Fermilab

May 13, 2026

FAN-C Feats:
Real-Time Science with Fast Adaptive Neural Control

Nhan Tran, Fermilab

NPTwins-2026

U.S. DEPARTMENT of ENERGY

Fermi National Accelerator Laboratory is managed by FermiForward for the U.S. Department of Energy Office of Science

FERMI NATIONAL ACCELERATOR LABORATORY

FAN-C

Fast Adaptive Neural Control

Neural control, served fancy.



F FAST



Rapid sensing, processing, and response.

Milliseconds matter.
Speed drives performance.

A ADAPTIVE



Continuously learns and adjusts to changing conditions.

Context changes.
FAN-C adapts.

N NEURAL



Leveraging neural networks for perception, prediction, and control.

Inspired by biology.
Powered by computation.

C CONTROL



Precise, robust control for complex, dynamic environments.

Stay on course.
Achieve the objective.

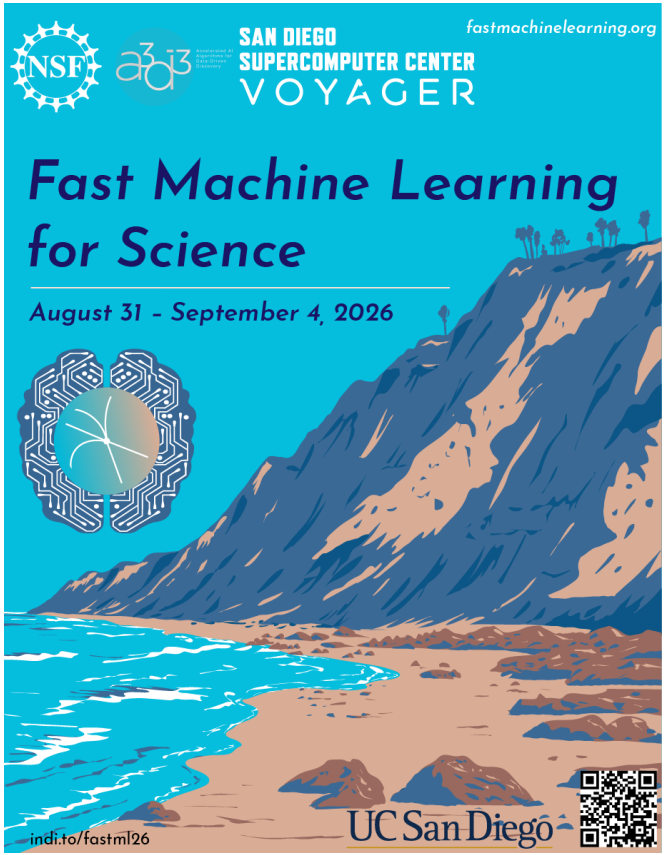
FAN-C: Because performance deserves a seat at the table.

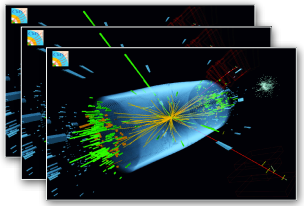
Outline

- Fast Control
- Fast Neural Control
- Fast Adaptive Neural Control

"Scientific discoveries come from groundbreaking ideas and the capability to validate those ideas by testing nature at new scales - finer and more precise temporal and spatial resolution. This is leading to an explosion of data that must be interpreted, and ML is proving a powerful approach. The more efficiently we can test our hypotheses, the faster we can achieve discovery. To fully unleash the power of ML and accelerate discoveries, it is necessary to embed it into our scientific process, into our instruments and detectors."

Applications and Techniques for Fast Machine Learning in Science
https://doi.org/10.3389/fdata.2022.787421

