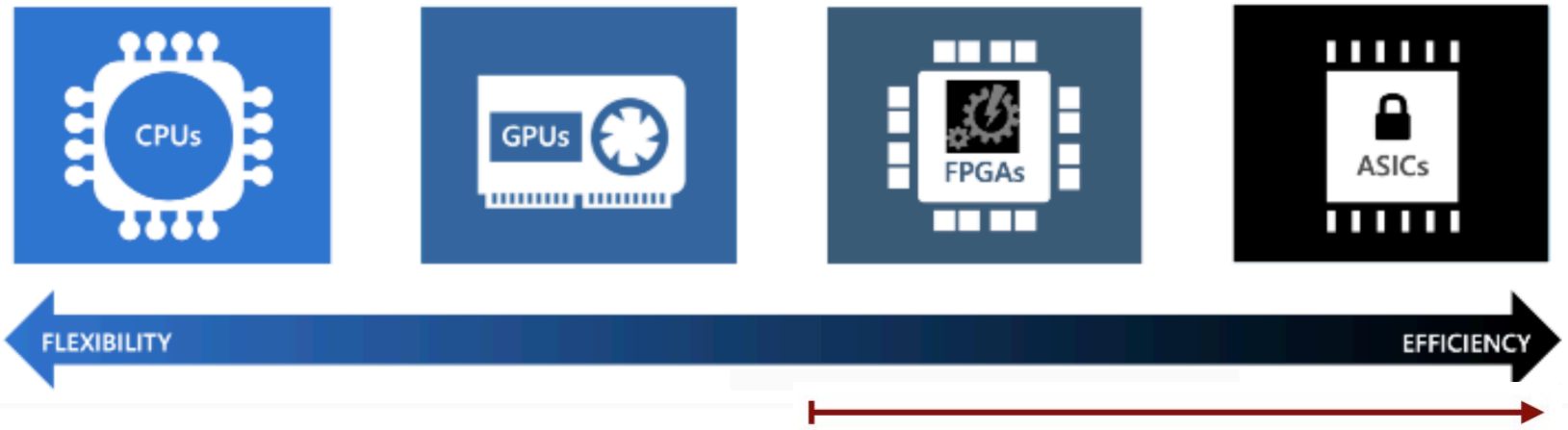
Fast ML Principles & Tools

Computing Hardware

- What computing to use?

Computing Hardware

- What computing to use?
  - GPUs can only get you down to ~O(ms) latencies…
  - But we need much faster!
    - LHC trigger level 1 ~ O(μs)
    - LHC front end ~ O(ns)



Guidelines:
> 100 Gbps throughput
< 1ms computational latency
< 10W power budget

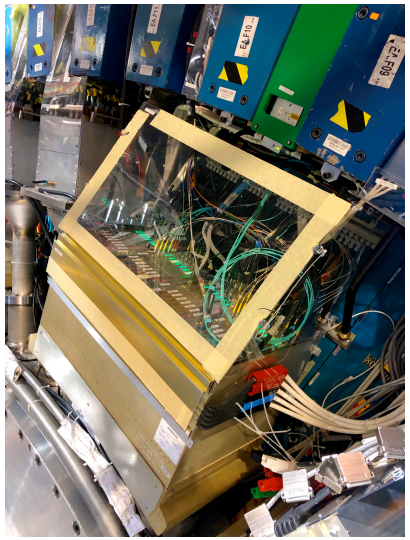
ASICs

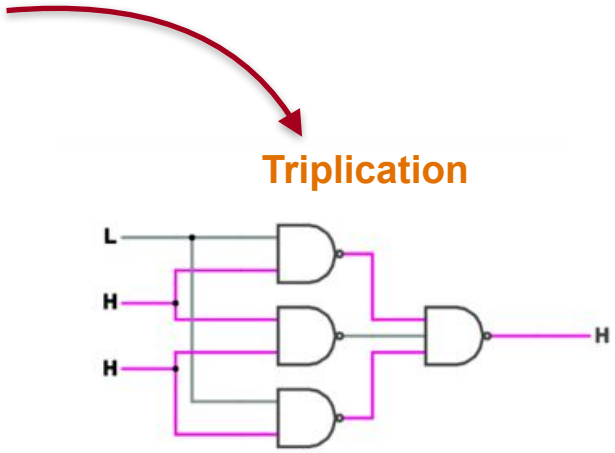
- "Application specific integrated circuit": set of electronic logical components etched into silicon
- Optimized to do one specific calculation (but lowest power & highest speed)
  - Ex. in HEP: amplifier, shaper, analog-digital conversion at the front-end
- Total control over design (no resources pre-allocated)
  - To run ML in an ASIC, you need to **design the chip for a specific model** (with some configuration)
  - Can be made *radiation-hard* (eg. through logical triplication of bits)

ASIC Design starts with a blank tableau

You need to add the components you need/want

ATLAS LAr Calorimeter

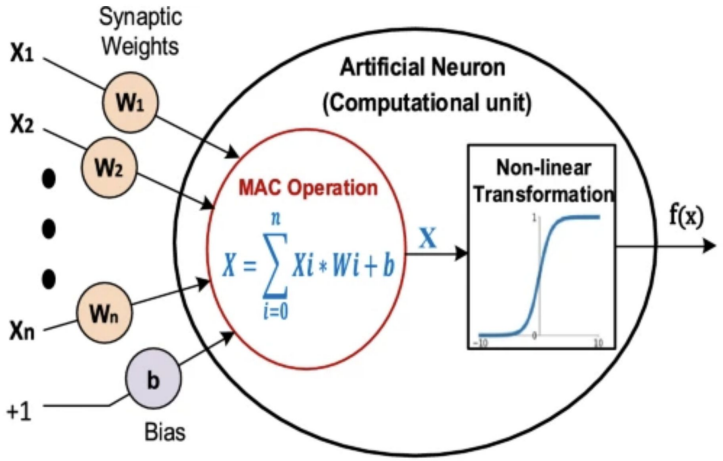


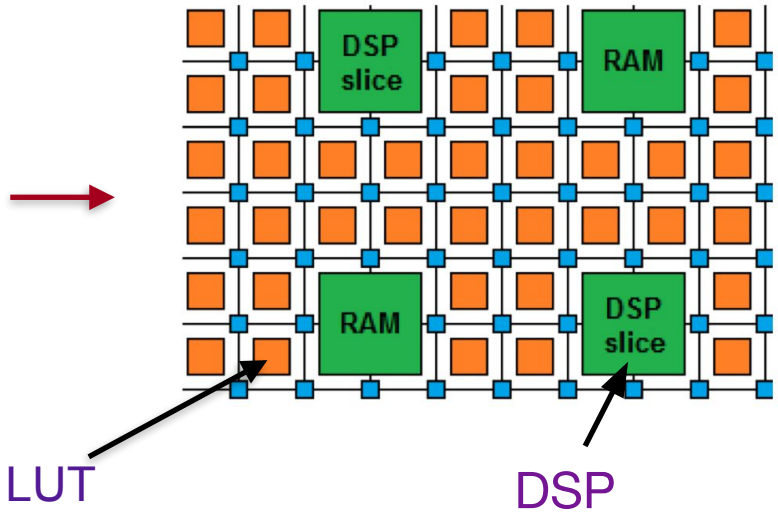


FPGAs

- "Field programmable gate array": reprogrammable integrated circuit
- Contain many different building blocks ('resources') which are connected together to efficiently perform calculation
  - **Digital signal processors (DSPs)**: fast/efficient multiply-and-accumulate operations; scarcest resource for ML evaluation
  - **Look-up tables (LUTs)**: perform arbitrary functions on small bit-width inputs (2-6)
- Resource allocation cannot be dynamically remapped while running

