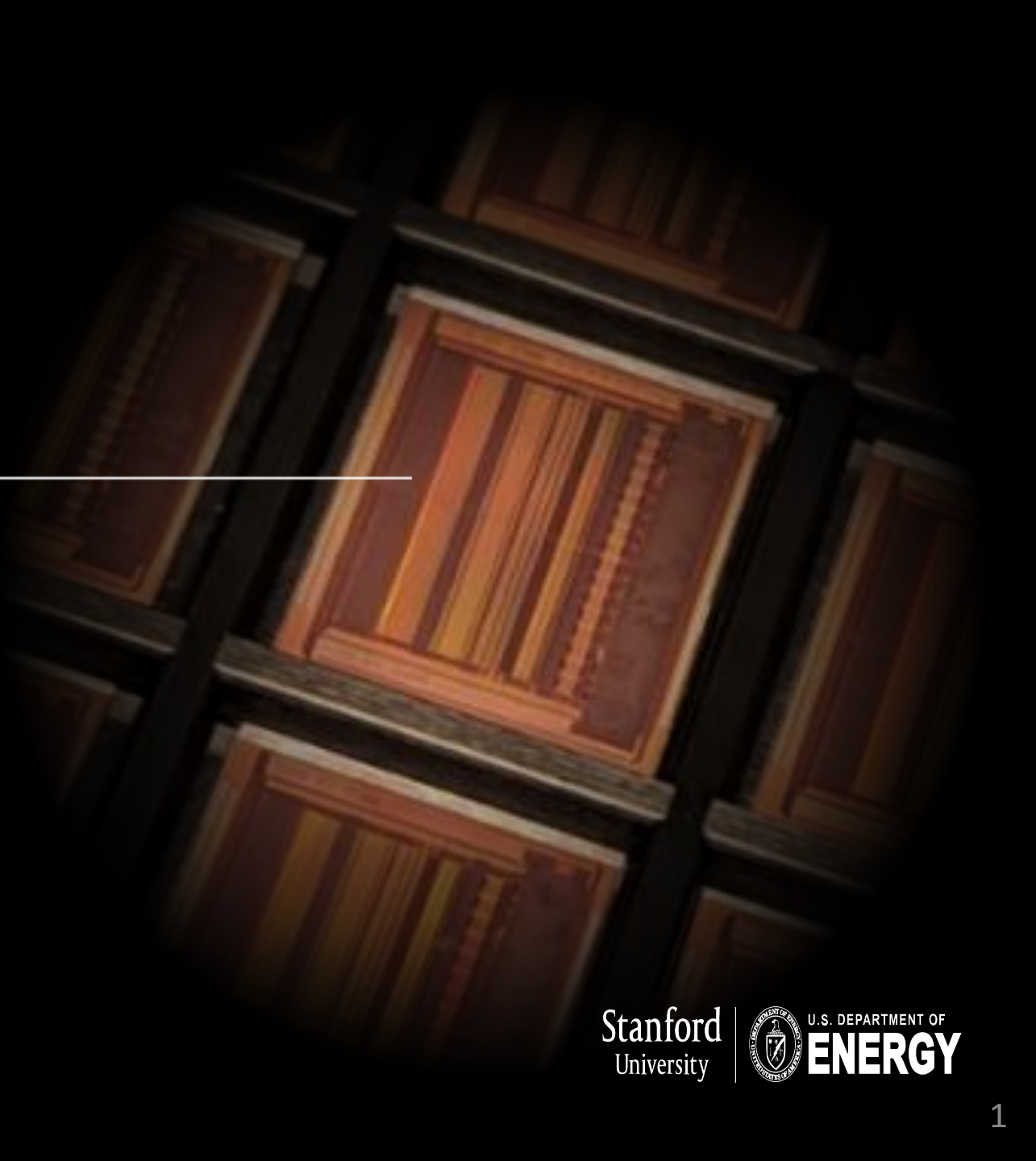
ML4FE Technology Session: Overview of challenges and goals

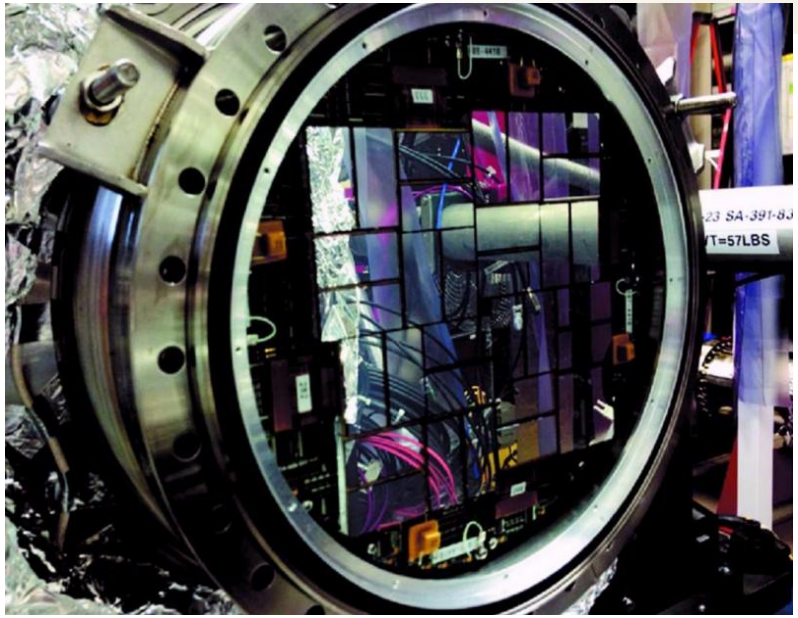
SLAC TID

Ryan Herbst On Behalf Of SLAC TID

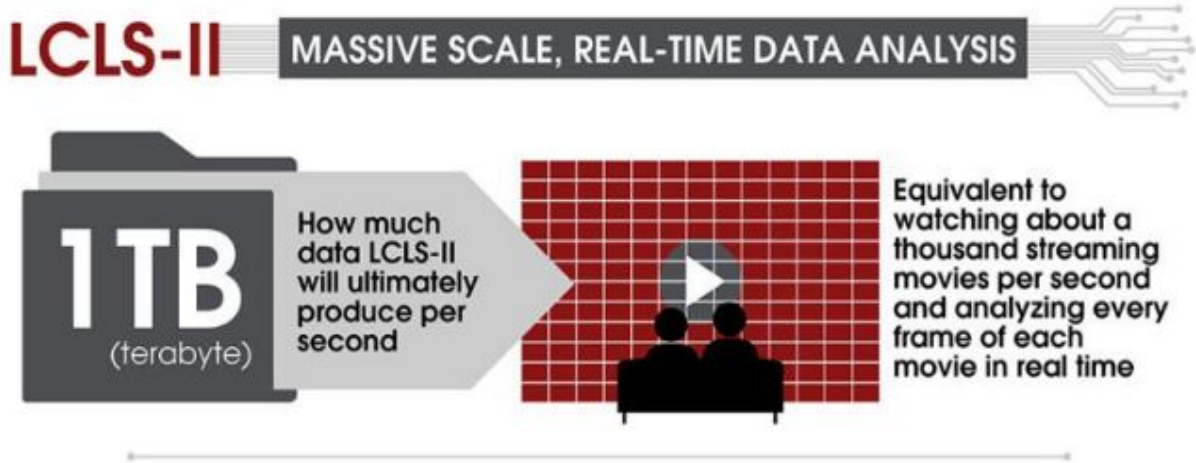
May 19 2025

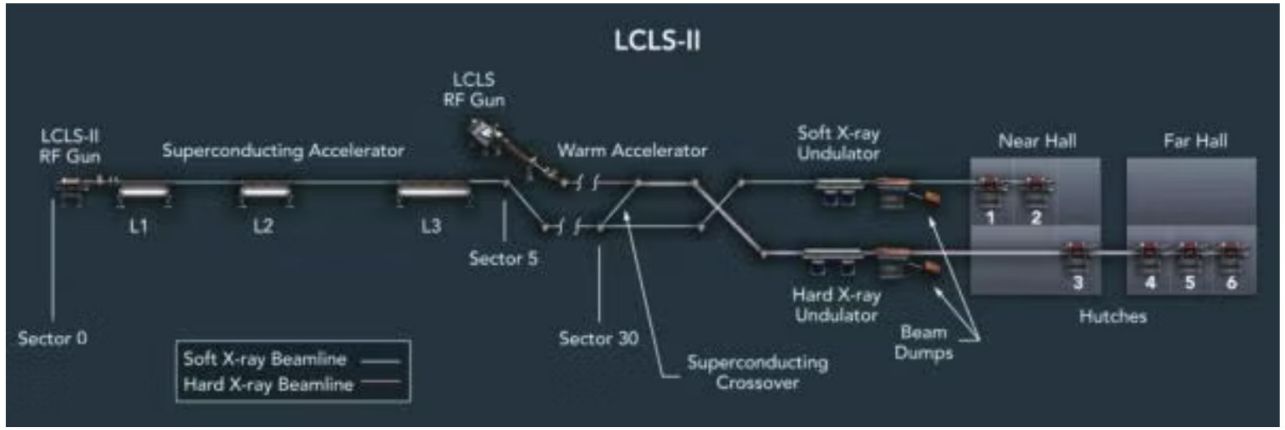


SLAC LCLS Data Challenge



https://www6.slac.stanford.edu/news/2021-02-17-bigger-faster-more-powerful-slacs-new-x-ray-laser-data-system-will-process-million





LCLS-II beamlines. (SLAC National Accelerator Laboratory)

Example Challenge: Data Streaming and Storage In LCLS-II

Image A

AI generated image

Image B

AI generated image

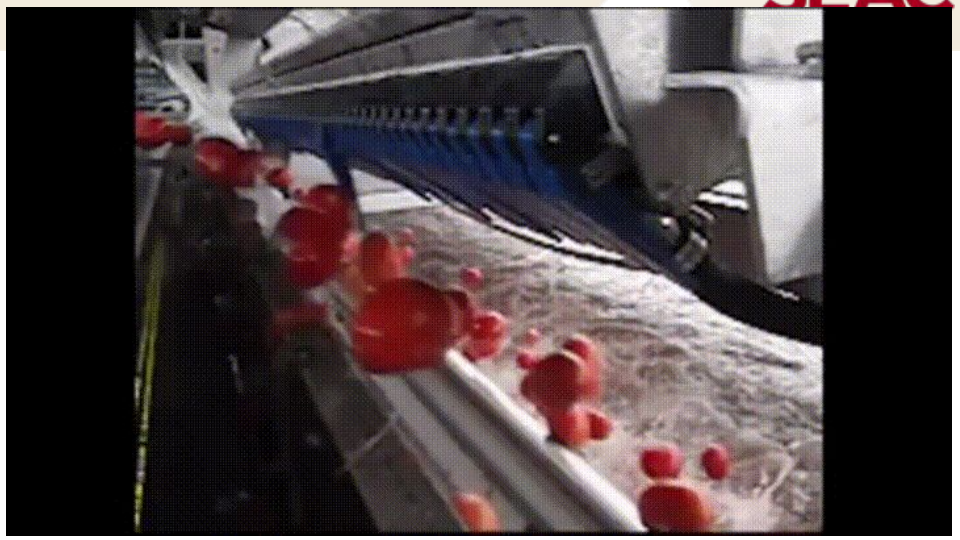
LCLS-I

- Generating & storing data at GB/sec

LCLS-II

- Generating data at TB/sec
- Streaming the raw data at every microsecond will overwhelm the system
- Storing this much raw data is not possible and could cost billion dollars annually

