Toward Hardware-Aware ML Inference: Exploring Inference Across Different Hardware Platforms

- Sharing implemented ML inference and exploring scalable hardware-aware futures.

Abhilasha Dave

AUREIS Mini-Workshop on Data-Flows for Intelligent Sensing

June 27th , 2025

SLAC NATIONAL ACCELERATOR LABORATORY

Stanford University | U.S. DEPARTMENT OF ENERGY

Motivation: Why Real-Time ML Inference Matters

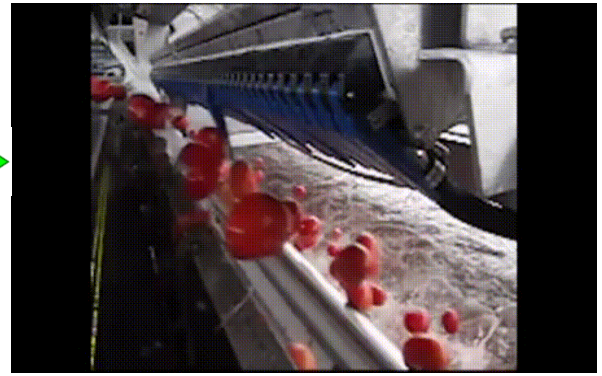
Problem: Detectors now produce more data than we can store or transfer

LCLS-I:
Generating & storing data at GB/sec

LCLS-II:
Generating data at 1TB/sec

LHC:
1 Petabyte/sec

Manual or offline filtering of data



Real-time filtering with ML

- Traditional **"store first, analyze later"** methods no longer work
- Need **real-time, intelligent processing** at or near the detector
- **ML enables data filtering, insight extraction, and classification (hit/miss)**
- Processing must happen **as close to the sensor as possible**

Making the Detector Pipeline Smarter with ML Inference

🎯 **Goal: Not replacing the current detector pipeline — but *enhancing* it with smarter decision-making.**

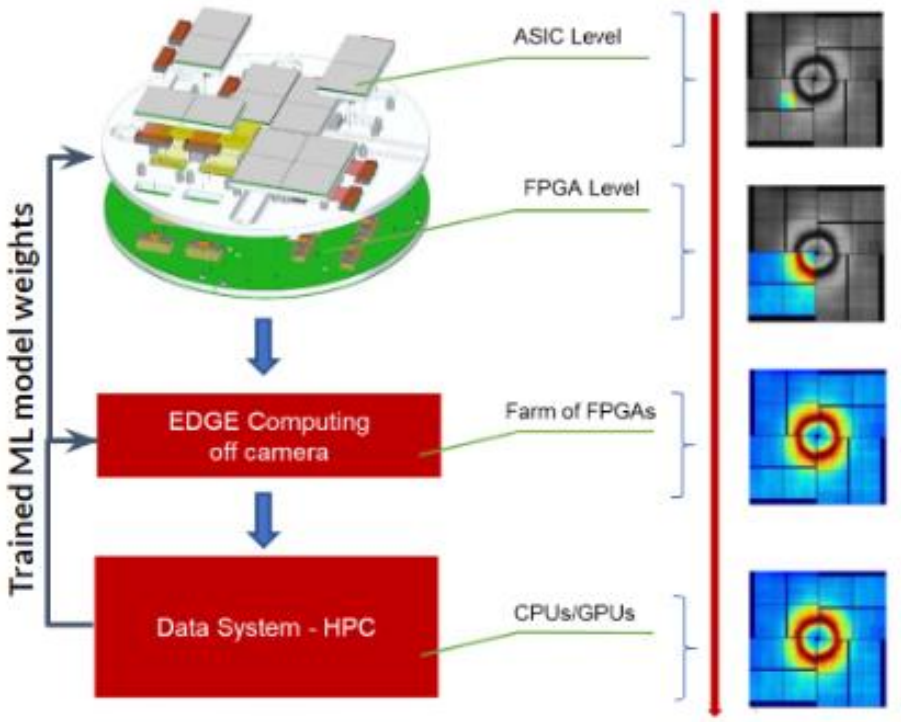
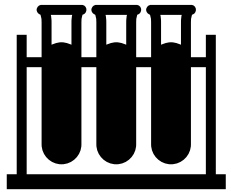


Image 1: Real-time Information Extraction and Detectors

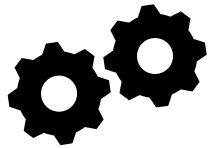
- ML inference can be introduced at multiple levels in the pipeline
  - ASIC Level:
  - FPGA Layer
  - HPC/Cluster (CPU/GPUs, TPUs , Groq, etc.)
- Each of these layers has **different constraints:**
  - Compute
  - Power
  - Latency
  - Bandwidth
  - Memory
- Instead of trying to run every ML task everywhere, we ask: **What task fits best on what hardware?“**
- **Some layers might be perfect for an FPGA, others need GPUs/CPUs/TPUs/LPUs.**
  - The key is maybe to designing an inference flow that’s distributed and optimized — not standardized to a single box

Profiling ML Architectures Across Diverse Hardware Platforms

- **Objective**: Understand which ML architecture components perform best on which hardware
  - To guide future hardware-aware model design in real-time detector pipelines.