Neural Net MAC

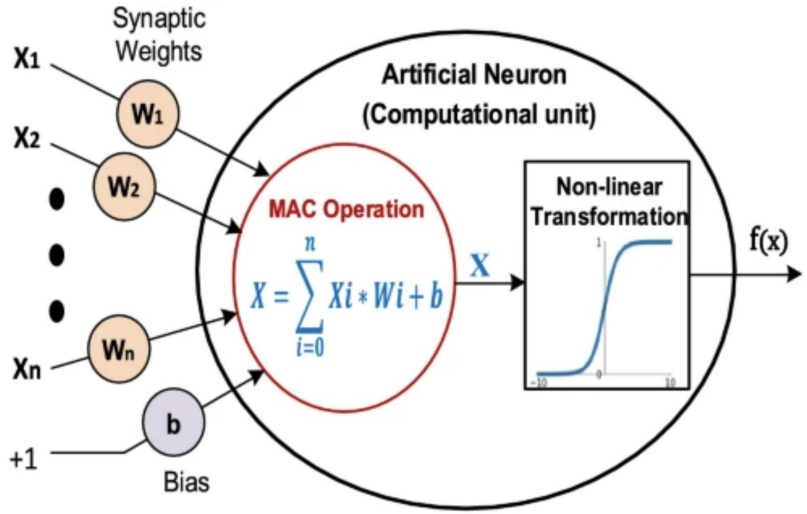




Running ML on Hardware

SLAC

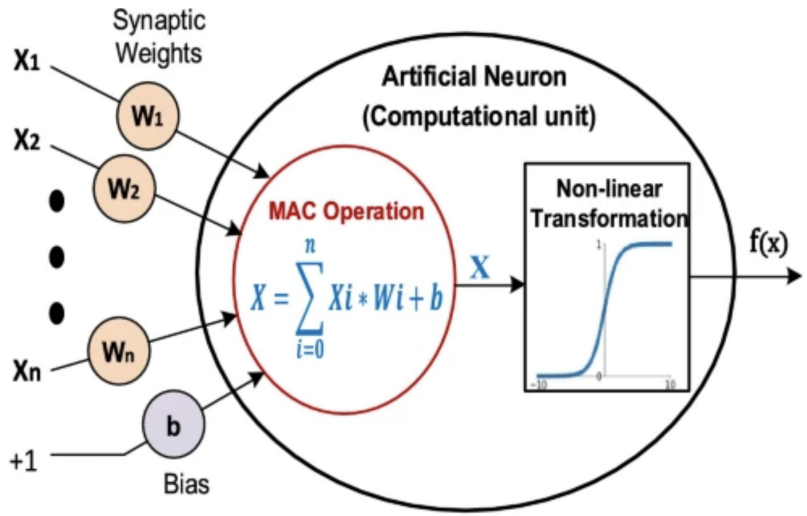
Calculation

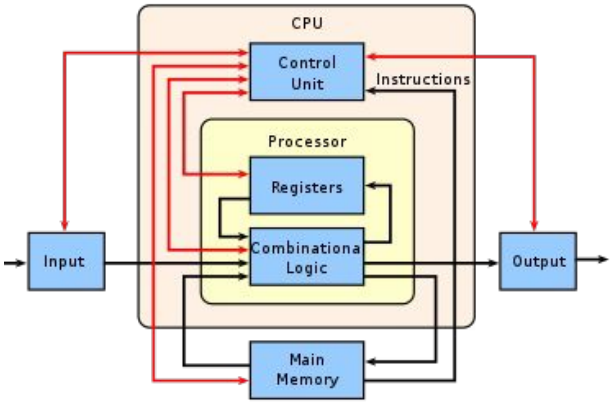


Running ML on Hardware

SLAC

Calculation ⟷ **Computational Resources**

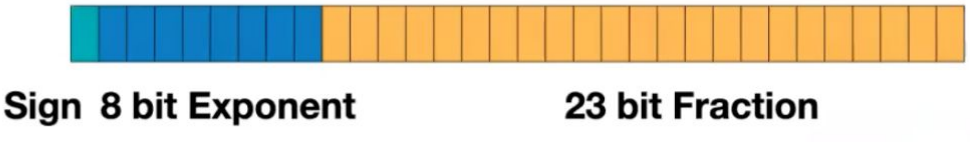




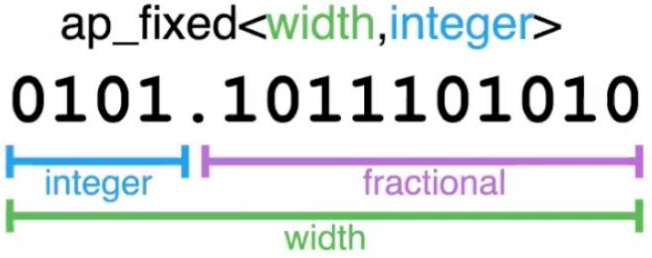
➡How can I tweak my ML model to fit a set of computational resources without sacrificing performance where I need it? (latency, power efficiency…)

Quantization

- **What is it?** Minimize amount of computation needed to evaluate network by *reducing number of bits* that describe calculation inputs
- **How do I do it?** Quantize model inputs, weights, and biases to lower precision
  - Baseline: 32-bit floating point precision



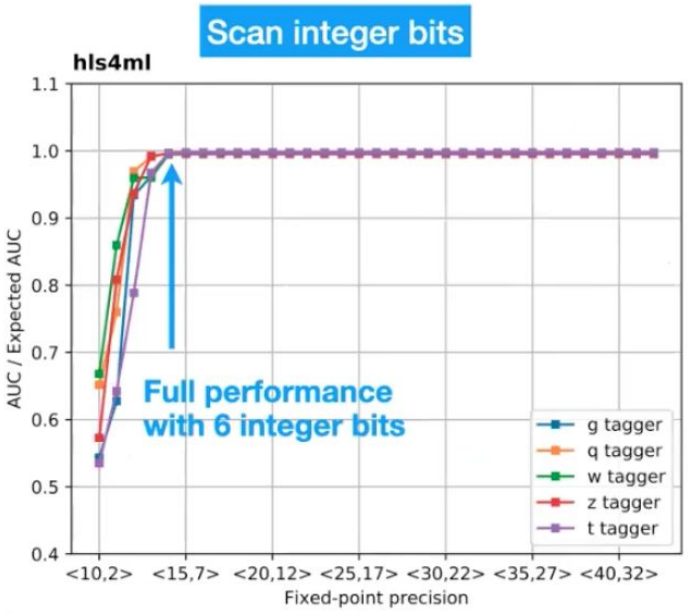
  - Quantized: N-bit fixed-point precision

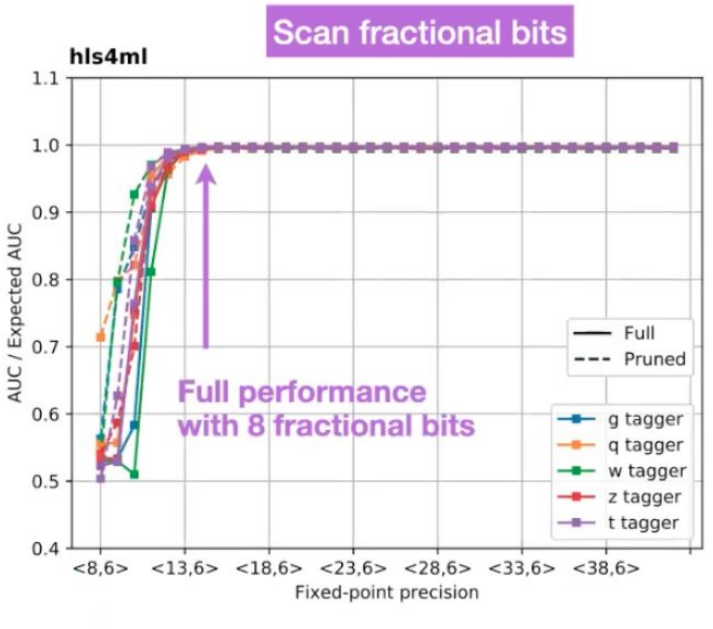


J. Duarte

Post-Training Quantization

- **Post-training quantization (PTQ):** quantize weights/biases on pre-trained model
- General strategy: avoid overflows in integer bit then scan the decimal bit until reaching optimal performance