Goals For LCLS-II Data Processing.. And Other Applications



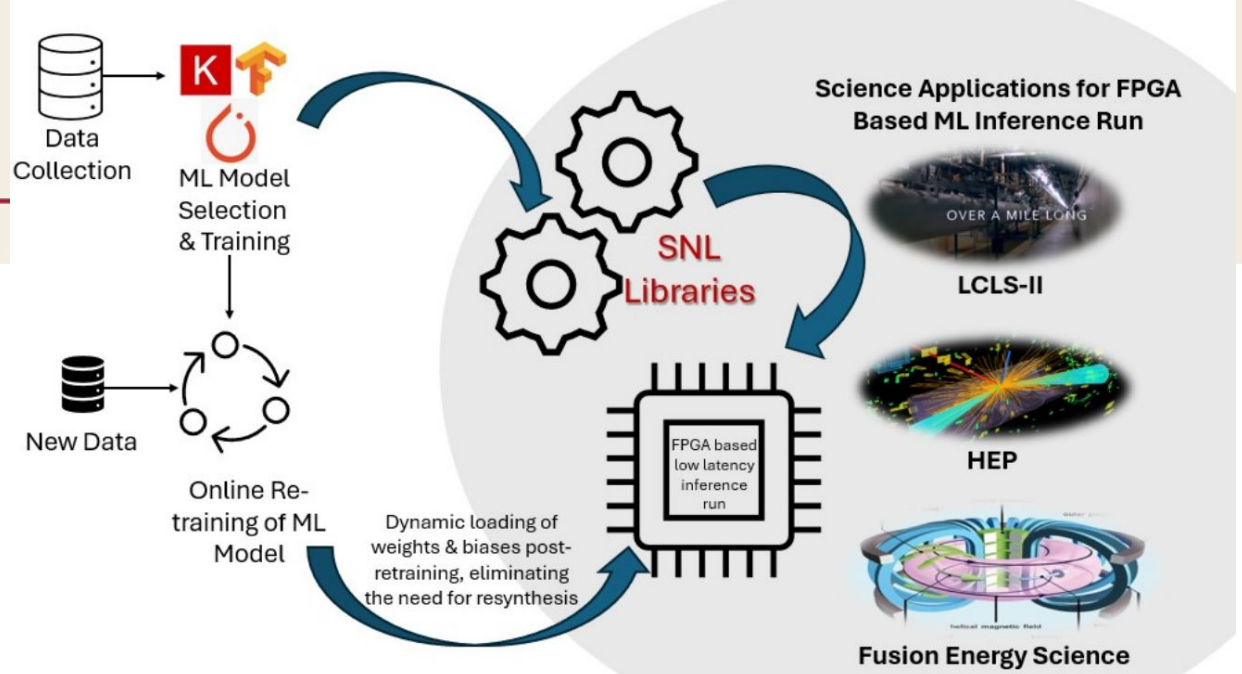
- The main goal of our research is to autonomously select data which contains valid scientific data, removing the unwanted data
- Machine Learning (ML) has demonstrated the potential to digest large datasets to extract relevant insights, or do the classifications for hit or miss targets
- Data volumes and image rates necessitate processing at the edge, near or on the detector embedded within the devices, as close to the sensor as possible.

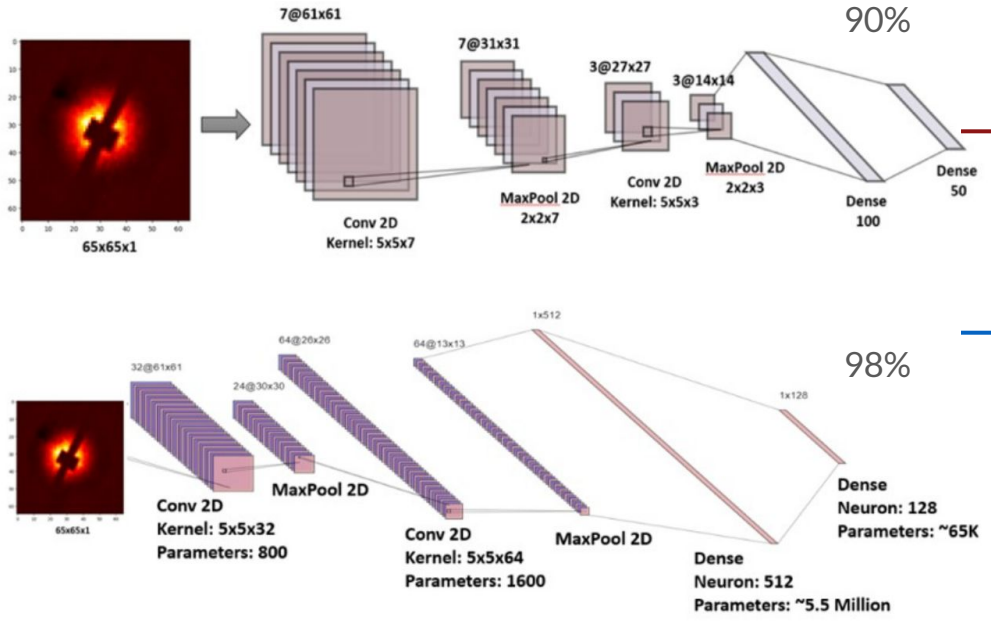
EdgeAI is Heterogeneous

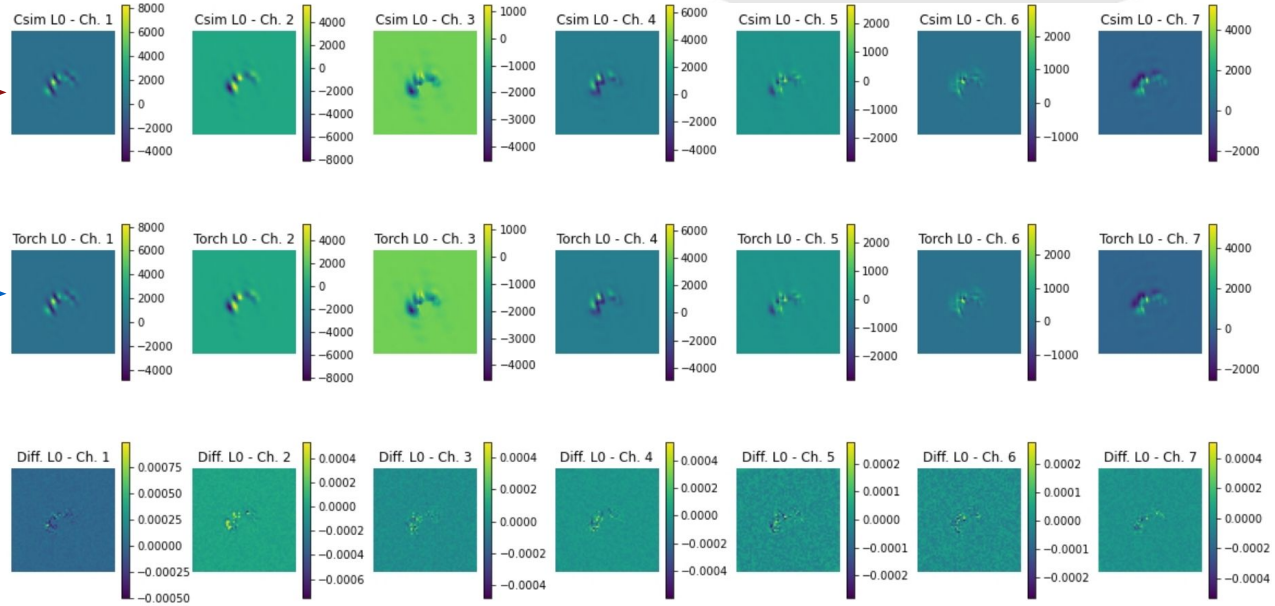
Data Rates are superhuman

Beamline optimization happens at human timescales

The firehose is filling drives with $#!+







A Dave, *et al.*, "FPGA-Accelerated SpeckleNN with SNL…," arXiv 2502.19734

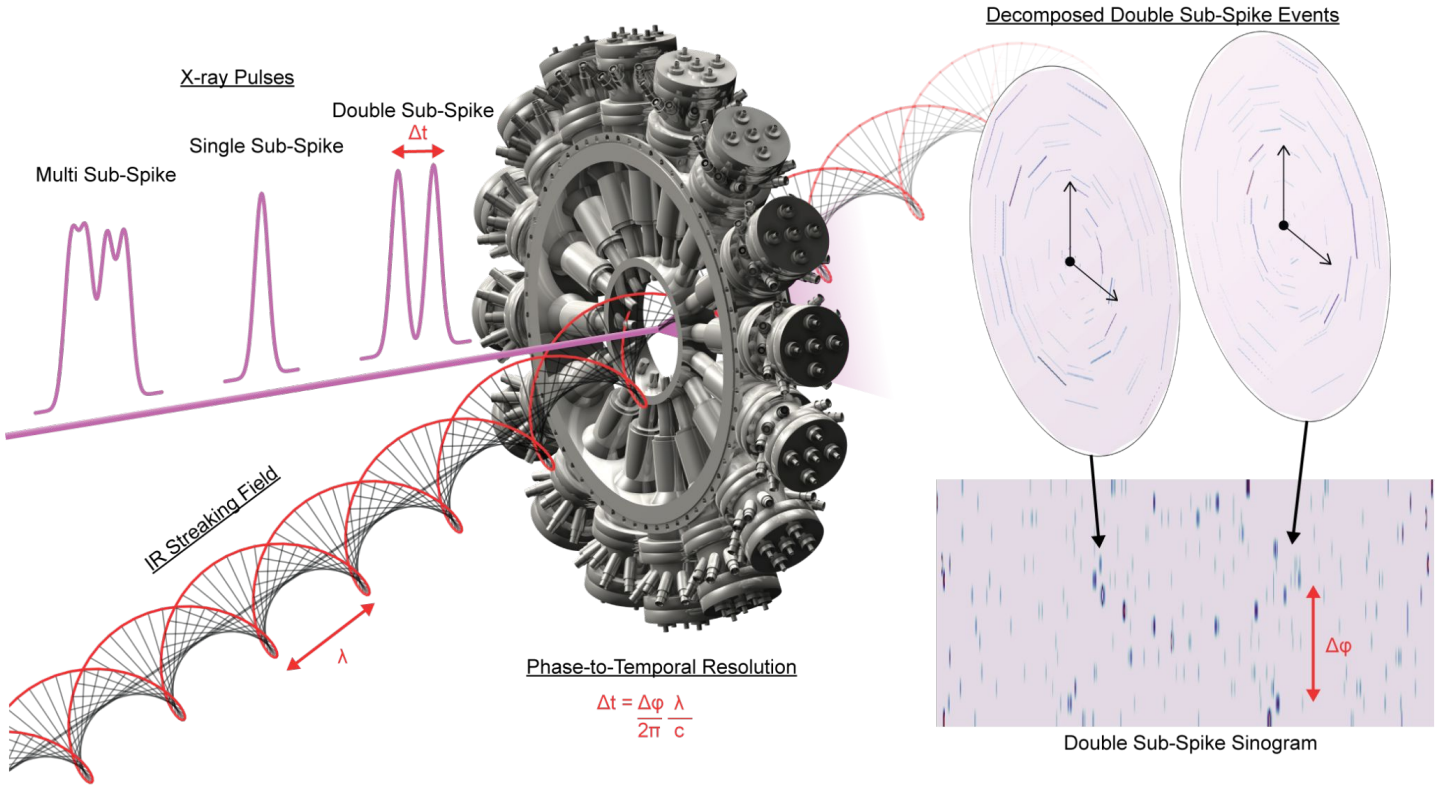
CookieBox Example – “Attosecond Streaking”