- The ML architecture testing:
  - MLP (fully connected/dense layers)
  - CNN (convolutional layers)
  - MLP-Mixer (Transformer-style block)

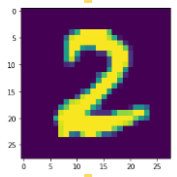


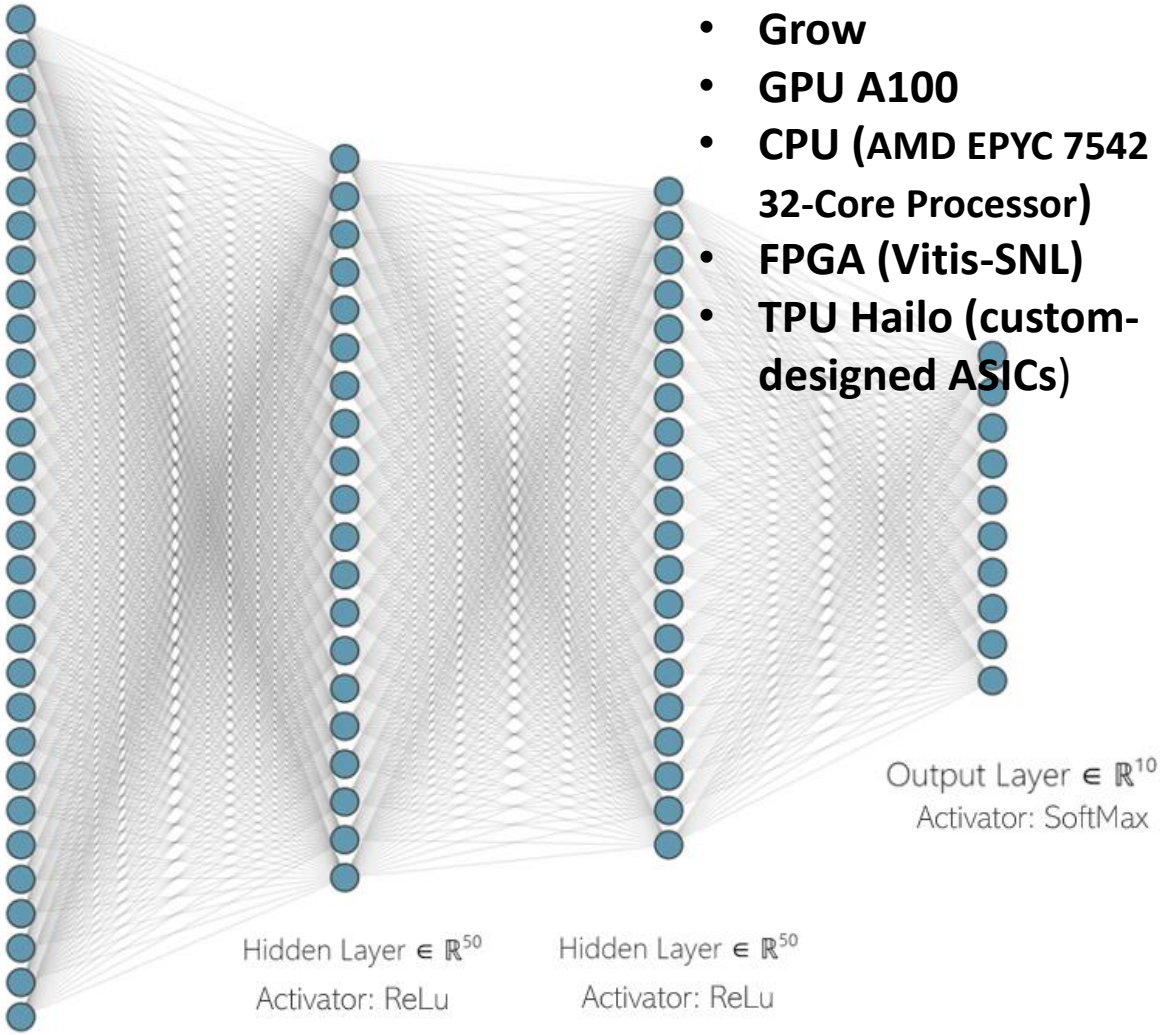
- The Hardware Platforms to test on:
  - CPU (AMD EPYC 7542)
  - GPU (NVIDIA A100)
  - FPGA (SNL with Vitis)
  - TPU Hailo Board (custom-designed ASICs)
  - Groq

- Methodology (progressing):
  - Run the full models inference on each platform
  - Measure inference latency for different batch sizes
  - Observe which **layers run fastest** were.
  - Will be including in future the power of each device for different architecture

- **Insight Gained:** Even when running whole models, we can infer which layers (e.g., MLP vs. conv vs. mix) are hardware-optimized, guiding future partitioning strategies.