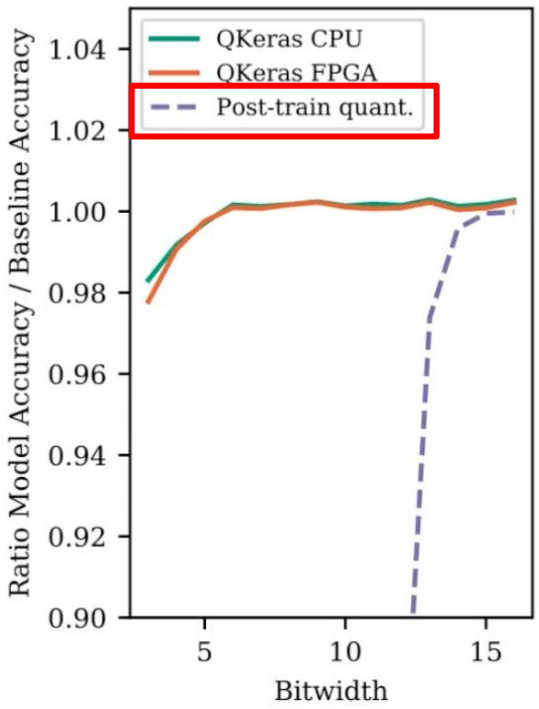
J. Duarte

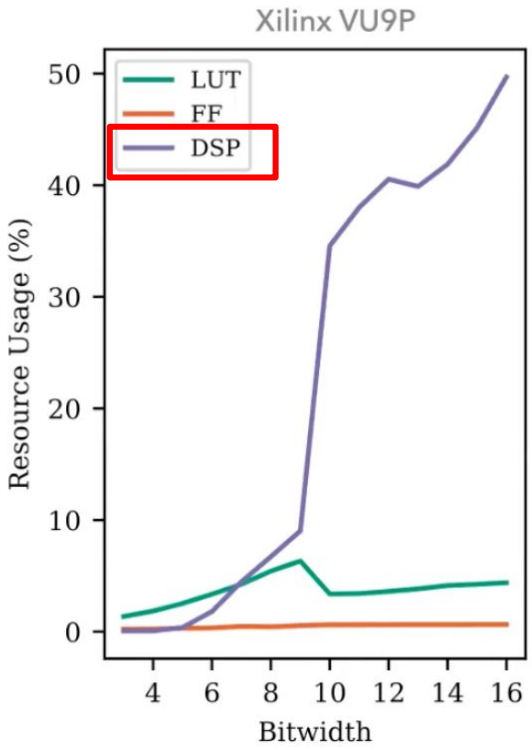
Quantization Aware Training

SLAC

- QKeras is a library to train models with quantization **in training**
  - Developed & maintained by Google
- Easy to use, drop-in replacements for Keras layers
  - e.g. Dense → QDense, Conv2D → QConv2D
  - Use 'quantizers' to specify how many bits to use where
- With QAT, full performance achievable with 6 bits instead of 14 bits
  - Much smaller fraction of resources required

2006.10159

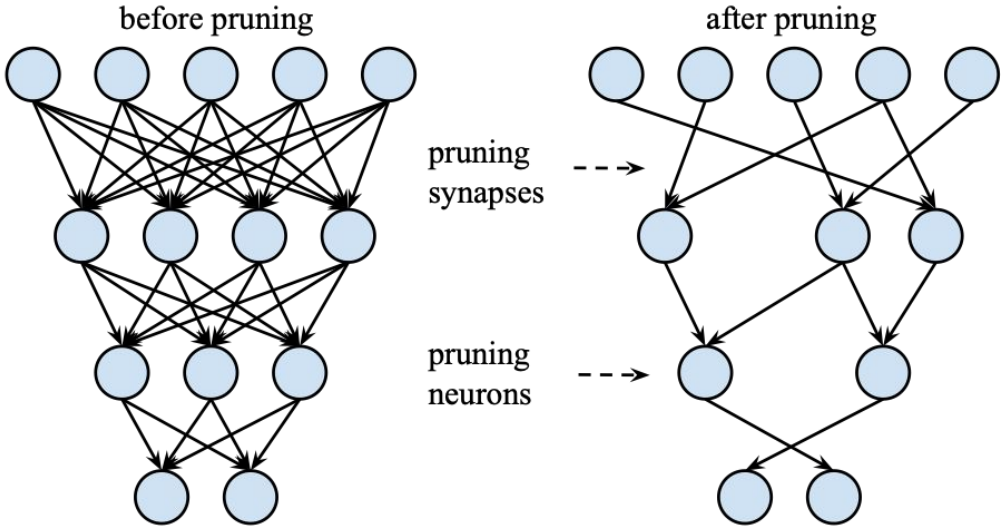




J. Duarte

Pruning

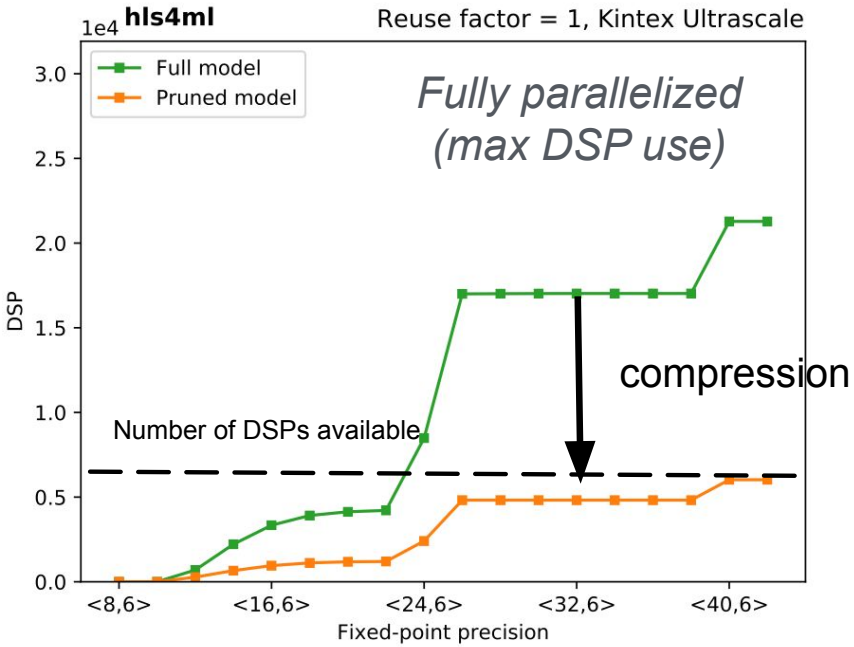
- **What is it?** Minimize number of calculations needed to evaluate network by removing low-impact nodes
- **How do I do it?** Iterative training to downweight unimportant synapses
  - Ex: L1 regularization $L_\lambda(\boldsymbol{w}) = L(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1$ $\|\boldsymbol{w}\|_1 = \sum_i |w_i|$