Need for MHz inference

Inverse problem of pulse reconstruction

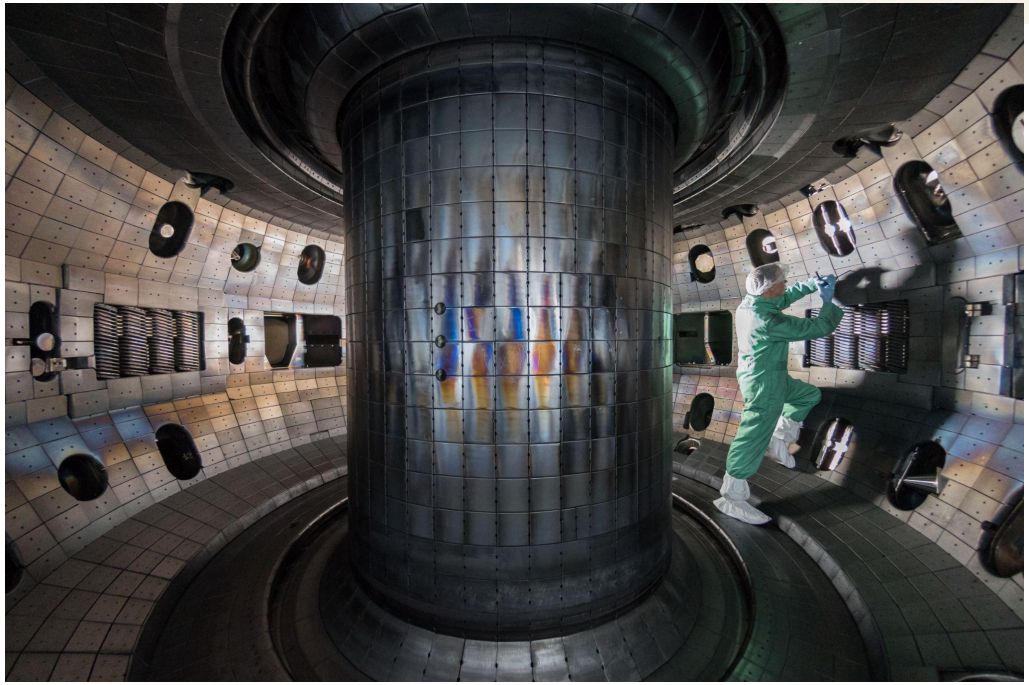
Enables time-bandwidth version of “super-resolution”

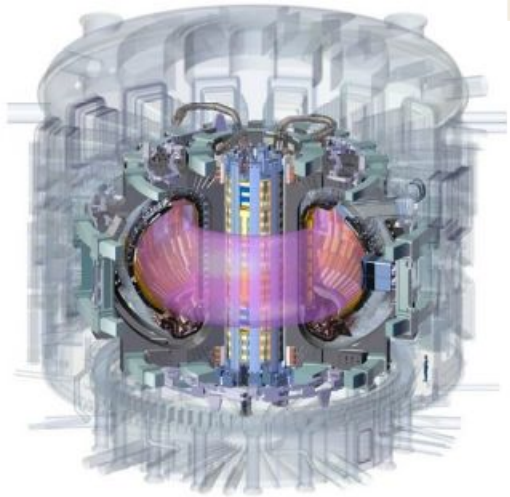
Where to produce which links of the inference chain?



Hirschman *et al.*, “A Hybrid Neural Network for High-Throughput Attosecond Resolution Single-shot X-ray Pulse Characterization”, (2025) arXiv 2502.16141

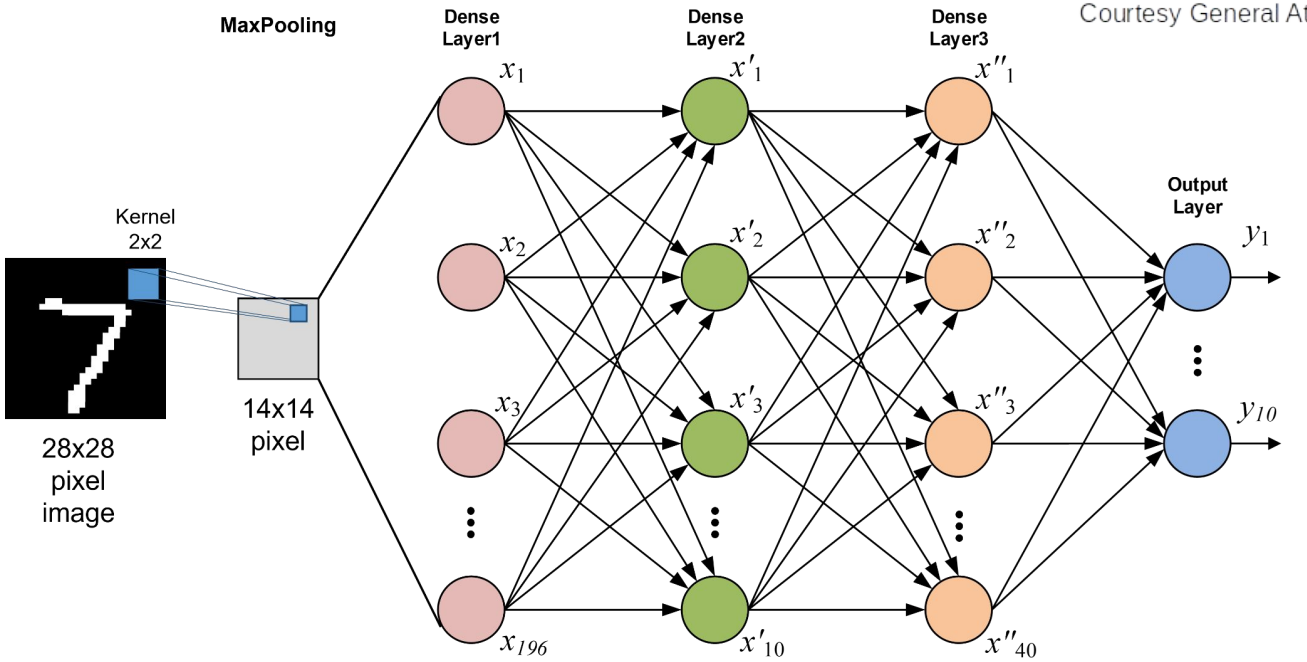
ML For Fusion Reactors - DIII-D Example



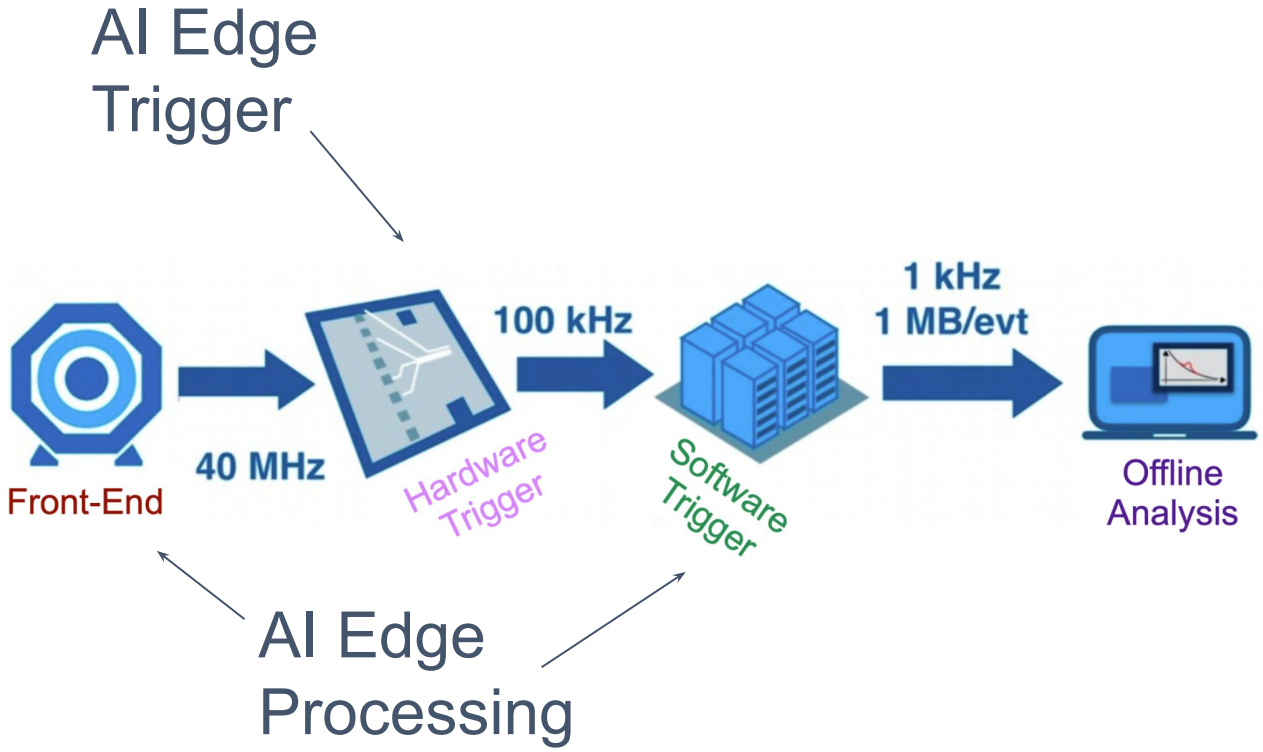


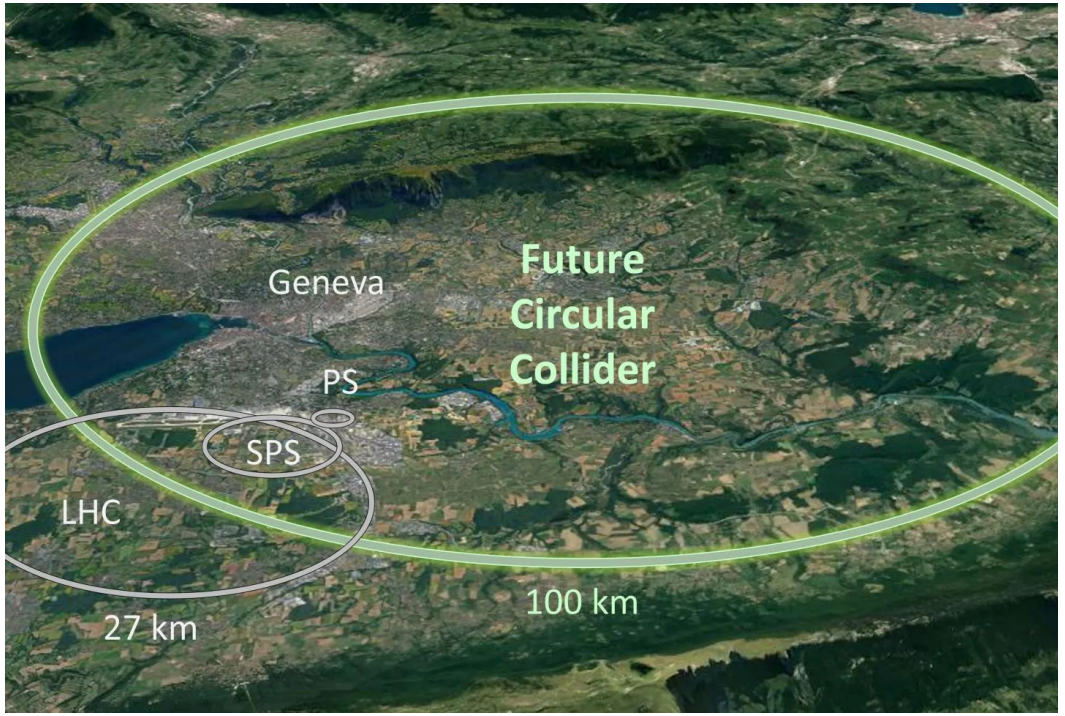
Courtesy General Atomics

- Beam Emission Spectroscopy (BES) for tokamak **disruption event identification** (Edge Localized Mode - ELM)
- Developed with intention of **forecasting ELMs**
- L. Malhotra et al., APS DPP Meeting 2021, CP11.67
- L. Malhotra et al., IAEA FEC 2023, Contribution IAEA-CN-307/TH/P8-6

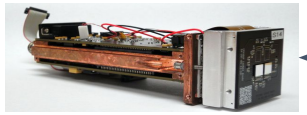


LHC - FCC Trigger & Data Reduction



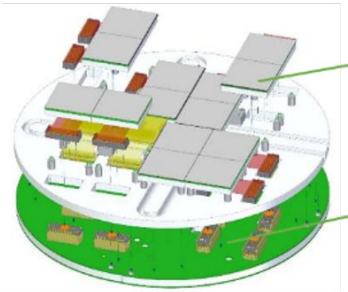


Evolution Of Data Processing: LCLS-2



On detector processing using classical and ML based approaches. FPGA + ASIC + co-processor

In gateway FPGA processing using classical and ML based approaches (SNL)