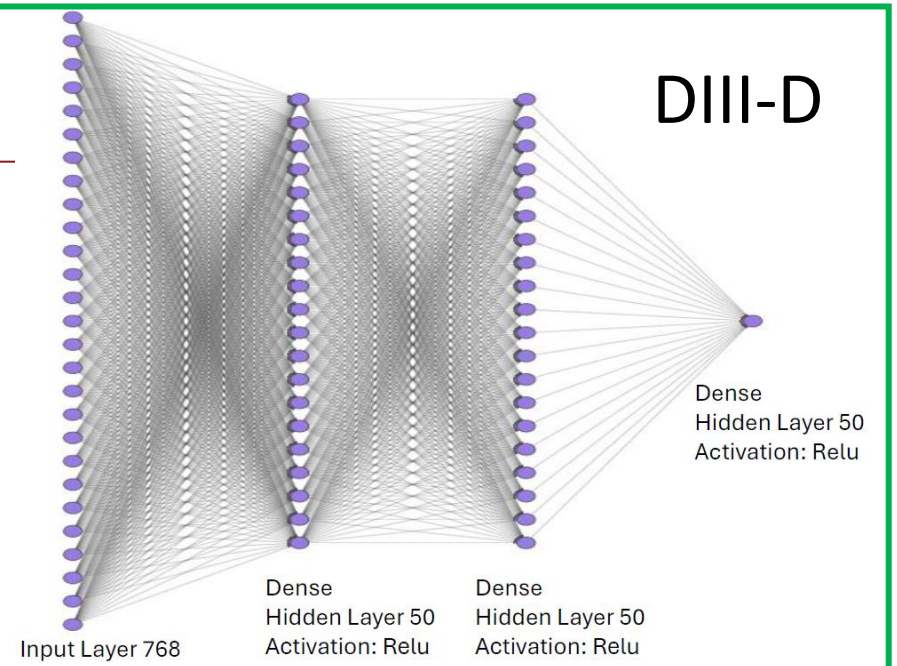
MLP NN on MNIST DataSet

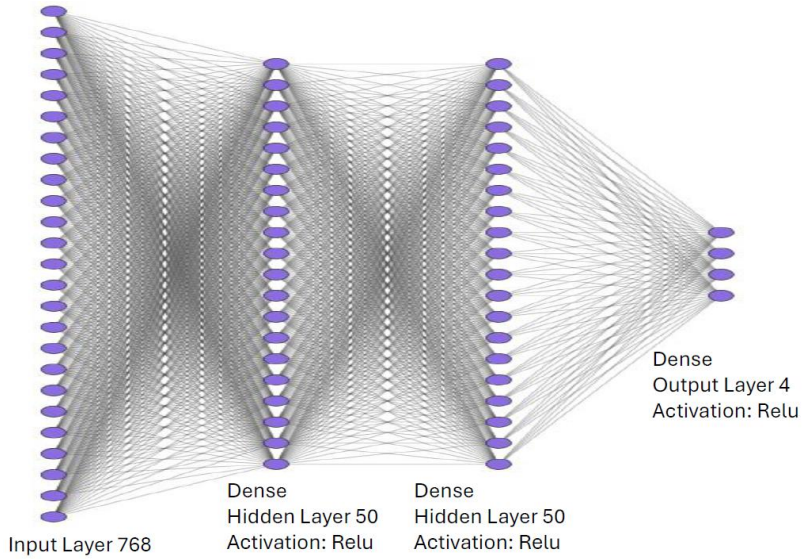




- **Grow**
- **GPU A100**
- **CPU (AMD EPYC 7542 32-Core Processor)**
- **FPGA (Vitis-SNL)**
- **TPU Hailo (custom-designed ASICs)**