Pruning In Practice

- Can significantly reduce valuable resources with ~minimal hit to performance

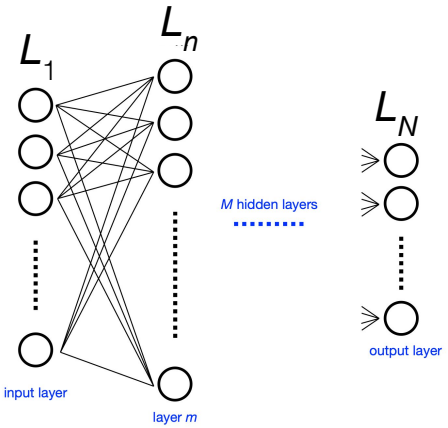
```
from tensorflow_model_optimization.python.core.sparsity.keras import prune, pruning_callbacks, pruning_schedule
from tensorflow_model_optimization.sparsity.keras import strip_pruning

pruning_params = {"pruning_schedule": pruning_schedule.ConstantSparsity(0.75, begin_step=2000, frequency=100)}
model = prune.prune_low_magnitude(model, **pruning_params)
```



70% compression ~ 70% fewer DSPs

J. Duarte

High Level Synthesis



$$N_{\text{multiplications}} = \sum_{n=2}^{N} L_{n-1} \times L_n$$

$$\mathbf{x}_n = g_n(\mathbf{W}_{n,n-1}\mathbf{x}_{n-1} + \mathbf{b}_n)$$

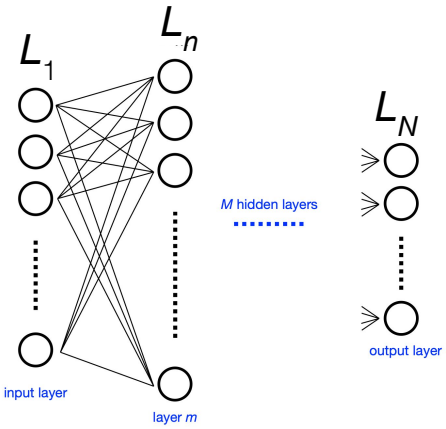
precomputed and stored in BRAMs

DSPs

logic cells (LUTs)

How many resources? DSPs, LUTs, FFs?
Does the model fit in the latency requirements?

High Level Synthesis



$$N_{\text{multiplications}} = \sum_{n=2}^{N} L_{n-1} \times L_n$$

$$\mathbf{x}_n = g_n(\mathbf{W}_{n,n-1}\mathbf{x}_{n-1} + \mathbf{b}_n)$$

precomputed and stored in BRAMs

DSPs

logic cells (LUTs)

How many resources? DSPs, LUTs, FFs?
Does the model fit in the latency requirements?

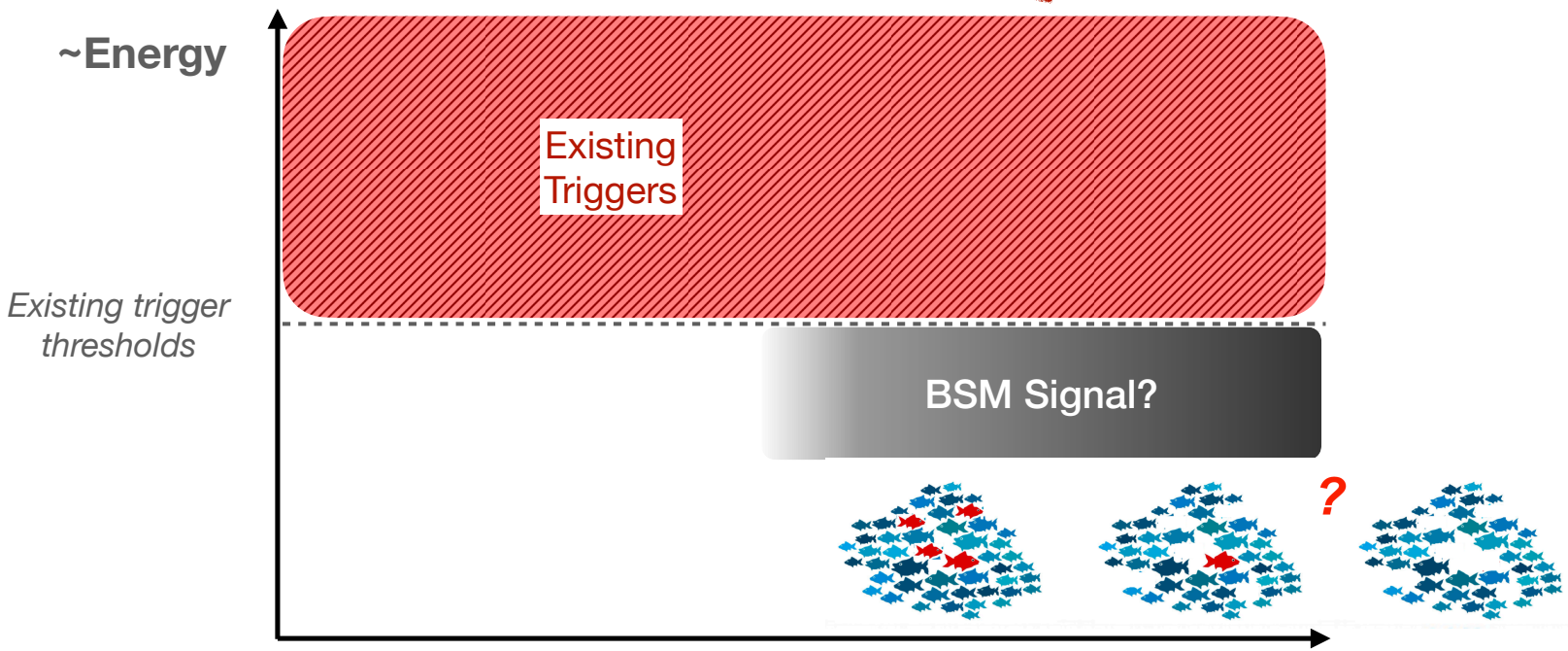
➡Frameworks (like hls4ml or SNL) allow you to compile your ML algorithm from python through C++ to a hardware description language (HDL)

Examples from HEP

Real-time Anomaly Detection

SLAC

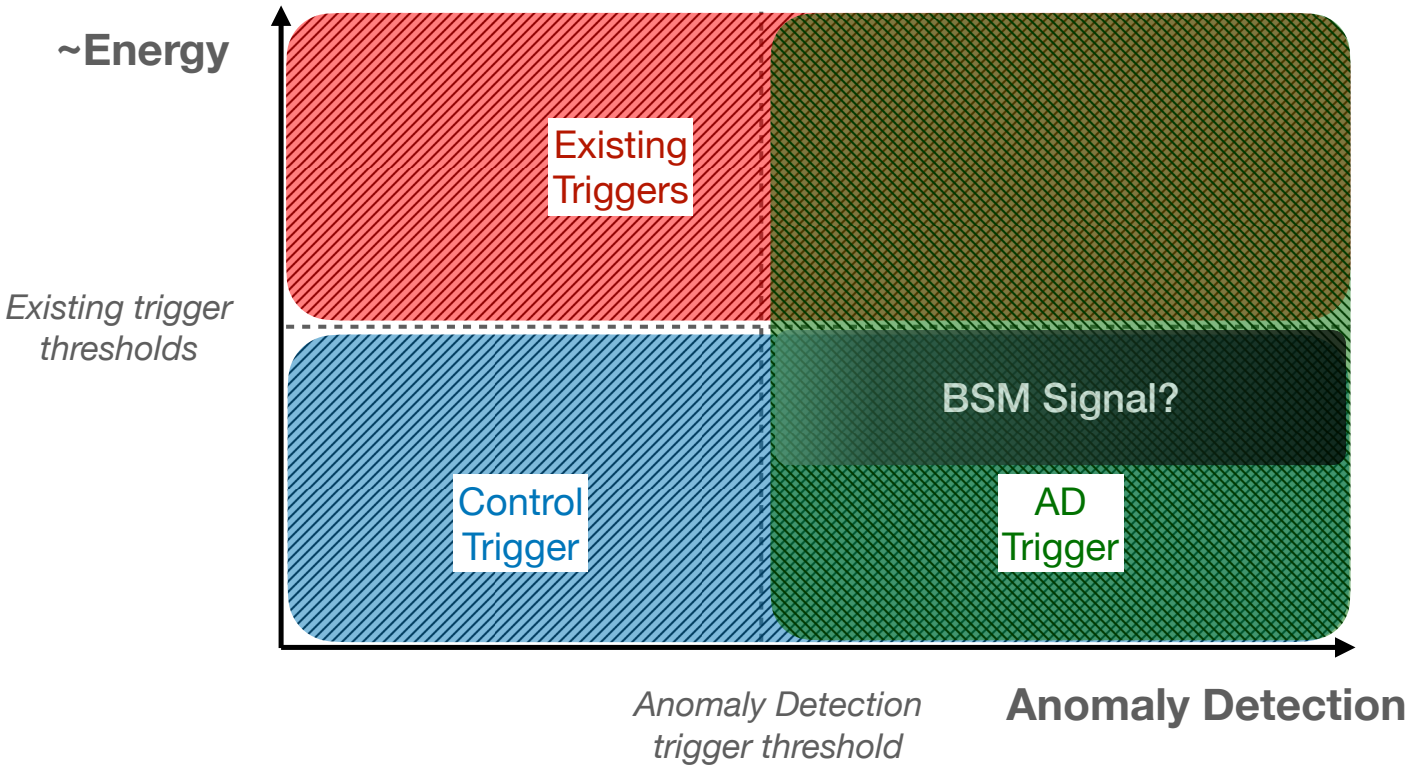
AD here is still very interesting but doesn't need new triggering strategy