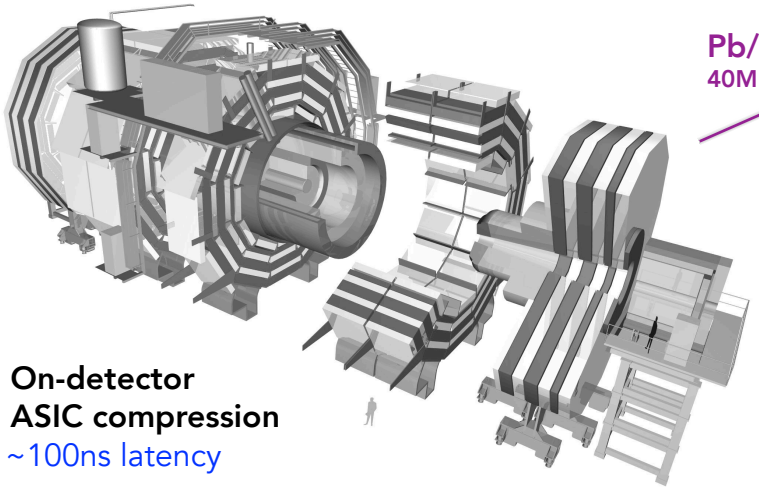




CMS Experiment

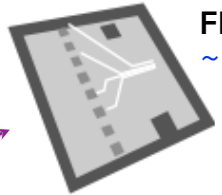
40MHz collision rate

~1B detector channels

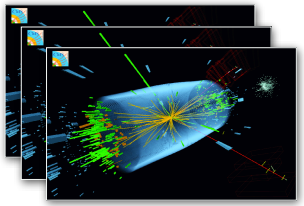


On-detector
ASIC compression
~100ns latency

Pb/s
40MHz



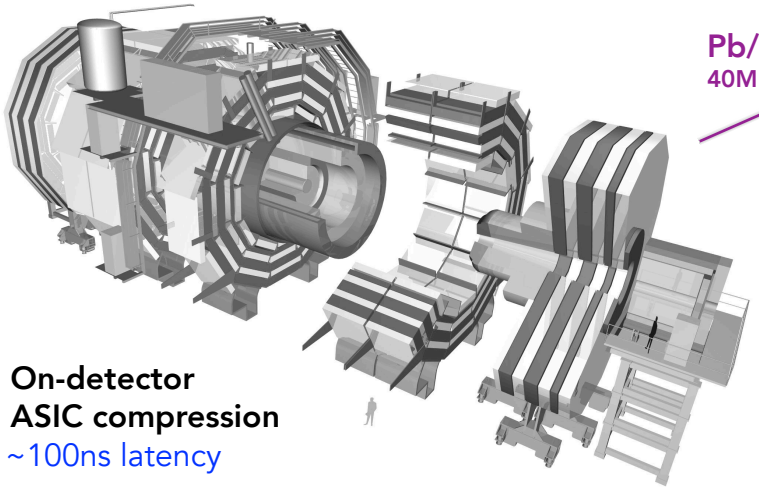
FPGA filter stack
~us latency



CMS Experiment

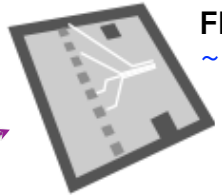
40MHz collision rate

~1B detector channels



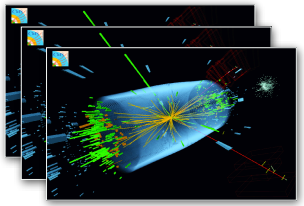
On-detector
ASIC compression
~100ns latency

Pb/s
40MHz



FPGA filter stack
~us latency

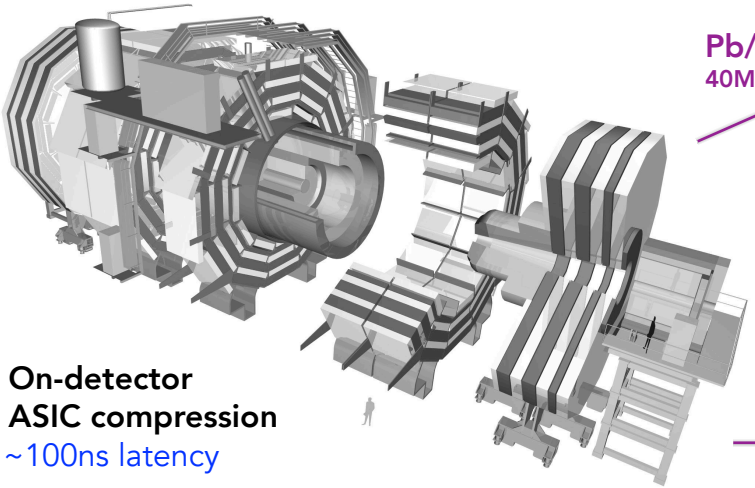
COMPARABLE TO GLOBAL INTERNET TRAFFIC BANDWIDTH



CMS Experiment

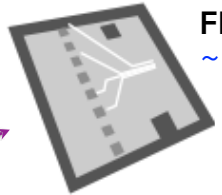
40MHz collision rate

~1B detector channels



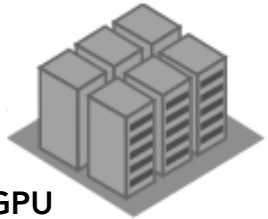
On-detector
ASIC compression
~100ns latency