ASIC Level

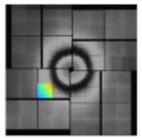
FPGA Level

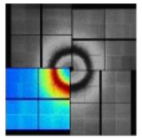
EDGE Computing on camera

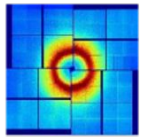
Farm of FPGAs

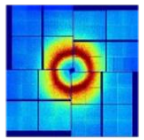
Data System

CPUs/GPUs









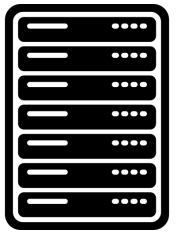
Rate reduction

- Application specific
- Limited number of techniques
  - Sparsification,
  - Event driven triggered based techniques,
  - Back-end zero suppression
  - Region of Interest (RoI)

- Algorithms can be tailored
- Limited number of techniques
  - Back-end zero suppression
  - Region of Interest (RoI)

- Algorithms can be tailored to different applications
- Fast feedback to the detector (trigger generation)
- Calibration (required)
- Large number of lossless techniques

Direct to disk storage. Processing done offline (LCLS & slower imagers)



Online CPU based data processing

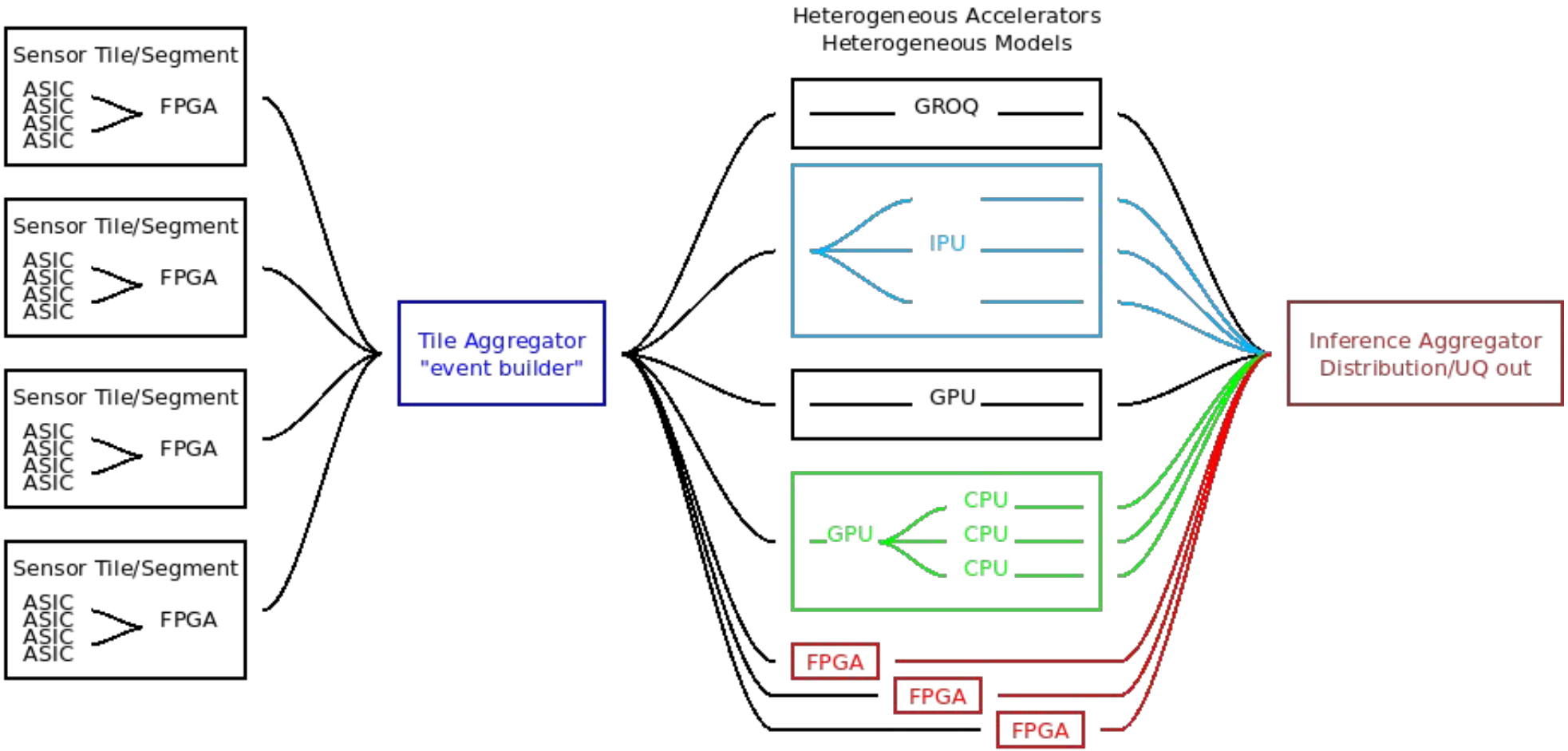
GPU Based Online Processing

Other commercial hardware processing (Groq, etc)

This application described in this talk supports 35Khz events from 5 FPGA boards = 50GB/s processed in a single GPU with an output bandwidth of 5GB/s

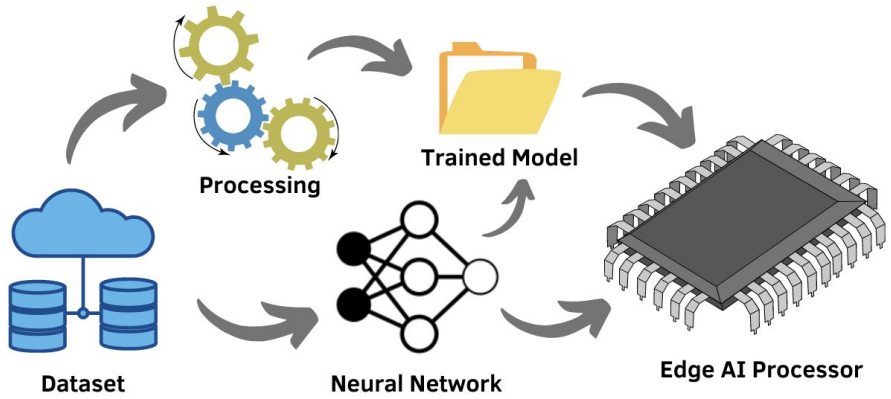
HeteroFlow Streaming Pipelines

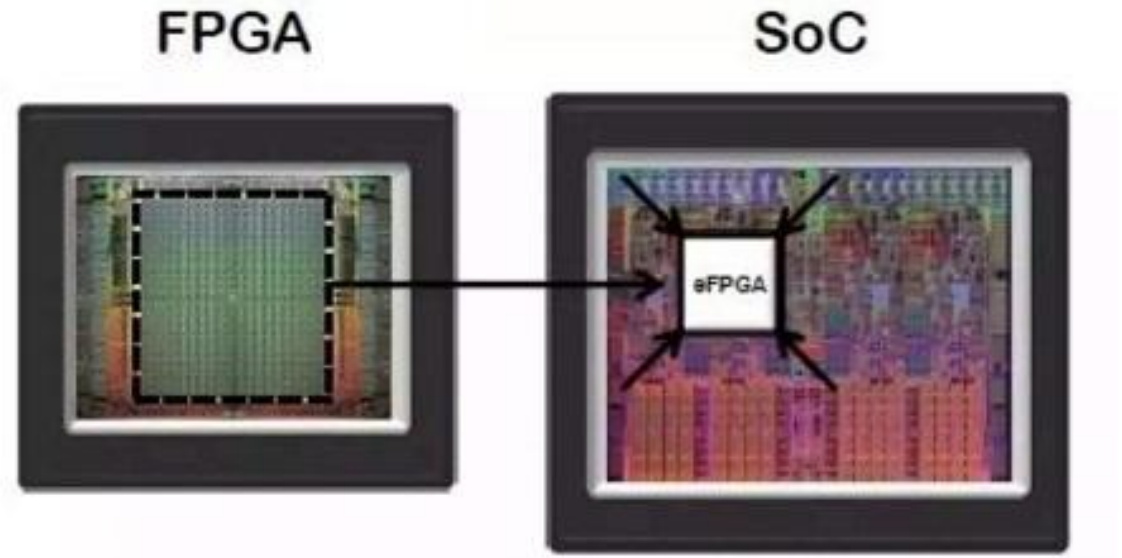
Know the "Why", need the "How" **… S3AI to support hetero-stream exploration**

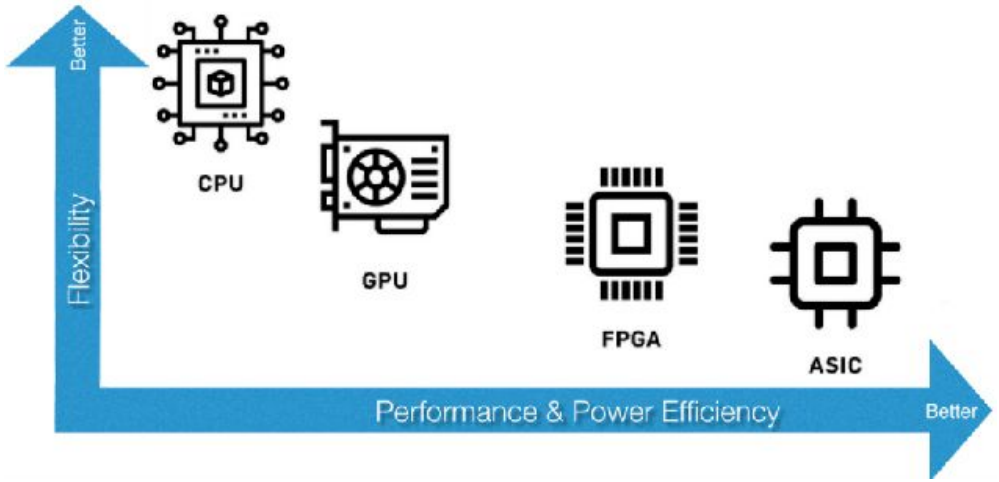


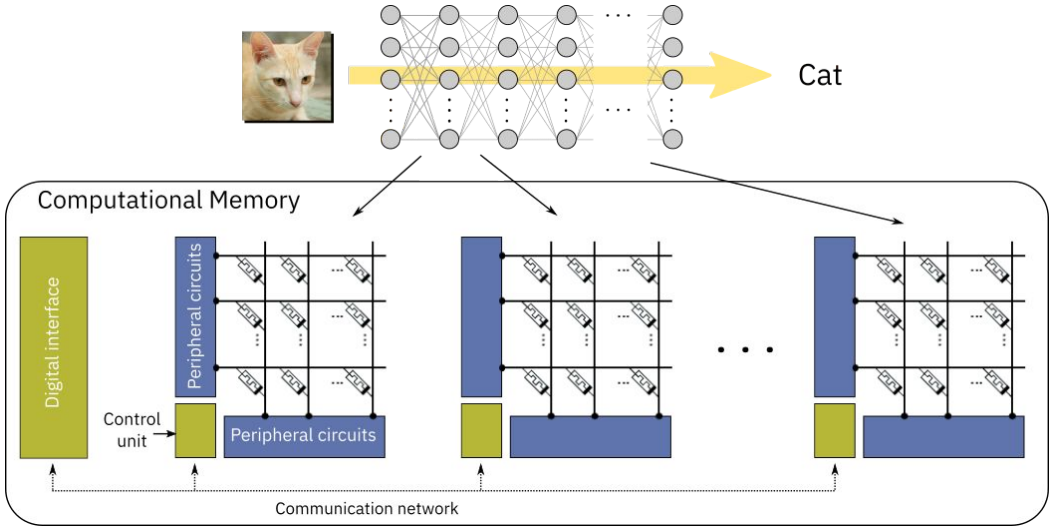
S3AI = **S**LAC **S**andbox for **S**treaming **AI**