D. Rankin

Real-time Anomaly Detection

SLAC

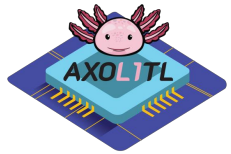


D. Rankin

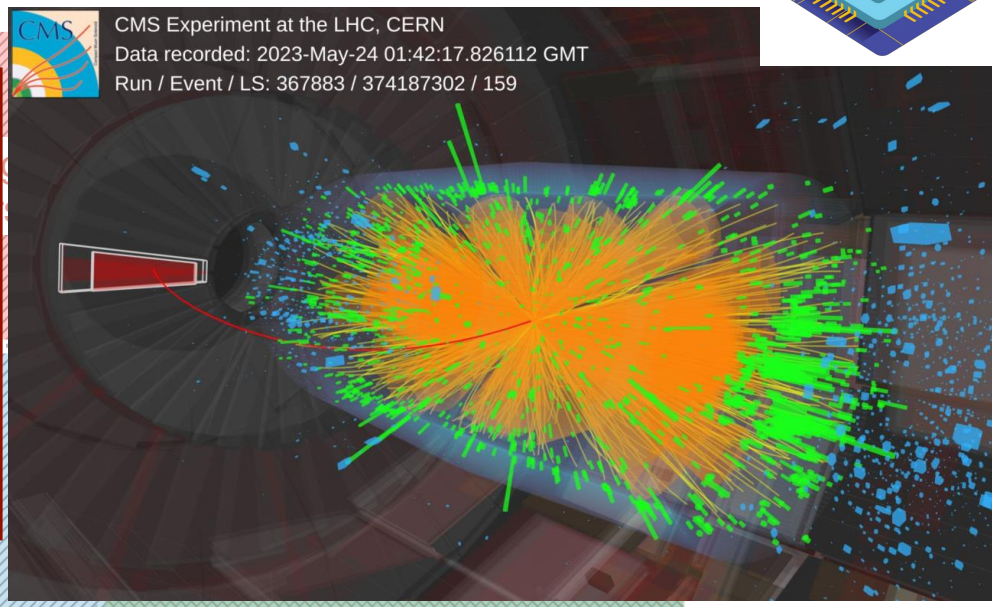
Real-time Anomaly Detection

SLAC

CMS Most Anomalous Event



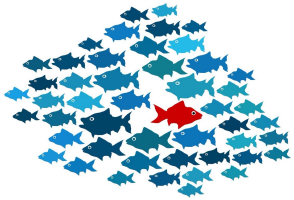
- Proof of concept from CMS experiment already running in test crate in Run 3 [C. Sun]

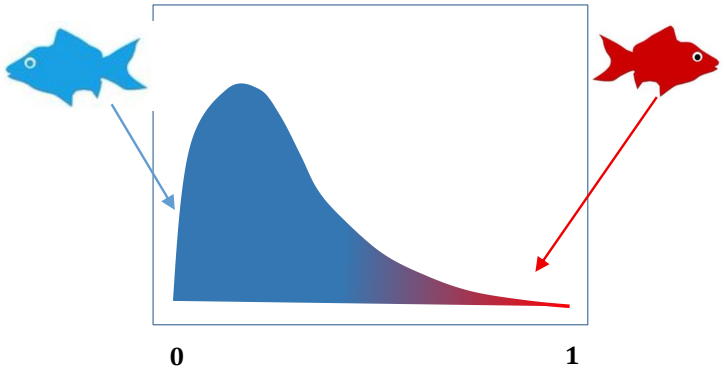


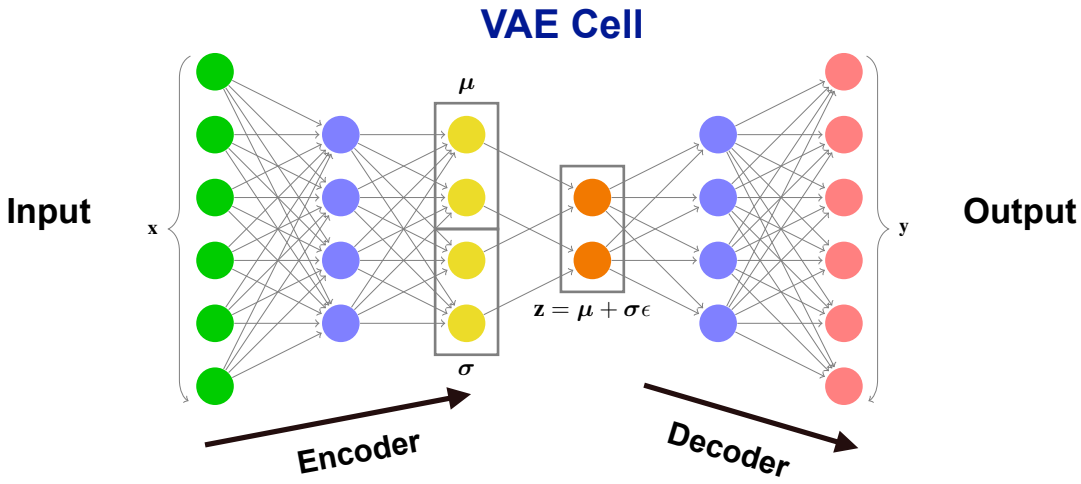
D. Rankin

VAEs for Anomaly Detection Triggers

- Propose **variational autoencoder**-based AD at both L1 (hardware/~few ns) and HLT (software/~few ms)