Pb/s
40MHz



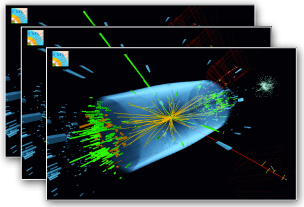
FPGA filter stack
~us latency

10s Tb/s
100s kHz



On-prem CPU/GPU filter farm
~100 ms latency

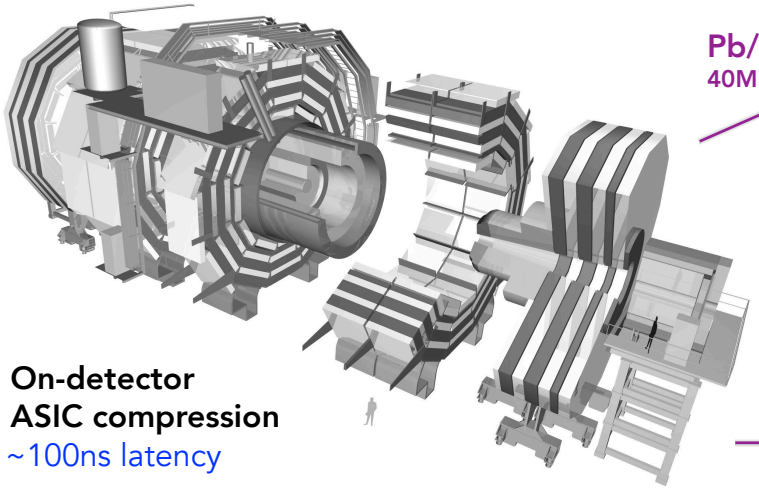
COMPARABLE TO GLOBAL INTERNET TRAFFIC BANDWIDTH



CMS Experiment

40MHz collision rate

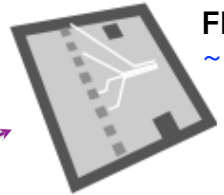
~1B detector channels



On-detector ASIC compression

~100ns latency

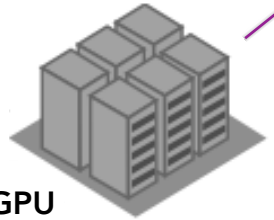
Pb/s
40MHz



FPGA filter stack

~us latency

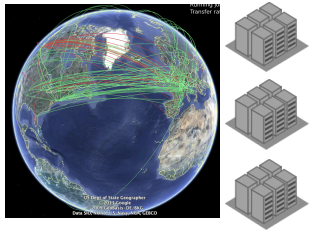
10s Tb/s
100s kHz



On-prem CPU/GPU filter farm

~100 ms latency

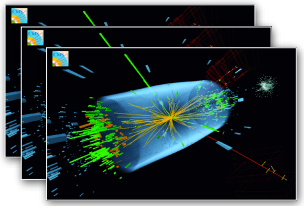
10s Gb/s
~5 kHz



Worldwide computing grid

Exabyte-scale datasets