Solving The Problem By Moving Artificial Intelligence Closer To The Sensor



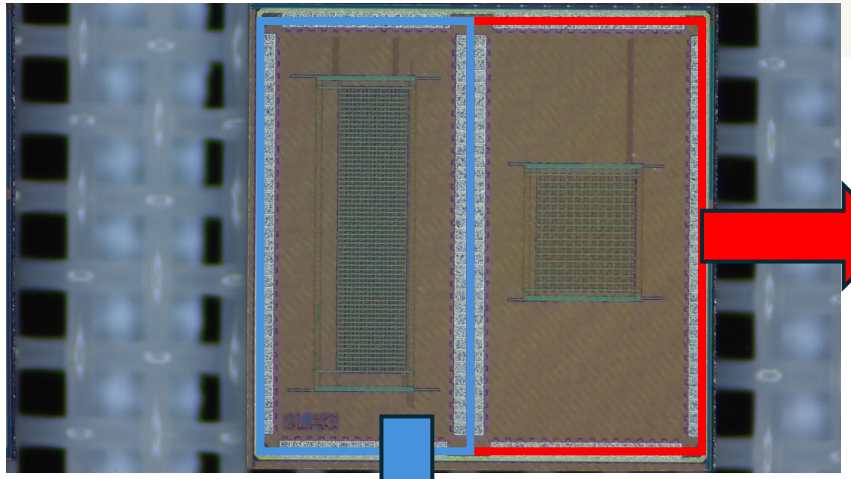




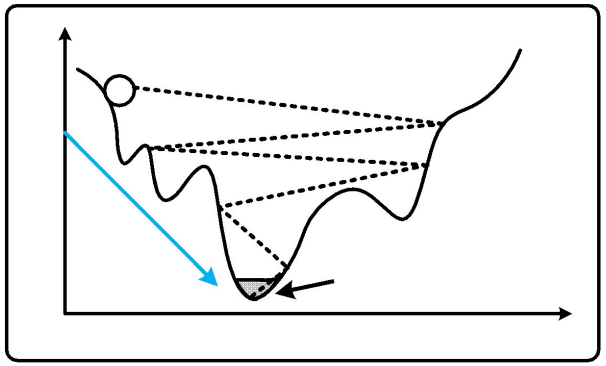


AI FPGA - DOI: 10.48550/ARXIV.2305.19455
ML/AI eFPGA - DOI: 10.1088/1748-0221/19/08/p08023

ASIC Analog In-Memory Compute & Icing Structures

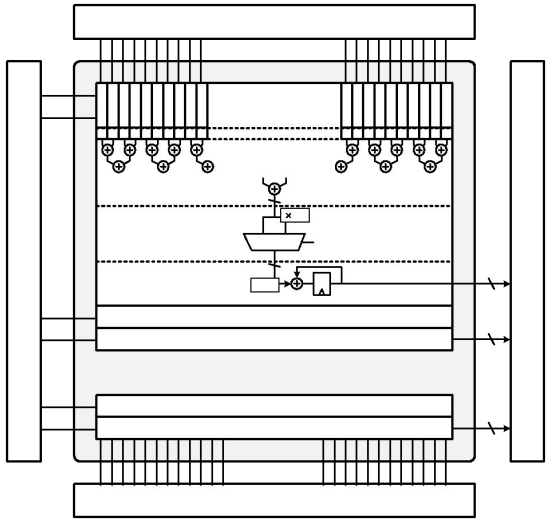


FY24-25



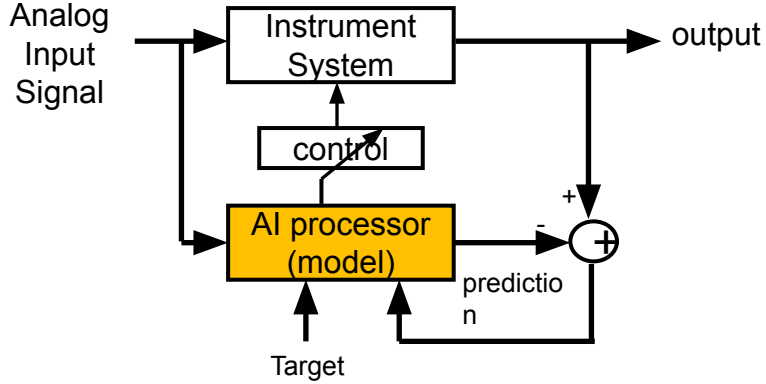
CMOS-Based Simulated Ising Machine research for solving optimization problems in high energy physics (particle tracking, jet clustering and etc) on edge.

FY24-25



SRAM-Based compute in-memory macro architecture and compiler-like design methodology research for energy/area efficient data processing in AI-ML workloads on the edge with ASIC form factor.

New for FY25



Advanced modeling, real-time optimization and autonomous operation of scientific instruments enabled by an analog AI processor at the edge

Hardware Resource Challenge

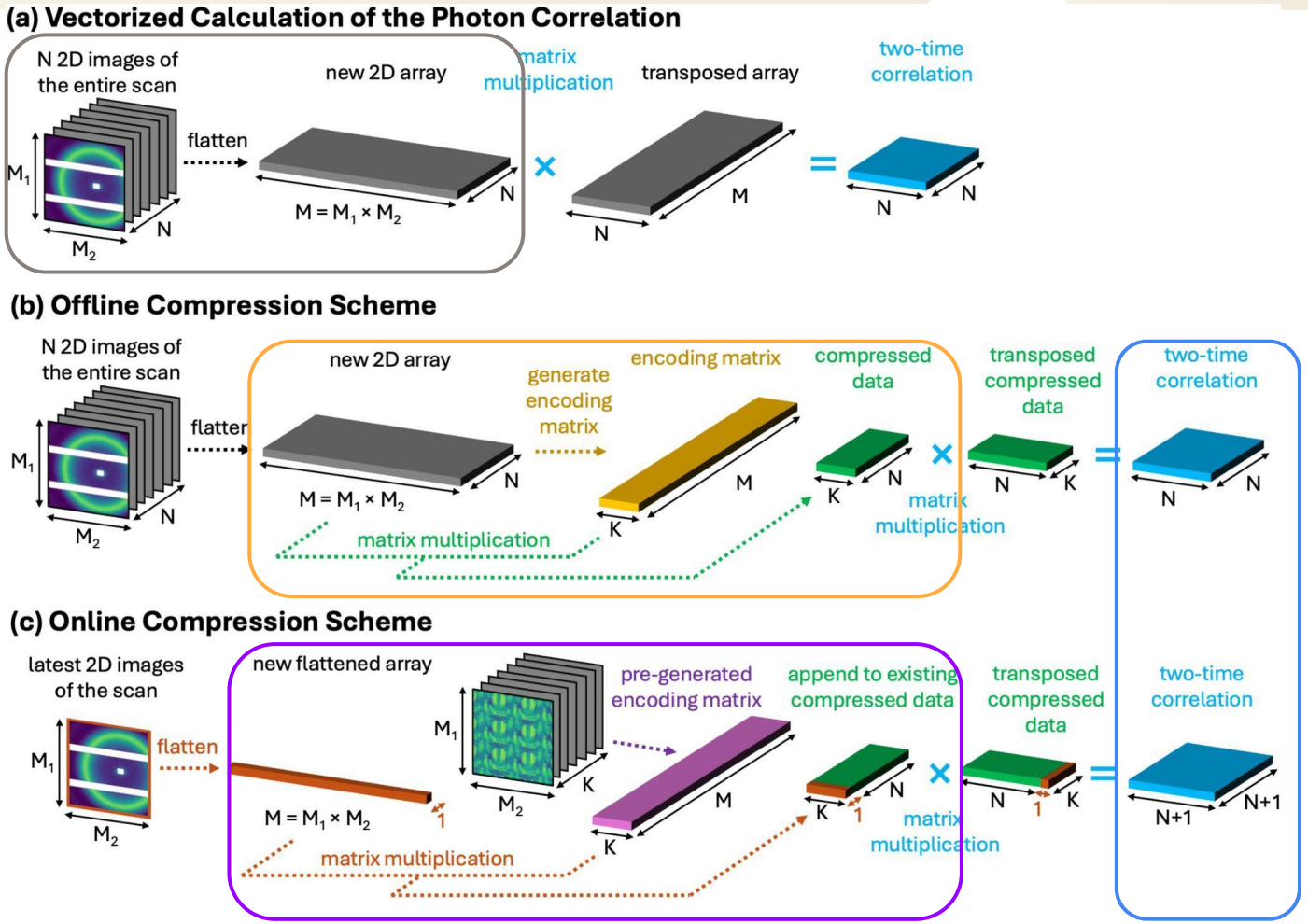
- Usually these Machine Learning models are oversized and the biggest challenge is how to fit them on to FPGA!

Domain specific vs Detector specific

Generalizabe algorithms – what users **need**, not what they **want**

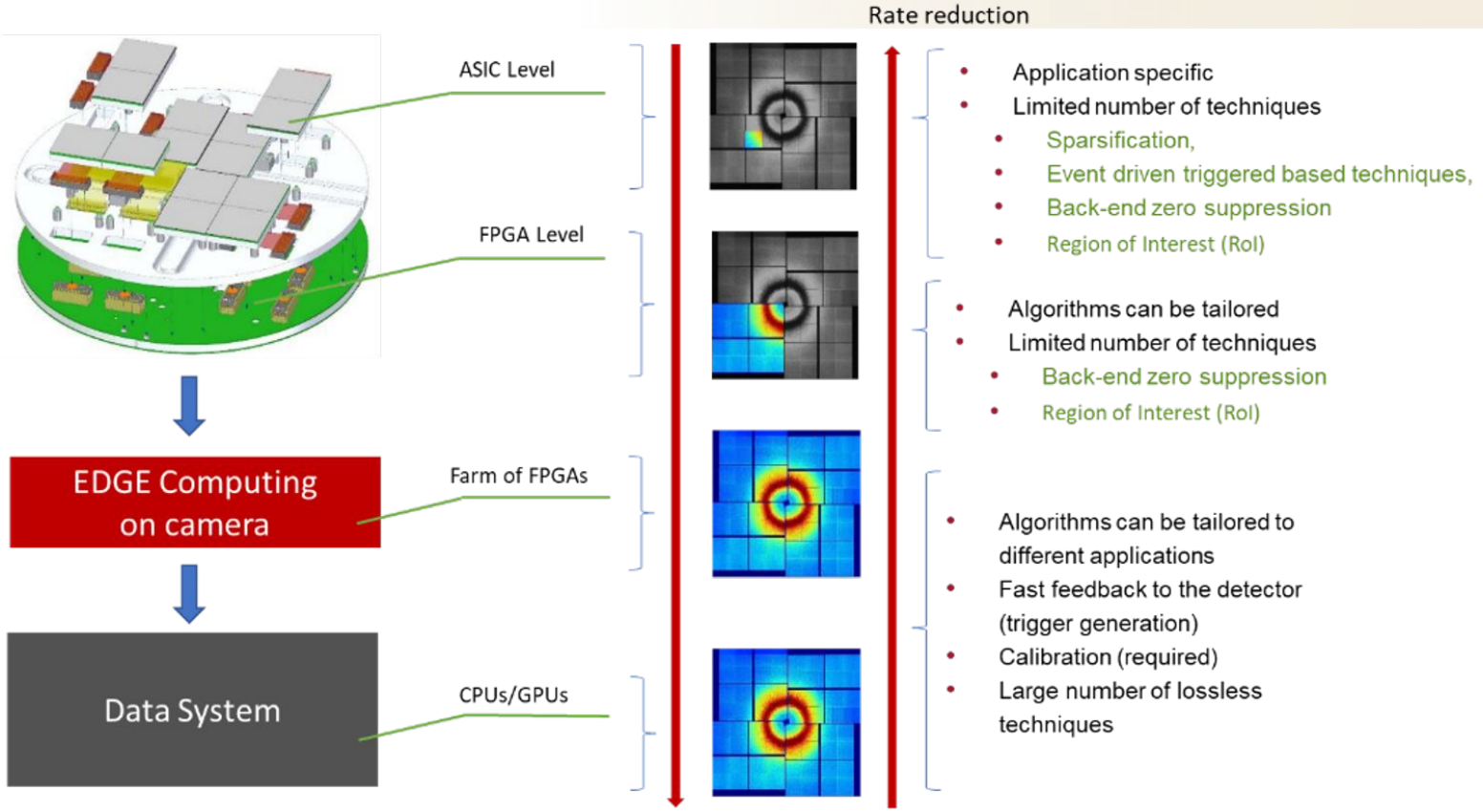
Recast offline for streaming – **offline encoding** explores **information sufficiency**

FPGA/ASIC encoding opens a new can of worms, e.g. firmware is meta-data



Strempfer *et al.*, "Homomorphic data compression for real time photon correlation analysis," Opt. Express 33, 12059-12070 (2025)