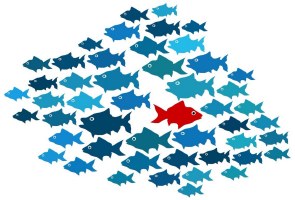


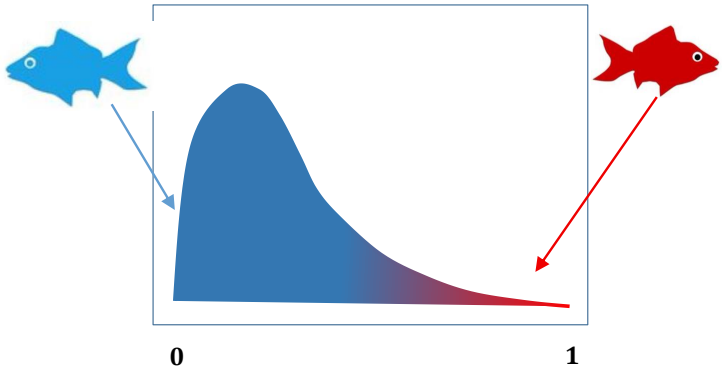


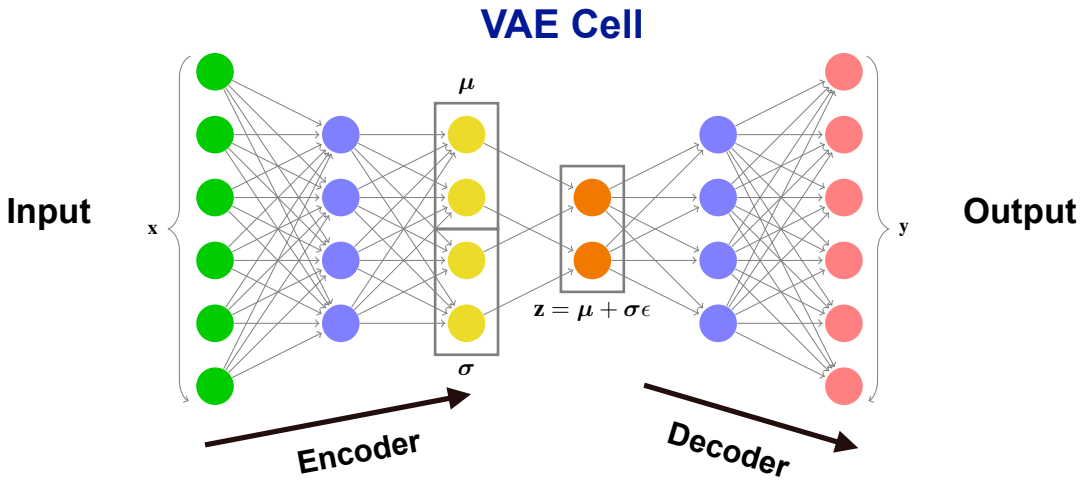
$$\mathcal{L} = |\mathbf{y} - \mathbf{x}|^2 + D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

VAEs for Anomaly Detection Triggers

- Propose **variational autoencoder**-based AD at both L1 (hardware/~few ns) and HLT (software/~few ms)





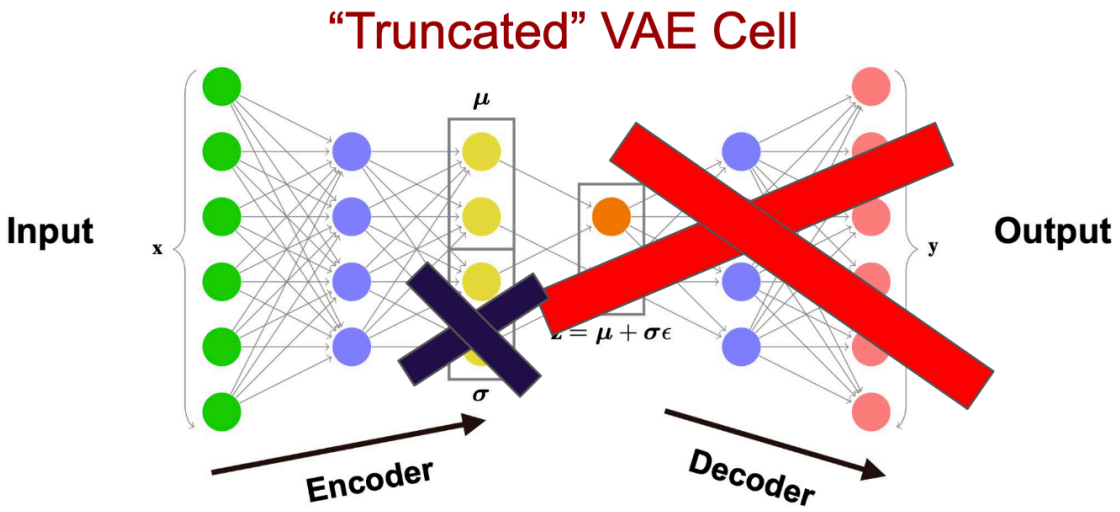


$$\mathcal{L} = |\mathbf{y} - \mathbf{x}|^2 + D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

➡How to evaluate in < 25 ns?

VAEs for Anomaly Detection Triggers

- Propose **variational autoencoder**-based AD at both L1 (hardware/~few ns) and HLT (software/~few ms)
- Latency < 25 ns requires considerable model compression
  - "Truncated" VAE: remove decoder
  - "Clipped" loss: only evaluate KL divergence terms using mean μ
  - At fixed<16,10> precision latency estimate in O(ns)! ✅



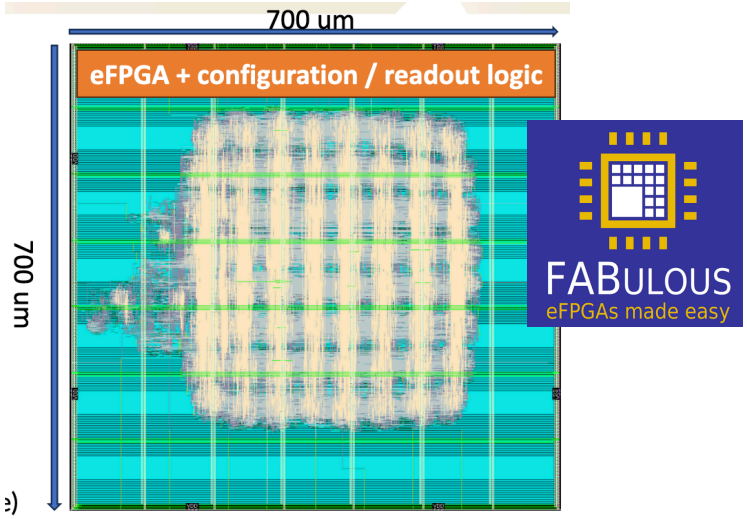
"Clipped" KL divergence

$$2 \cdot \mathrm{KL} = \mu^2 + \sigma^2 - 1 - \log \sigma^2$$

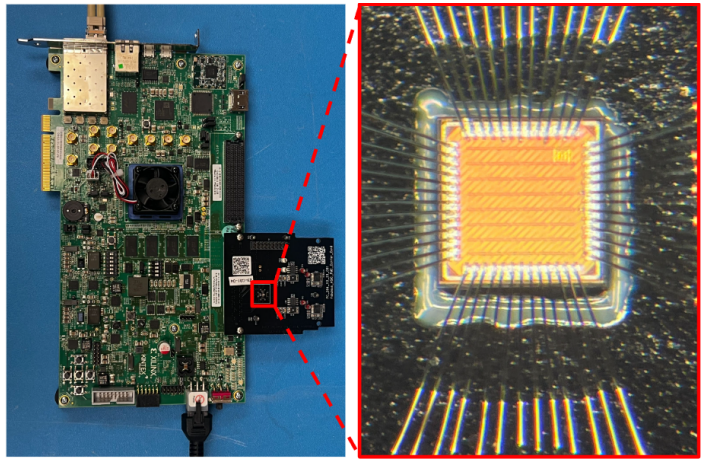
ML at the Edge: eFPGAs

SLAC

- **“Embedded” FPGAs**: reconfigurable logic in ASIC design for configurability ease of FPGA with low power/footprint of chip
- Open-source frameworks (eg. FABulous) allow for lowered barrier to entry for ASIC design!