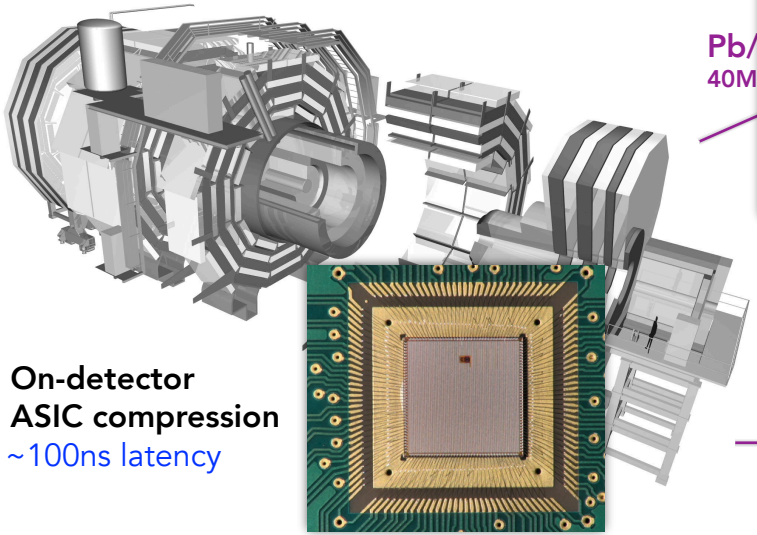
COMPARABLE TO GLOBAL INTERNET TRAFFIC BANDWIDTH



CMS Experiment

40MHz collision rate

~1B detector channels

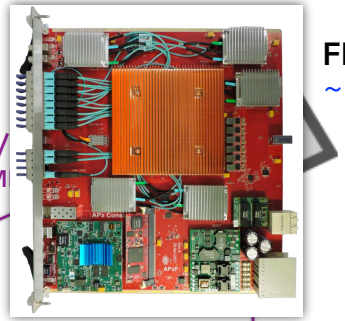


On-detector ASIC compression

~100ns latency

Pb/

40M

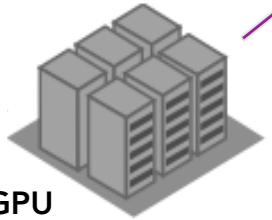


FPGA filter stack

~us latency

10s Tb/s

100s kHz

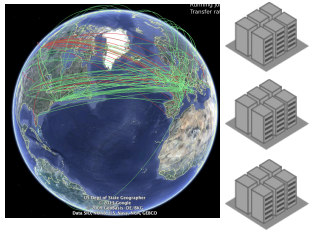


On-prem CPU/GPU filter farm

~100 ms latency

10s Gb/s

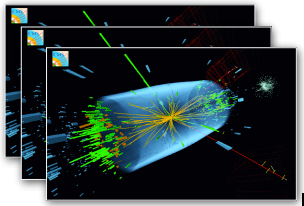
~5 kHz



Worldwide computing grid

Exabyte-scale datasets

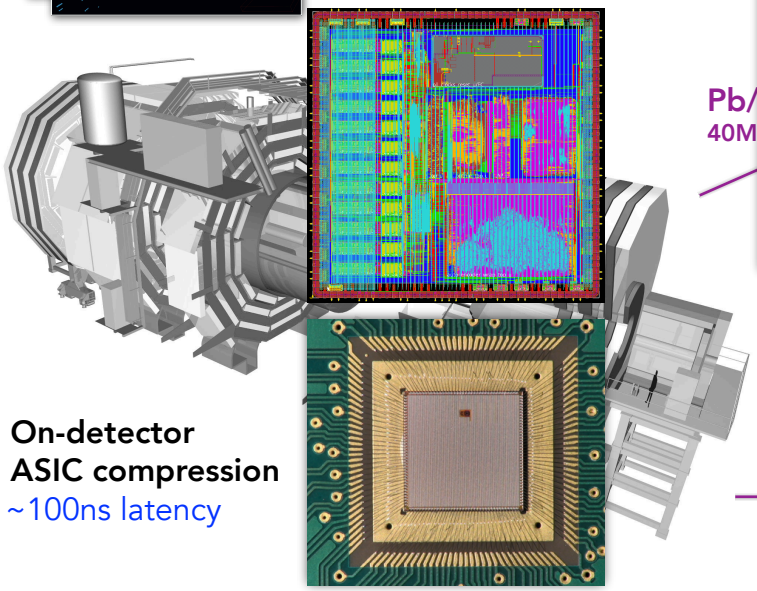
COMPARABLE TO GLOBAL INTERNET TRAFFIC BANDWIDTH



CMS Experiment

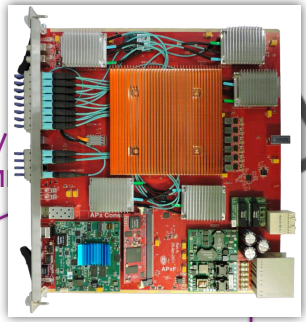
40MHz collision rate