AUREIS Overview - Moving Processing Into The Detector

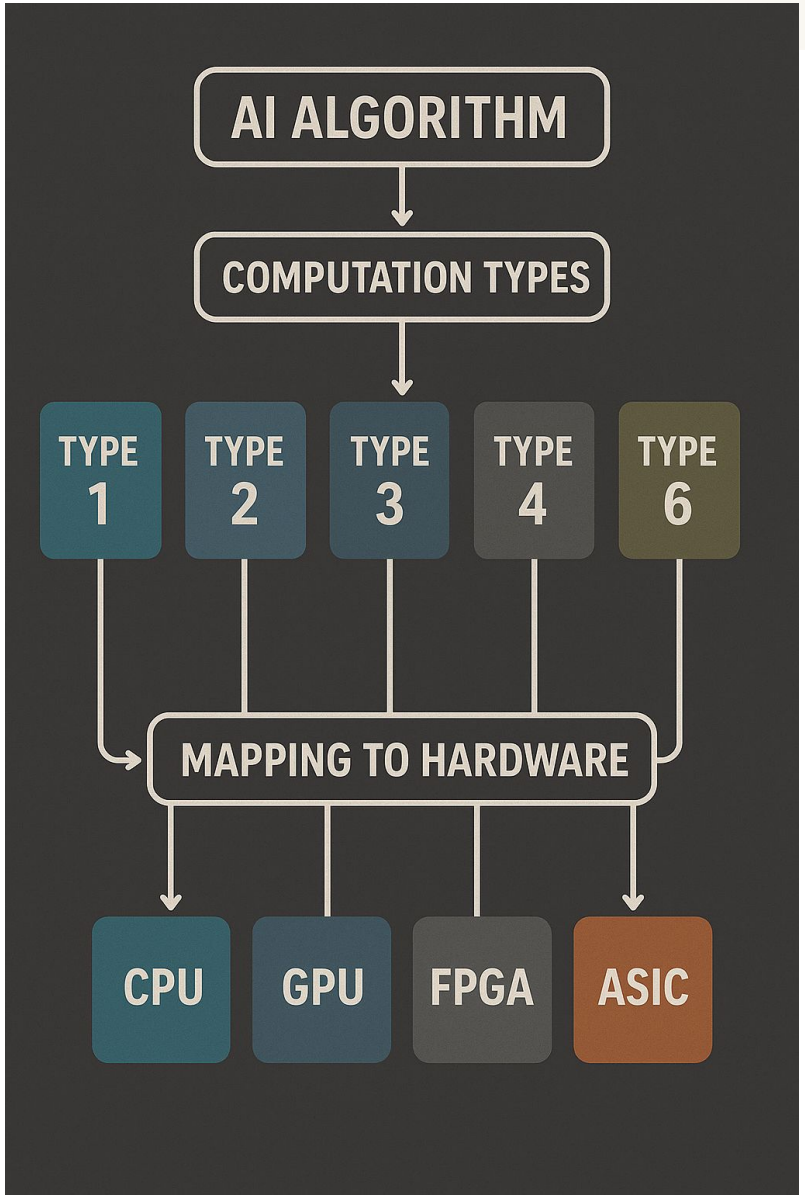


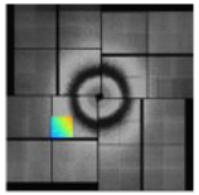
MEERCAT - Microelectronics Energy Efficiency Research Center for Advanced Technologies

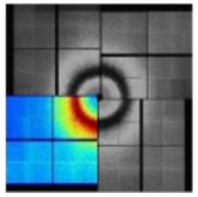
AUREIS aims to develop the underpinning microelectronics technologies for ultrafast, energy-efficient, dense networks of sensors, large-area imagers, and future detectors, solving the extreme data deluge problem by:

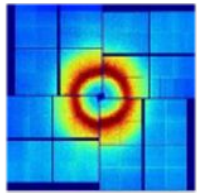
- Reimagining network of sensors as data-driven adaptive intelligent architectures
- Introducing optimally distributed computing at the data source (edge)
- Leveraging meshed hierarchical interconnections
- Optimizing and benchmarking energy and information extraction efficiency

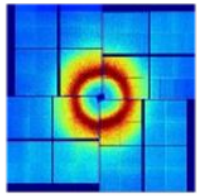
AUREIS Overview - Algorithm Mapping & Interconnects











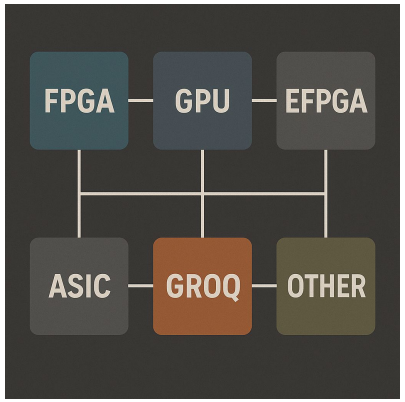
Thrust 1 – Hardware/resource-aware ML/AI-based workflows for dynamic real-time experiment operation.

Thrust 2 - Energy-efficient distributed network organization and AI-based edge computing architecture trainable across architectures -- analog and digital -- for optimized information extraction.

Thrust 3 - Ultra-high-rate trainable front-end ASIC architectures with adaptive analog interfaces.

Thrust 4 - Integration of dense multi-tile sensor planes with inter-tile communication.

Thrust 5 - Wide bandgap material-based sensor for efficiency over a wide range of energies.

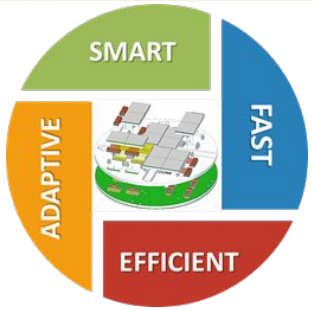


Project Morpheus: what do we need to study to achieve adaptive detectors?

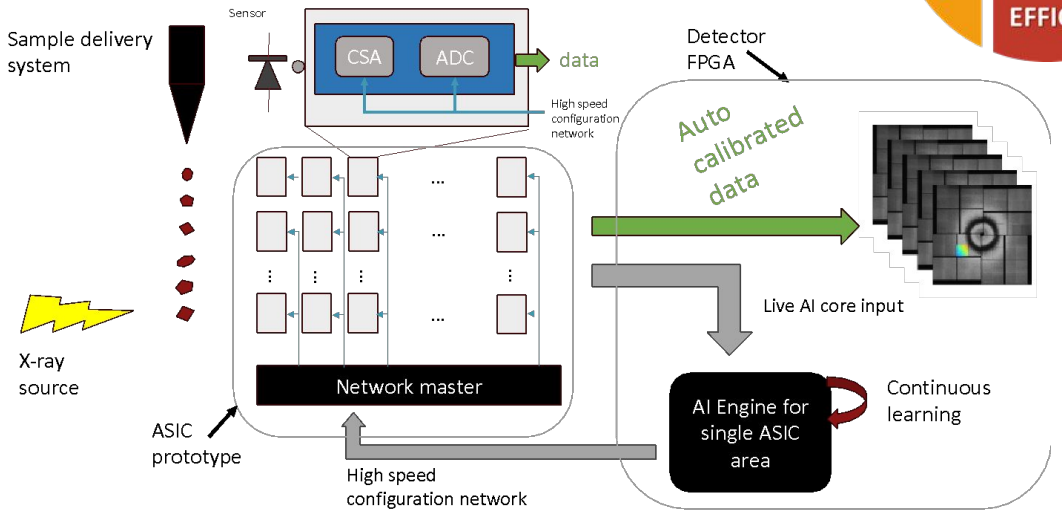
- High speed detector create a data deluge at every detector level (ASIC, in detector FPGA, support processing FPGA/CPU/GPU farms, super computers

- **Adaptive** resolution (spatial, energy and time) for real-time performance optimization
- **Auto-calibration** (electrical units -> physical units)
- **Trainable** to extract information in real-time (independent of the occupancy)
  - E.g., focused readout (on recognized features) based on neuromorphic architectures
  - E.g.. feedback loop with a training ML algorithm on HPC in the back reconfiguring AI inference in the detector front end
- Producing Data at the **Information** rate
  - Validated algorithms – no loss of information
- **Ultra-high frame rate** (programmable patterns)
- Energy **efficient**

Lots of questions, some ideas, few answers and limited possibility to explore these directions



Custom ASIC to enable fast re-configuration and AI core to enable adaptive detector

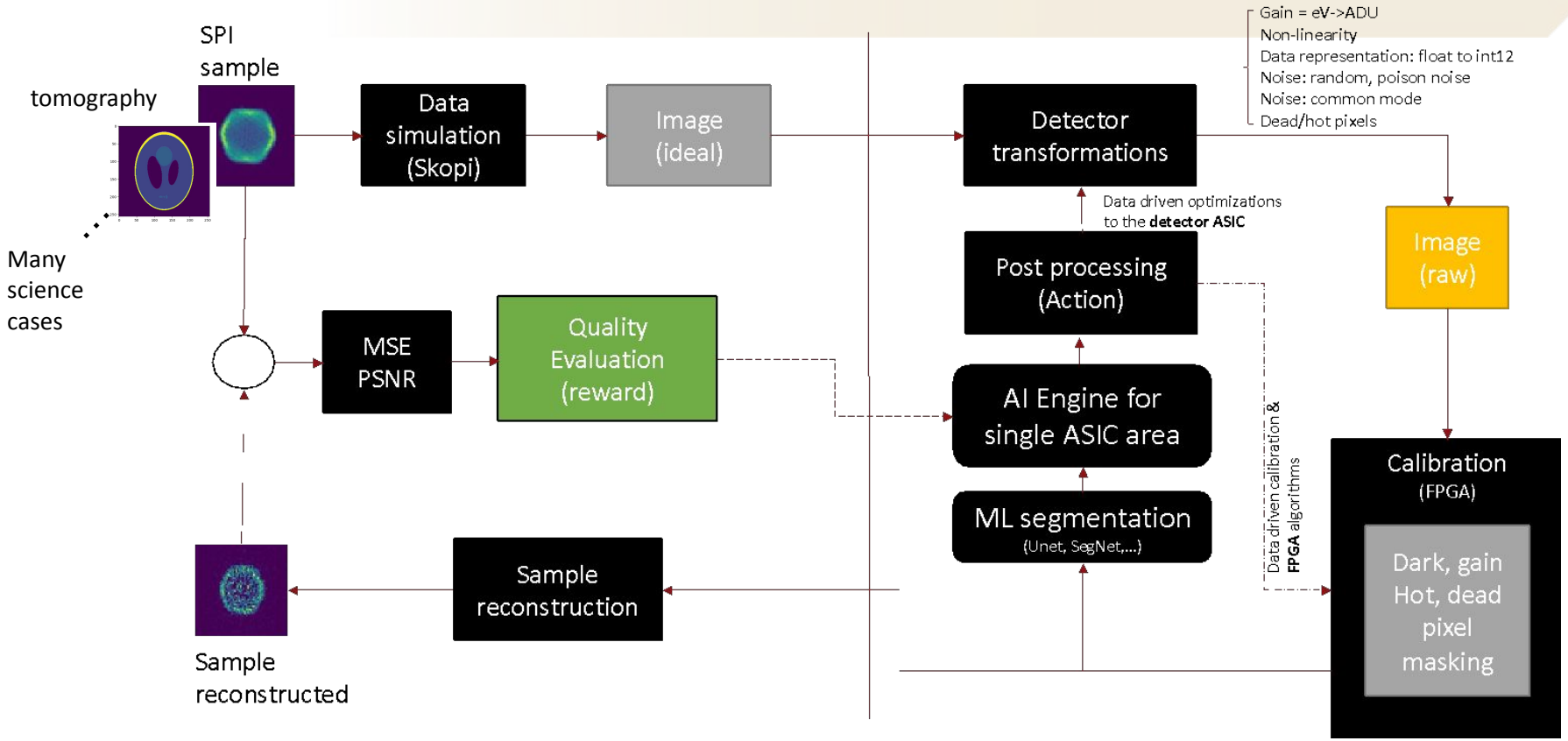


A Multi-Disciplinary Co-Design Foundational Research Problem

Need to leverage broad expertise

ANL and SLAC collaboration
PI: Dr. Angelo Dragone
Co-PIs: Dr. Antonino Miceli and Dr. Dionisio Doering