28nm eFPGA Testboard



➡SLAC proof-of-concept for open-source design tools for eFPGAs ✅

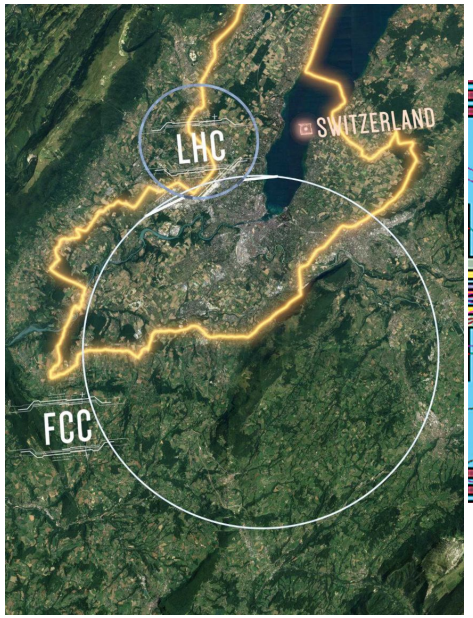
- Next steps: eFPGAs for “extreme environments” (high radiation, cryogenics)

[2404.17701]

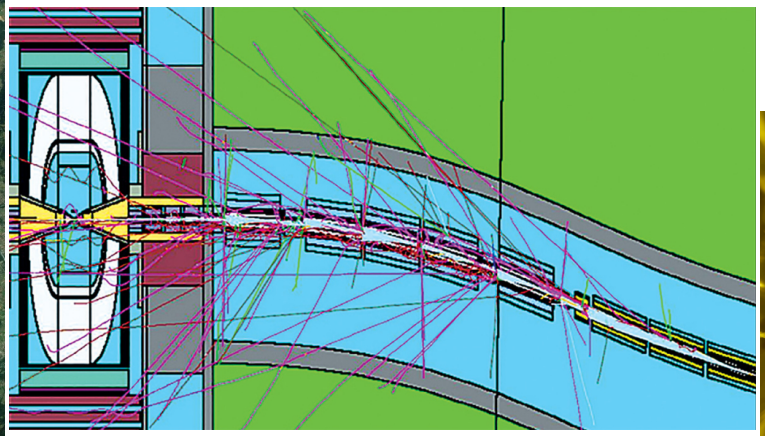
Future Experiments

SLAC

- Next-generation international HEP facilities will present even more data/electronics challenges
  - **Future Circular Collider (FCC)**: 91km tunnel with expected exascale data rates in hadron-hadron operation stage
  - **Muon collider**: real-time management of intense beam-induced and muon decay backgrounds
  - **Deep Underground Neutrino Experiment (DUNE)/DM detection**: precise/low-noise cryogenic readout

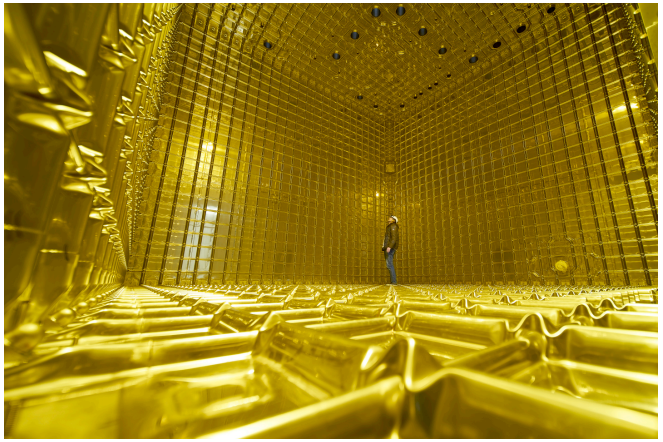


FCC



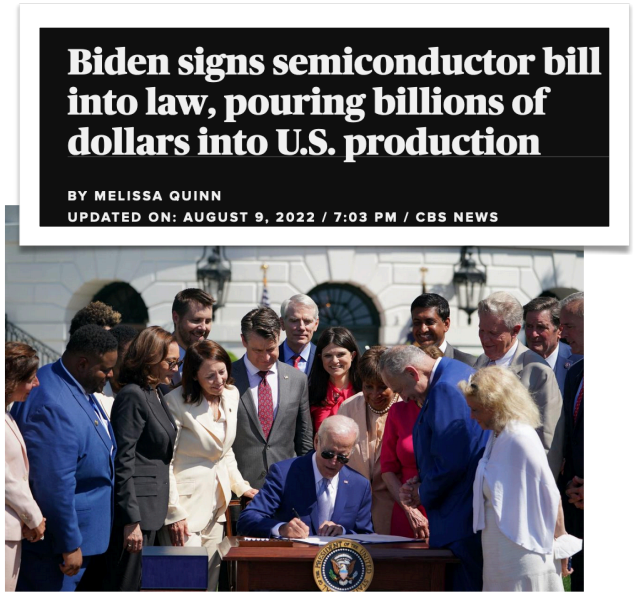
Muon Collider

DUNE



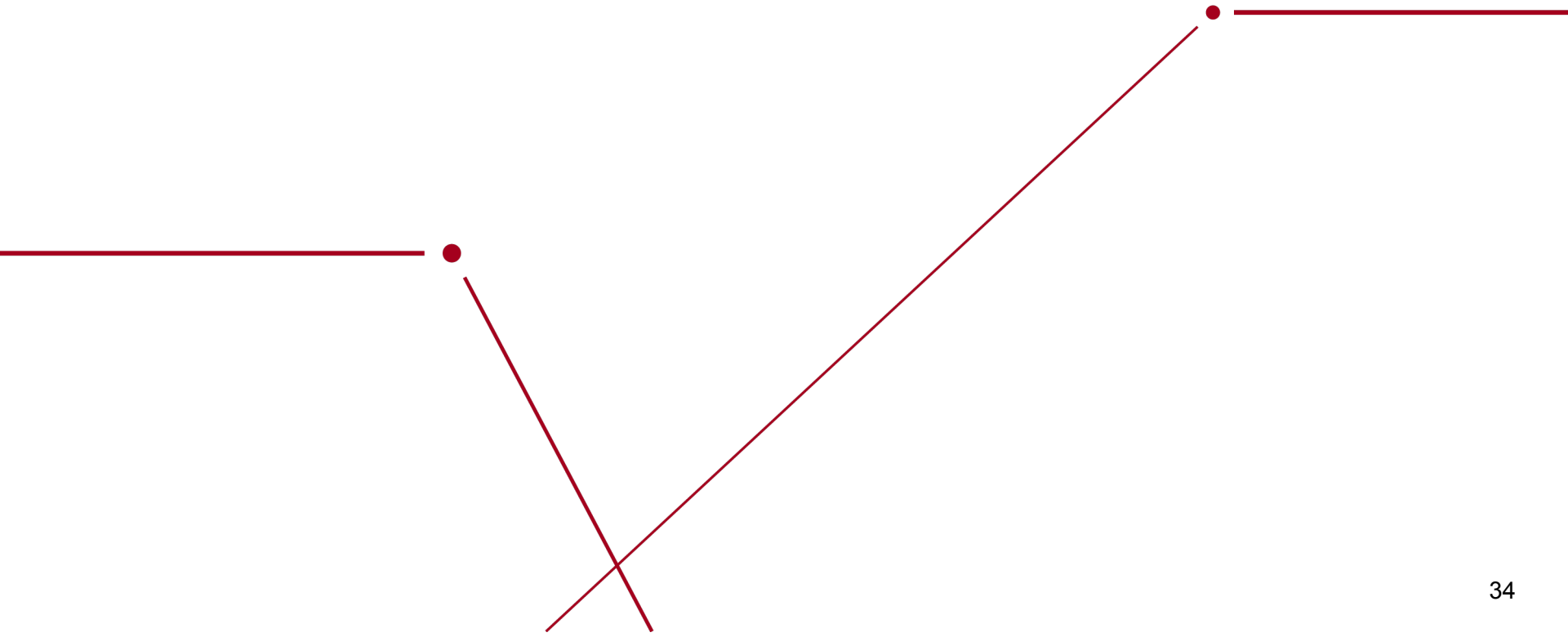
Conclusions

- **Machine learning and silicon microelectronics** are key interdisciplinary technologies with entire industries of their own
  - Building the best **particle physics instruments** requires us to pull from the latest & greatest in these areas!
- Fast ML in HEP electronics is a rapidly growing area of interest (with lots of room to contribute & **easy access**)