DIII-D

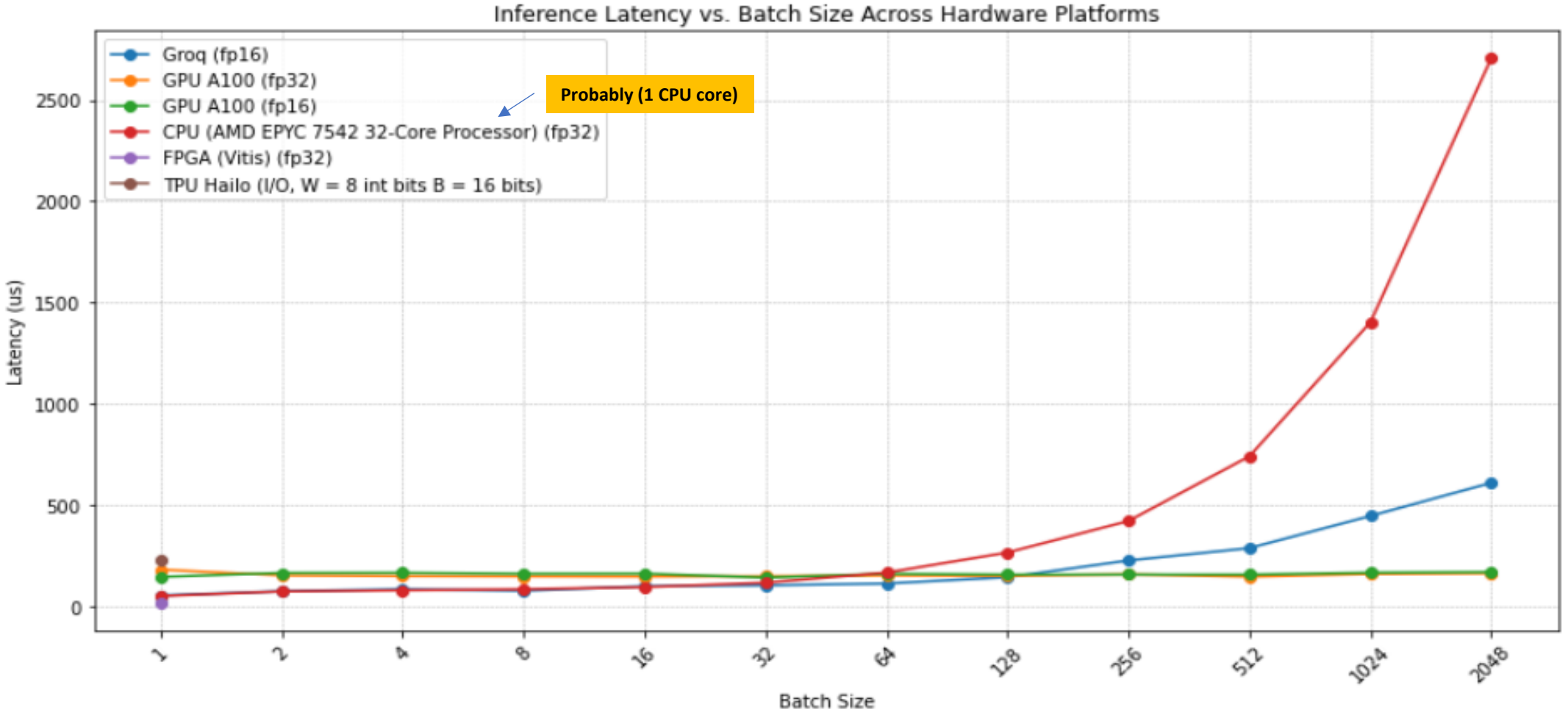




*Unpublished work
Paper will come on arxiv soon
*Ryan Coffee - LCLS SRD AMO Science

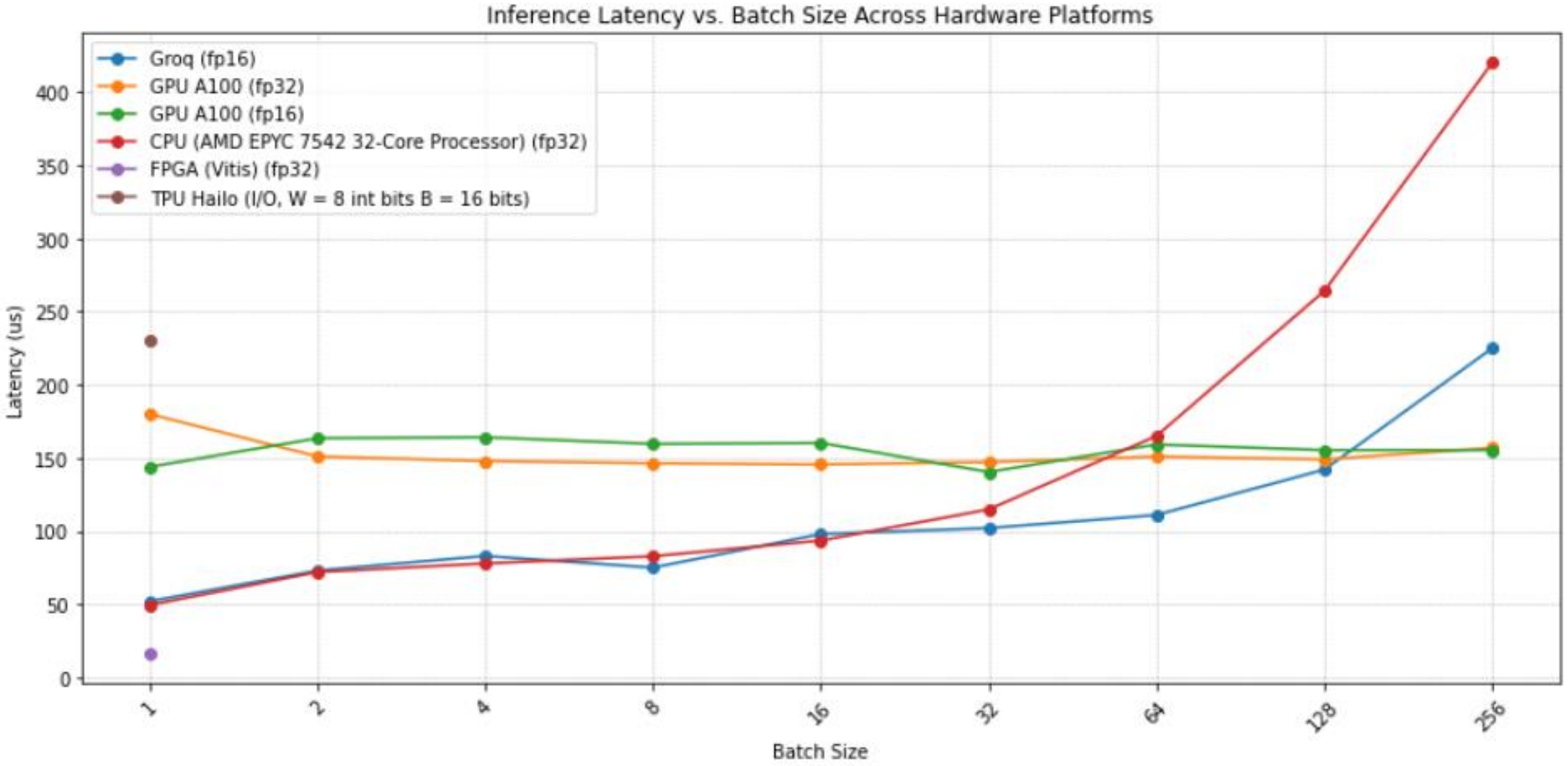
MLP NN on MNIST Dataset

Performance Analysis on **Groq, GPU A100, CPU (AMD EPYC 7542 32-Core Processor), FPGA (Vitis-SNL), TPU Hailo**