Science in the loop: Reinforcement learning and Morpheus



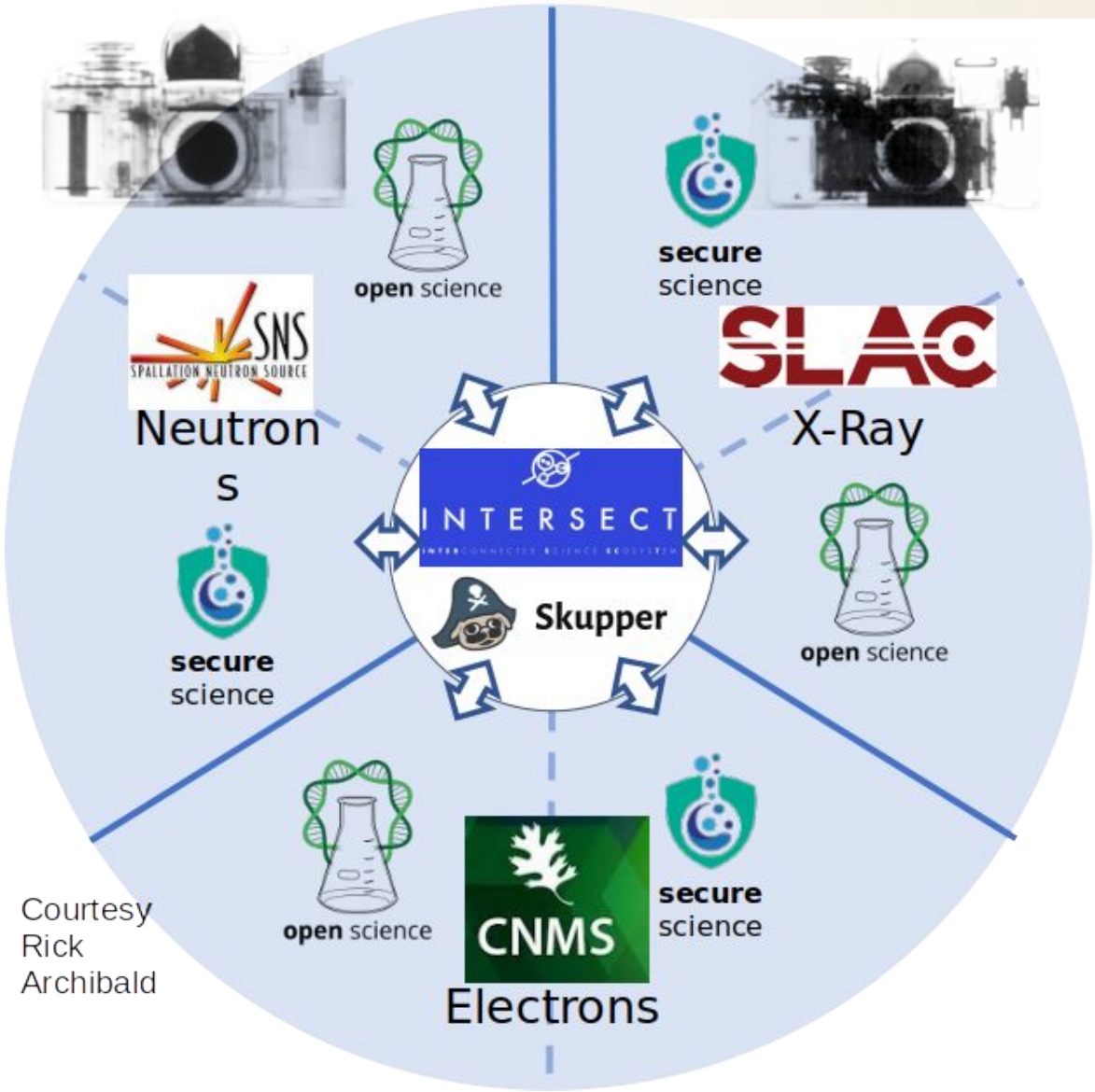
Science specific domain

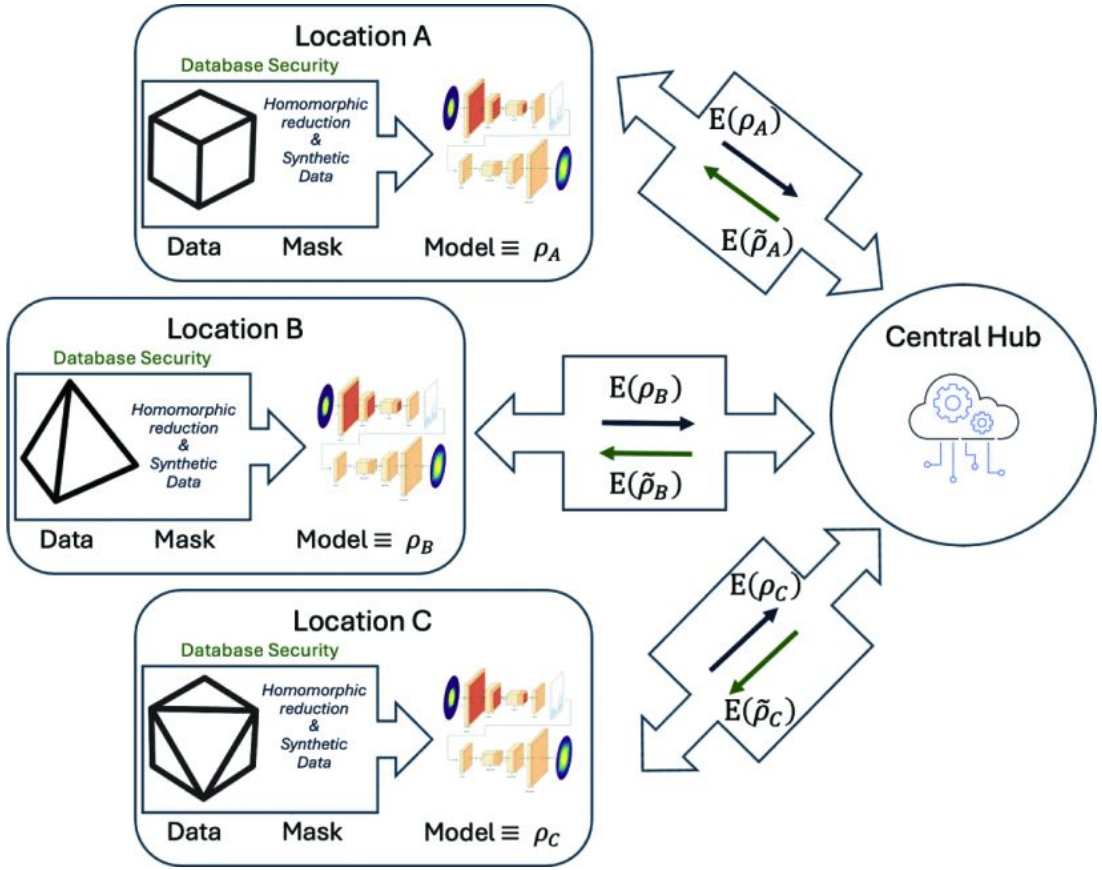
Adaptive detector

Simulations enable the study to can knowledge of the ground truth

Framework enables the detector to be trained and adapt to the domain

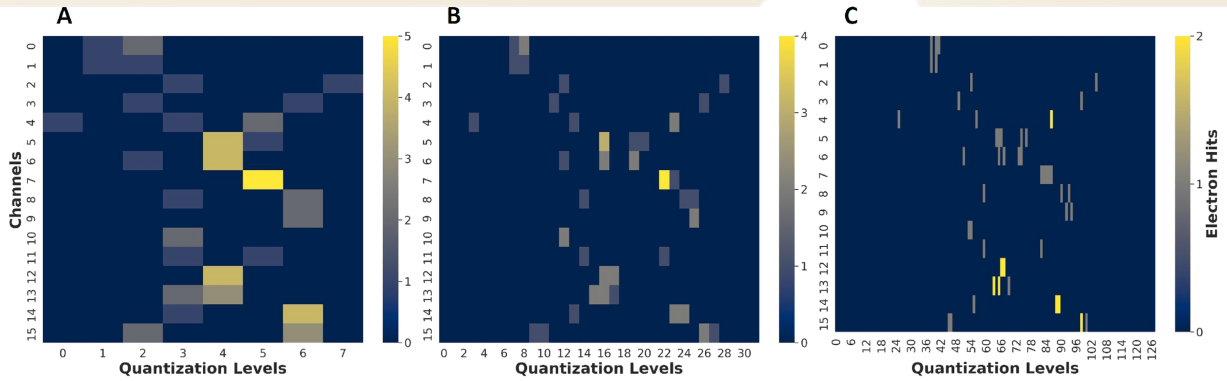
Multi-Model & Distributed Data Sources

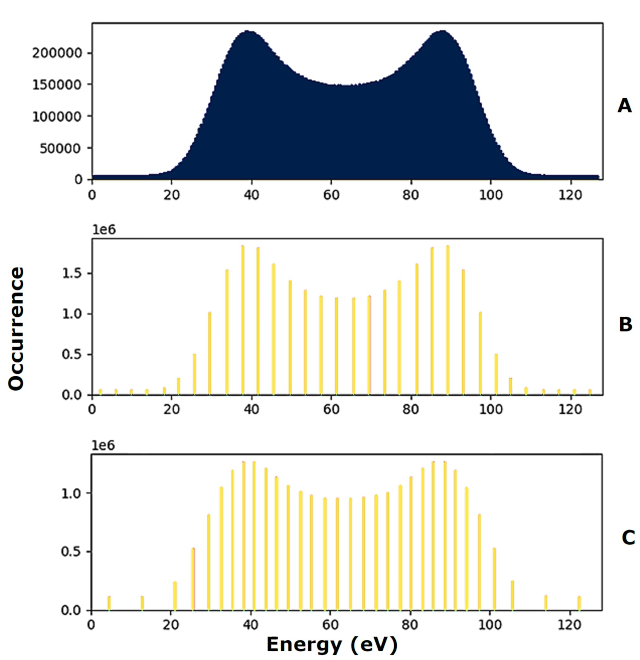


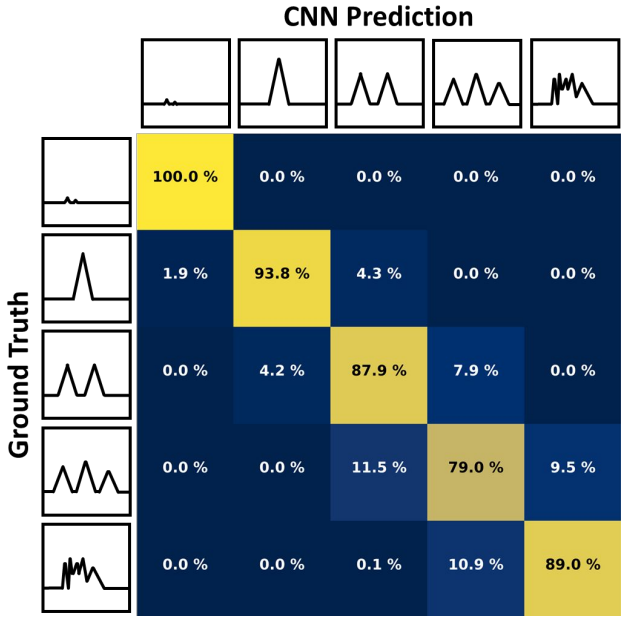


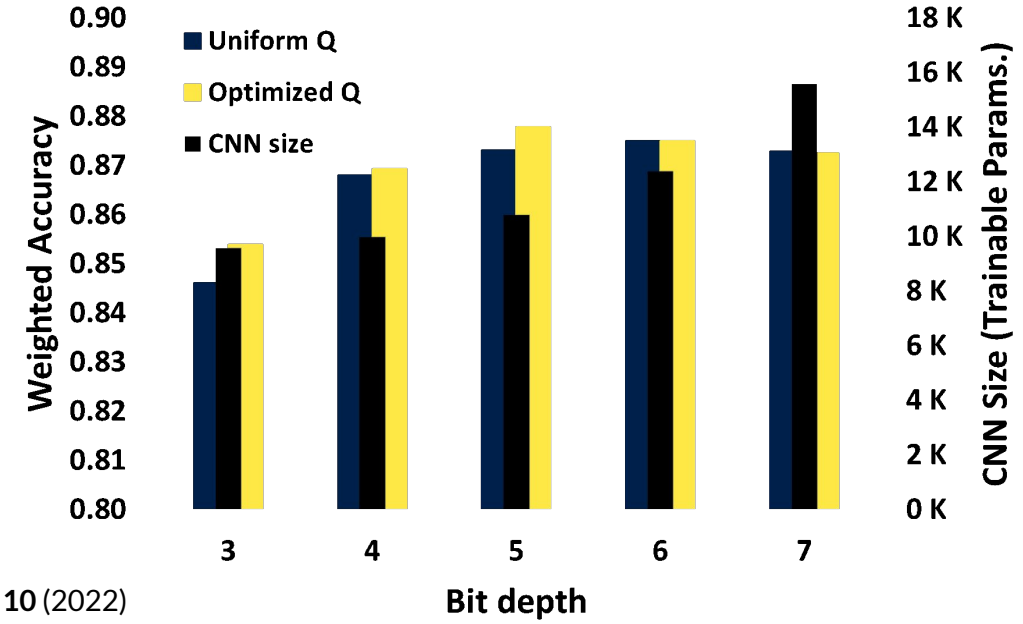
Quantization is Unavoidable – Variable binning