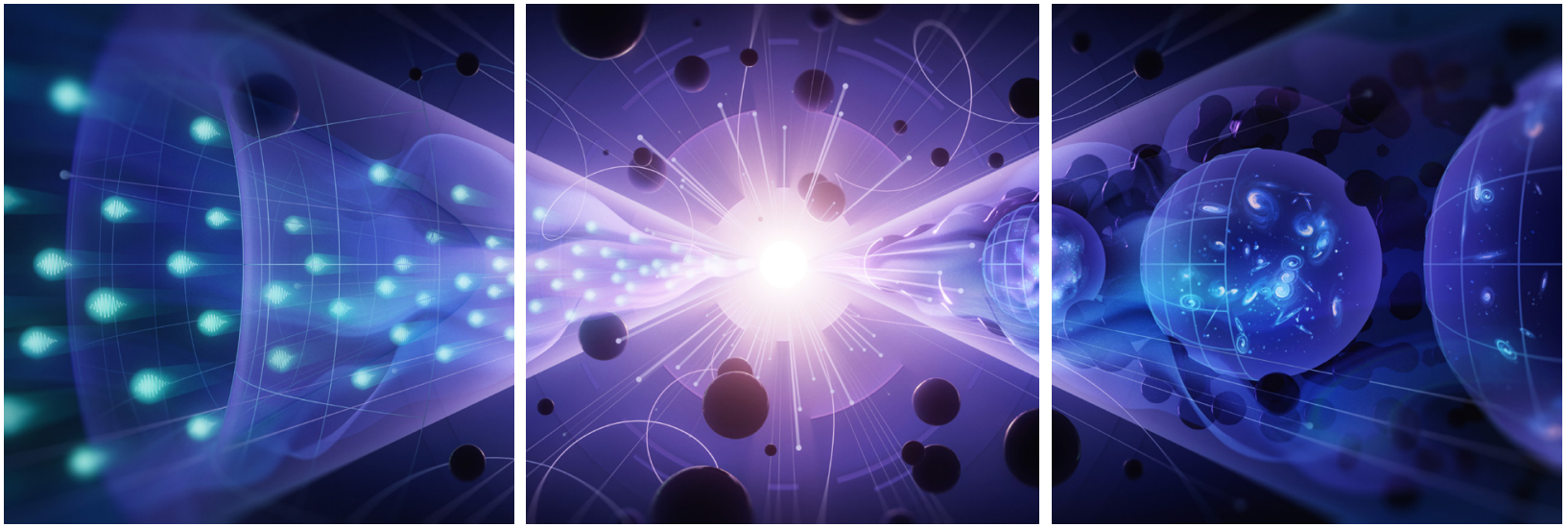


President Biden signs the Chips and Science Act of 2022 on the South Lawn of the White House on Aug. 9, 2022.

Backup



P5 2023



Decipher the Quantum Realm

Elucidate the Mysteries of Neutrinos

Reveal the Secrets of the Higgs Boson

Explore New Paradigms in Physics

Search for Direct Evidence of New Particles

Pursue Quantum Imprints of New Phenomena

Illuminate the Hidden Universe

Determine the Nature of Dark Matter

Understand What Drives Cosmic Evolution

P5 Project Recommendations

SLAC

Ongoing (Rec 1)

- **HL-LHC**
- Dune Phase 1
- Vera Rubin/LSST
- Smaller projects: ex. NOvA, IceCube, SuperCDMS, Belle II, LHCb, Mu2e, etc.

Construction (Rec 2)

1. CMB-S4
2. DUNE Phase-II
3. **Off-shore Higgs factory**
4. Gen-3 direct detection DM (preferably US-sited)
5. IceCube-Gen2

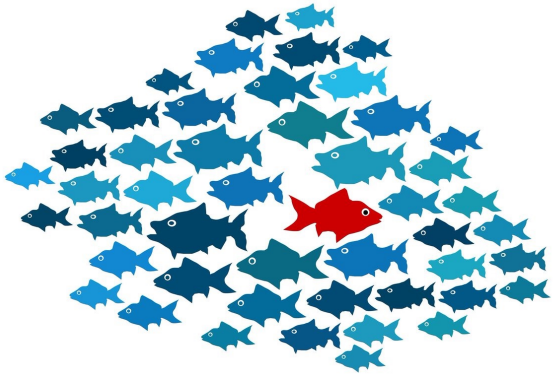
R&D (Rec 4)

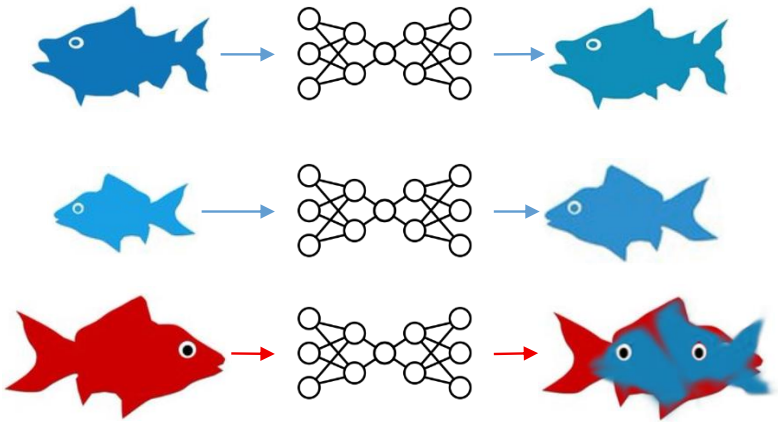
- **Cost-effective 10 TeV pCM collider: demonstrator within 10 years**
- Theory
- General Accelerator R&D (GARD)
- Instrumentation for scientific tools
- **Detectors for Higgs factory & 10 TeV pCM**
- Cyberinfrastructure/novel data analysis
- Fermilab accelerator complex

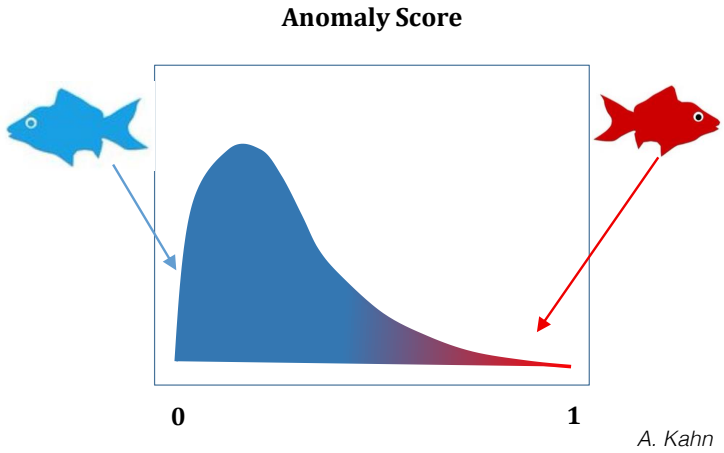
Machine Learning & Anomaly Detection

SLAC

- **Collaboration**: computer/data scientists with sophisticated machine learning algorithms for signal model-independent *anomaly detection*
  - Goal: identify features of the data that are inconsistent with a background-only model
- Strategy: train an ML architecture to reconstruct its input
  - Unsupervised: train over unlabeled events (data)
  - Rarer events with unusual features will be poorly reconstructed → reconstruction accuracy is a good discriminant







A. Kahn