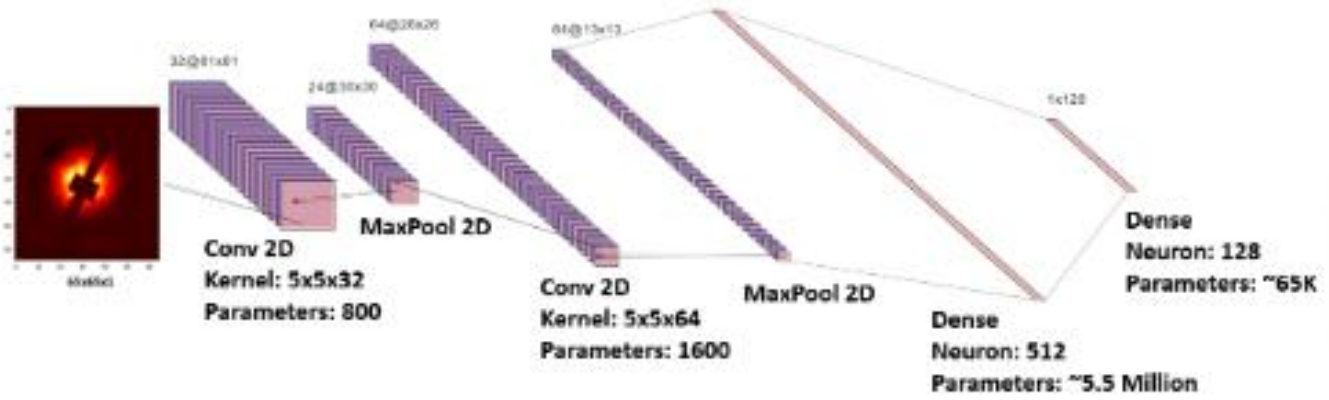
*TPU Hailo results collected by Fengrui Zuo

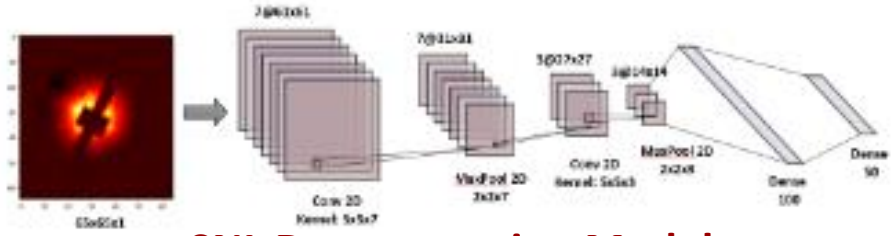
MLP NN on MNIST Dataset



*TPU Hailo results collected by Fengrui Zuo

SpeckleNN (CNN Style Model)





SNL Demonstration Model

* https://www.frontiersin.org/journals/high-performance-computing/articles/10.3389/fhpcp.2025.1520151/full

~99% Model size reduction

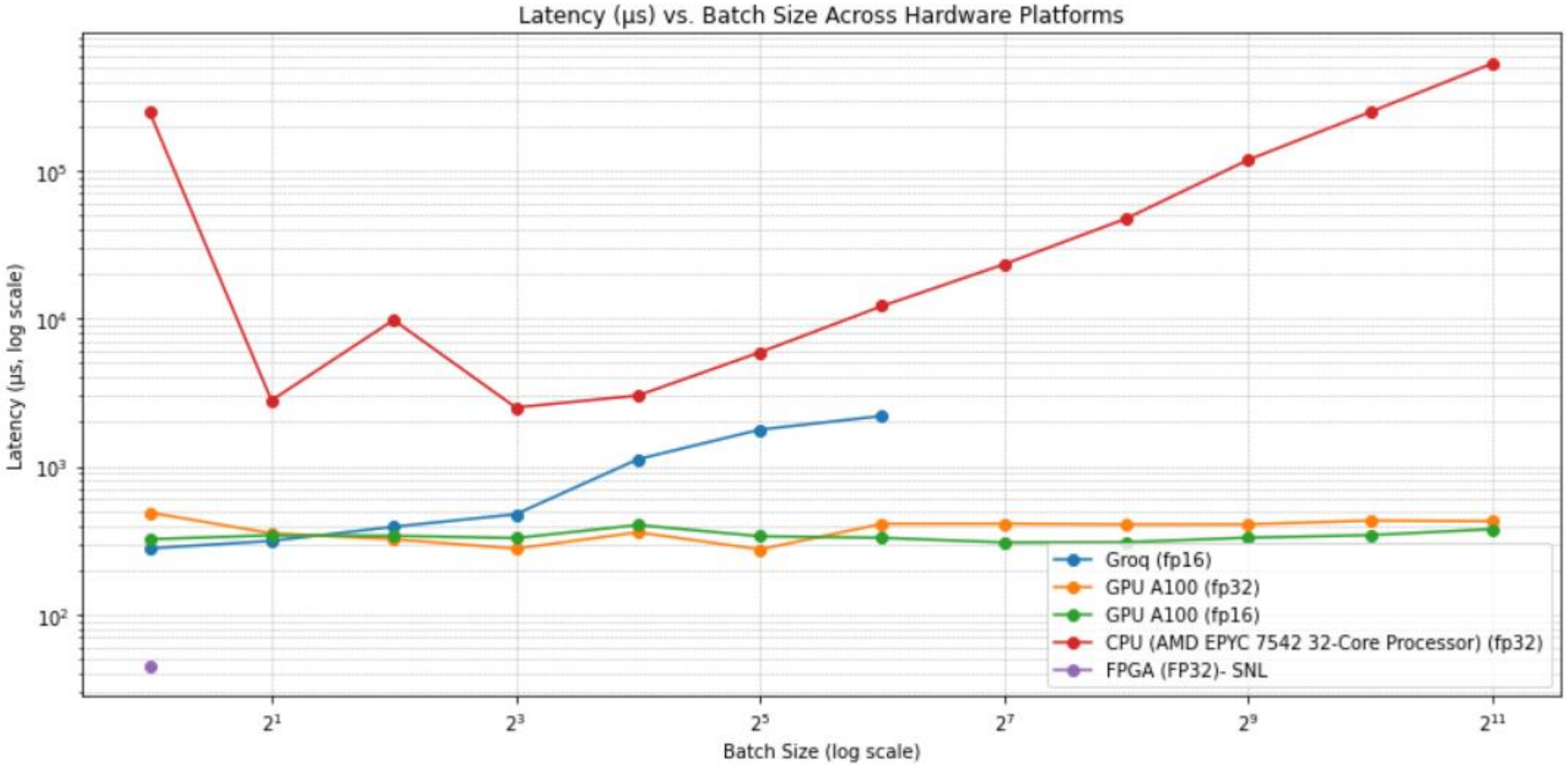
- Total Parameters:~**5.6Million**
- Data Volume Compression: **98%**
- Classification Accuracy: **94%**