- Optimizes information per bit
- Reduces input token dimensionality
- Puts metadata to work





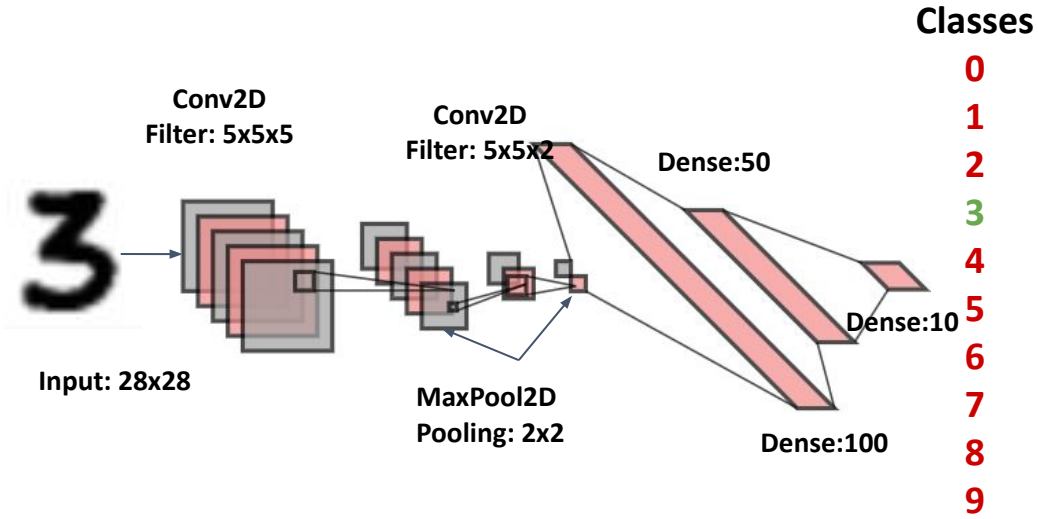




Gouin-Ferland, Coffee and Therrien, Front. Phys. **10** (2022)

Quantization Challenges

Layer by Layer Quantization to make ML model Lighter



Tiny LeNet for Digit Classification Parameters: 9242
Weights and biases: 32bit (High Precision) **95.07% Acc.**
Post Training Layer Wise Quantization **89.21% Acc.**

Quantization in ML reduces model precision to make it more efficient for deployment on resource-constrained hardware like FPGAs, GPUs, and other devices.

Digital vs. Analog Hardware: Quantization techniques and scaling methods (e.g., in QKeras for FPGAs) might not work well for analog hardware due to different characteristics.

Hardware-Specific Formats: Tools like QKeras use fixed-point formats (e.g., ap_fixed) suitable for FPGAs, while GPUs often use different formats like INT8 or FP16.

Different Tools, Different Approaches: Quantization methods vary by hardware and toolchain, impacting how models are scaled and represented.

Training tools & hardware must match!
Ideally the tool properly mimics the real hardware!

There are two types of Quantization:

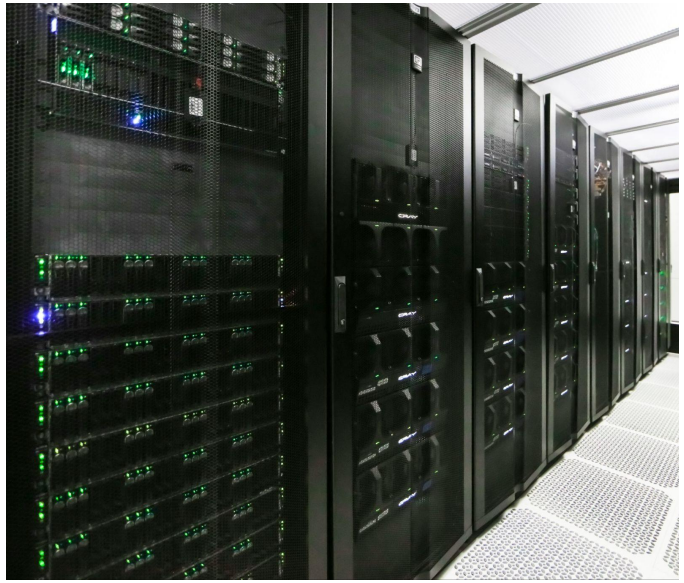
1. Quantization Aware Training (QAT): Quantize weights and biases during training
2. Post Training Quantization (PQT): Quantize weights and biases after training

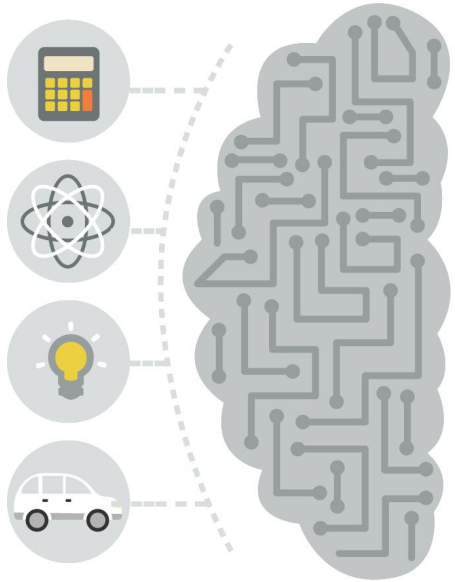
Edge-to-Exascale and Back!

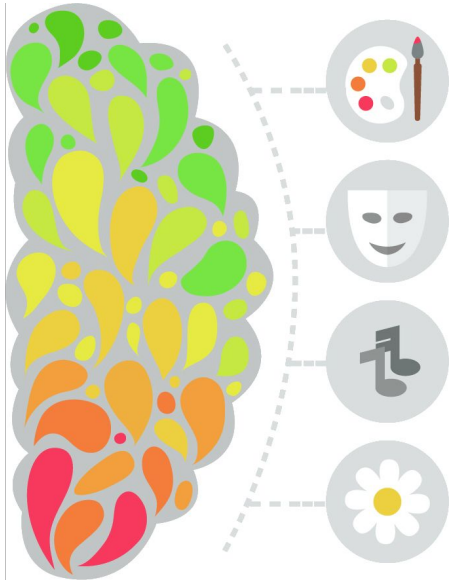
HPC testbeds linked to Edge Streaming Sensors and Early Access Hardware

- Testbeds that design for **Edge Integration with LCF**
- Real-world streaming tests to work out bugs and security
- Prototype domestic inter-lab federation, then international
- IRI Orchestration should align with future HEP international ecosystem

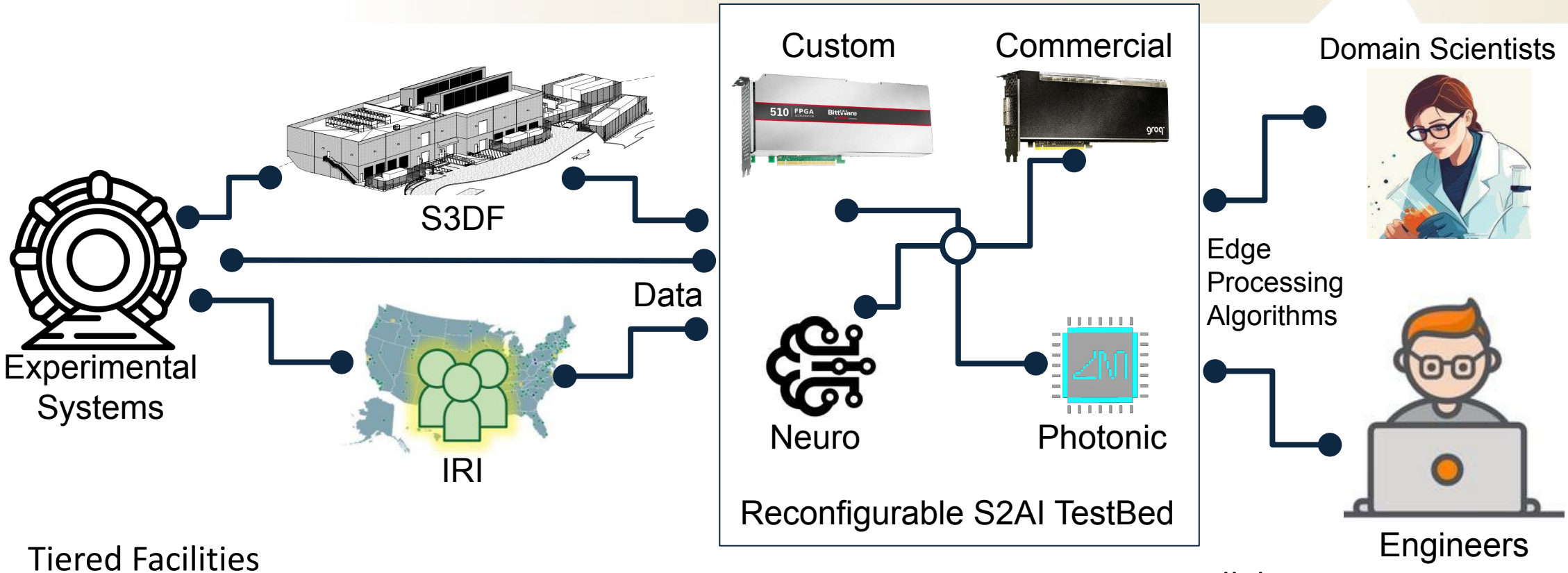
- Reconfigurable hardware and racks for **design exploration**
- Streaming imaging **(photonics)** and digitizers **(analog)**
- Early access for **inference hardware** and **custom ASICs** and HEP sensor prototypes
- Long DOE history in FPGA and leading **eFPGA** into age of chiplets for trigger, stream, and control systems







S3AI TestBed Integration With S3DF and IRI



- Tiered Facilities
  - Experimental sensors
  - Mid-scale HPC – also archival storage
  - LCF

- Community Collaboration
  - Workforce Development
  - Open the hood on weird hardware
  - HEP science drives global technology mission

Conclusion And A Plug: SLAC Neural Network Library (SNL)

Key points:

- Provides specialized set of libraries for deploying ML inference to FPGA, eFPGA & ASIC
- Using High-Level-Synthesis (HSL): C++ programming of FPGA/ASIC
- Supports Keras like API for layer definition
- **Dynamic reloading** of **weights** and **biases** to avoid re-synthesis
- Supports 10s of thousands of parameters or more depending on latency requirements for the inference model
- Total end to end latency of couple of **usec** to couple of **millisecond**.
- Streaming interface between layers.
- Allow for **pipeline** of the **data flow** for a **balance** of **latency vs frame rate**
- Library approach allows for user/application specific enhancements