- Total Parameters:~**56K**
- Data Volume Compression: **98.8%**
- Classification Accuracy: **~ 91%**

*Cong Wang - LCLS

SpeckleNN – LCLS-II (Smaller version)

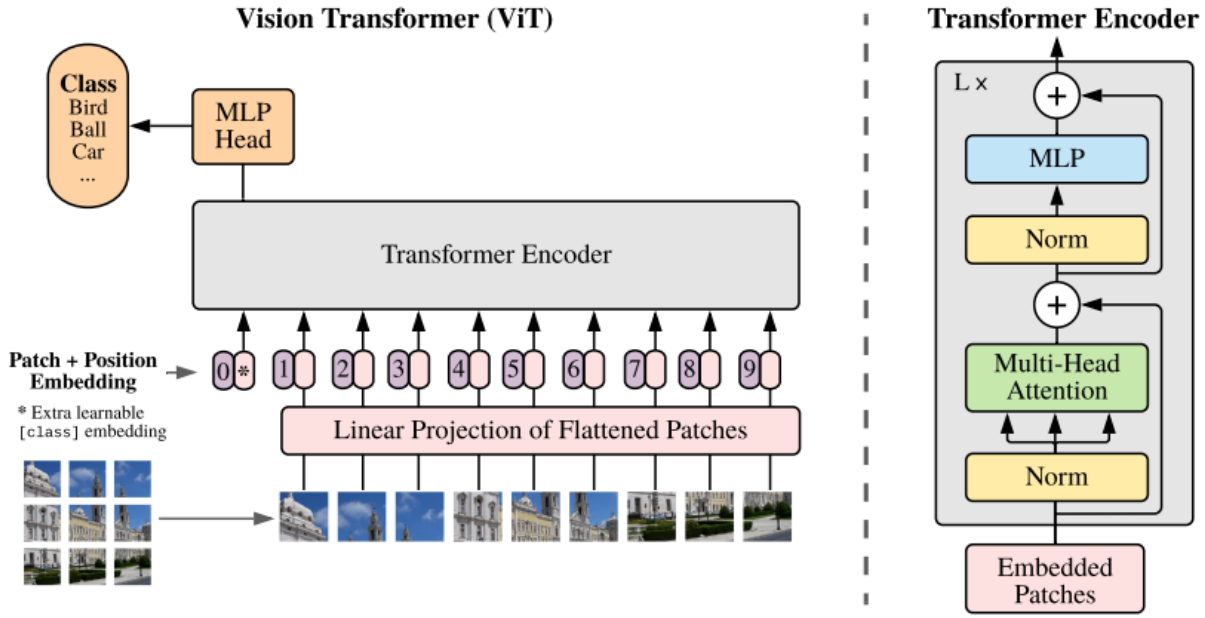
CNN Style model



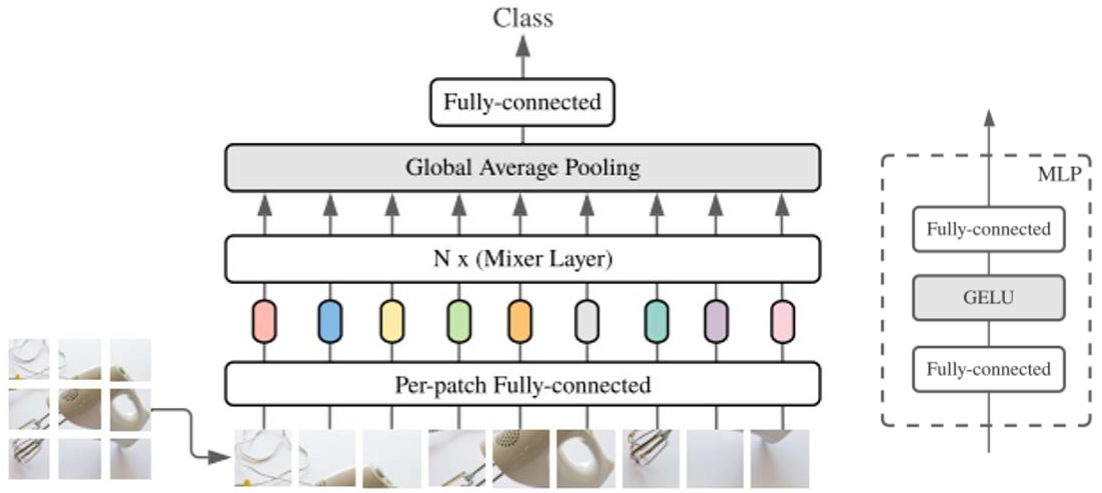
ATLAS LAr Calorimeter (LHC)

Problem: certain kinds of beyond the Standard Model physics can be detected only through cell-level analysis of calorimeter showers

- Dataset developed by Columbia University ATLAS group studying the decay of light axion-like particles into overlapping LAr showers with "substructure"



Trasformer model is implemented by Columbia University ATLAS group

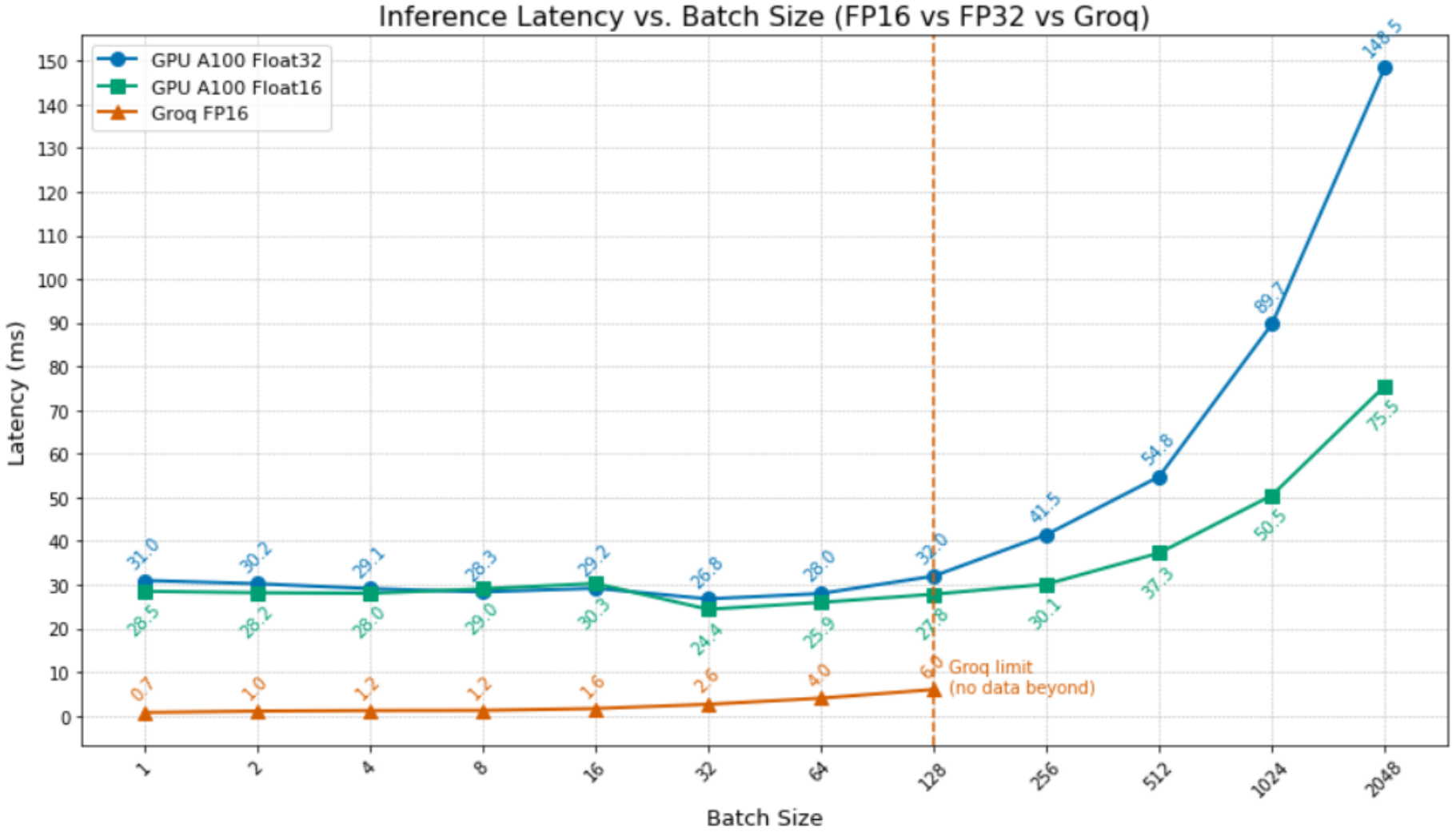


Equivalent MLP-Mixer model is implemented in-house SLAC-TID

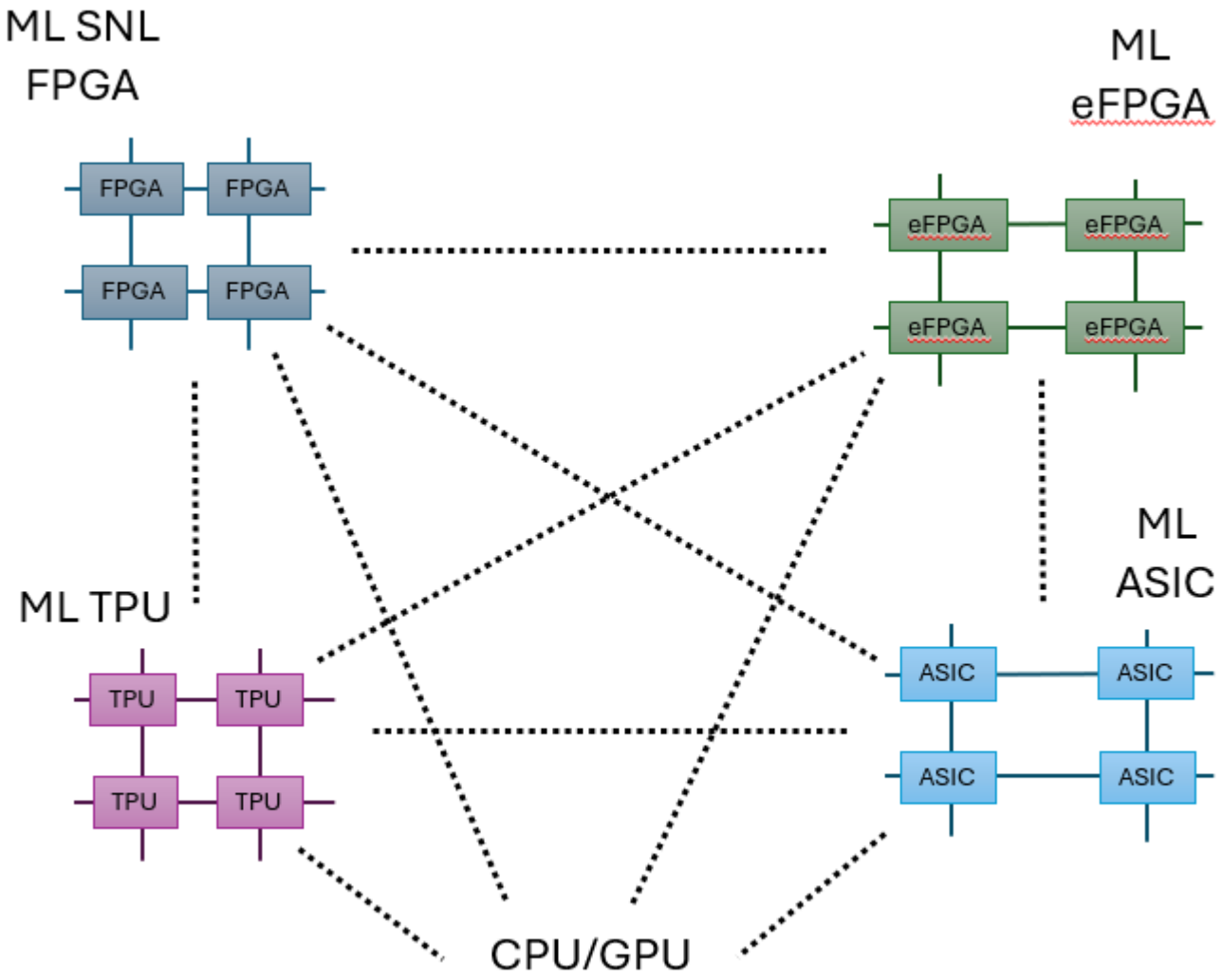
*Julia Gonski - FPD

ATLAS Calorie Meter Dataset

MLP Mixer (Trasformer like architecture)



Toward Distributed ML Inference Across Detector Hardware

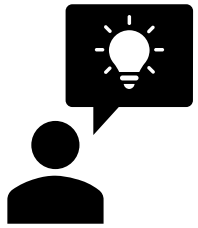


- ML inference can be **split across multiple hardware tiers**: FPGA, eFPGA, ASIC, TPU, CPU/GPU
- Each hardware type has **unique compute, power, and latency profiles**

Challenges in Multi-Hardware ML Pipelines

- **Intermediate data transfer overhead** between hardware units
- **Latency bottlenecks** due to mismatched speeds and protocols
- **Exploding tensor sizes** in intermediate layers
- Lack of tools for **model partitioning and co-optimization**

Learned from Hardware-Aware ML Inference



What We've Learned from Hardware-Aware ML Inference

- **No One-Size-Fits-All Hardware**
  - Each ML architecture exhibits different performance trends depending on the **type of computation** (dense ops vs. convolutions vs. token mixing).
- **Inference behavior varies dramatically** across CPU, GPU, FPGA, Groq, and ASIC TPU — driven by architecture–hardware affinity.
- We need to take design insights based on hardware, and NN architecture.

Challenges Ahead

- **We haven't yet split models across hardware** — but it's a natural next step.
  - Splitting risks **data explosion** between layers
  - If you want to run split NN between hardware need to figure out the strategies to transfer the exploded data in a better way.
  - Not all systems are same. So the benchmarking becomes challenging.
- **Lack of Scientific Benchmark Datasets**
  - **No** shared **MNIST, CIFAR, or ImageNet for science**.
  - We **need benchmark**-ready **datasets** from **LHC, LCLS-II, DIIID**, etc.
  - These should support inference benchmarking **with scientific relevance**.
  - This can be later useful in academic environment as well.