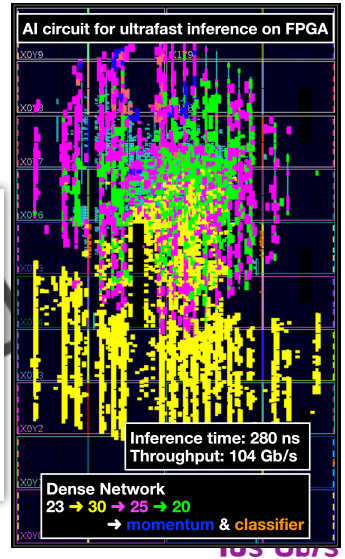
~1B detector channels



On-detector
ASIC compression
~100ns latency

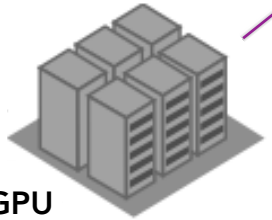
Pb/
40M



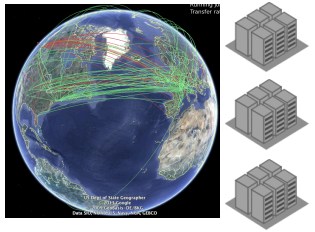


~5 kHz

10s Tb/s
100s kHz



On-prem CPU/GPU
filter farm
~100 ms latency



Worldwide
computing grid
Exabyte-scale
datasets

COMPARABLE TO GLOBAL INTERNET TRAFFIC BANDWIDTH

The need for speed

The need for speed

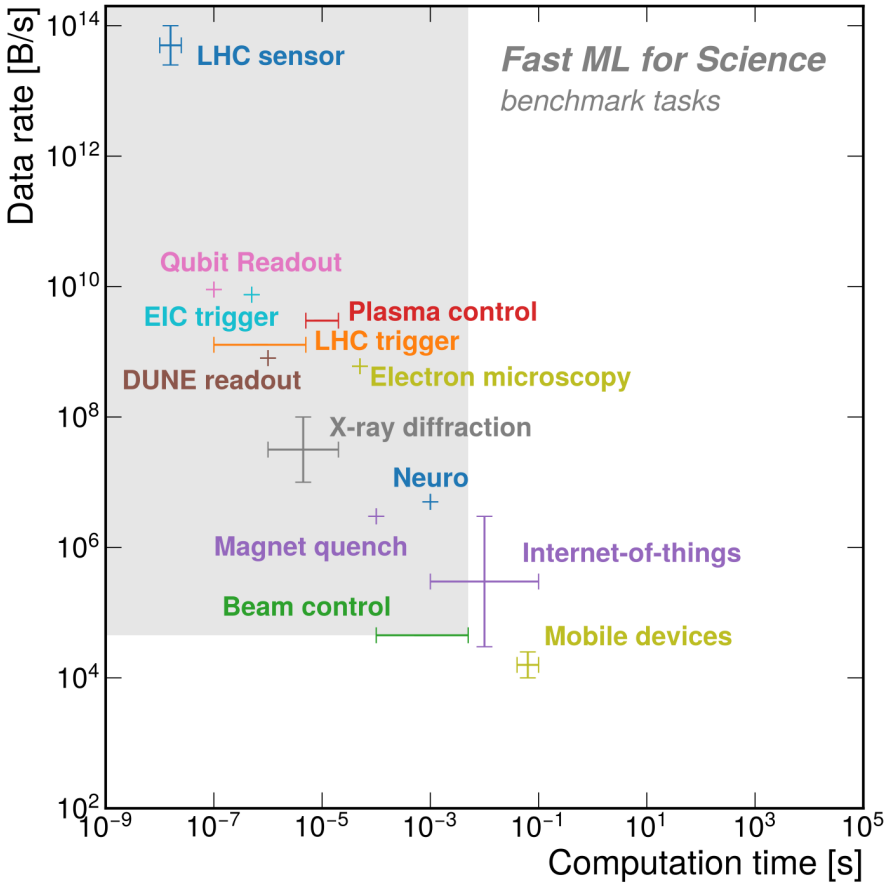
Benchmarks bring innovation

The Fast ML for Science community aims to bring **seemingly different domains** together to develop **techniques, tools, and platforms** for challenges that **far outpace industry.**

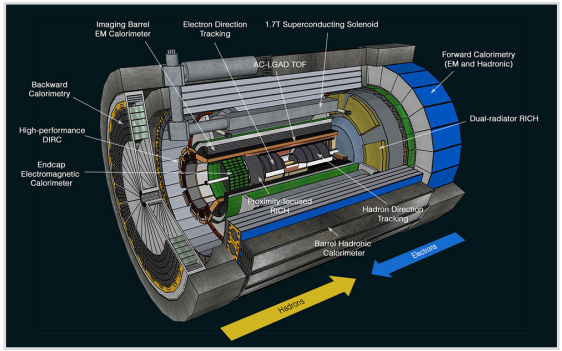
The need for speed

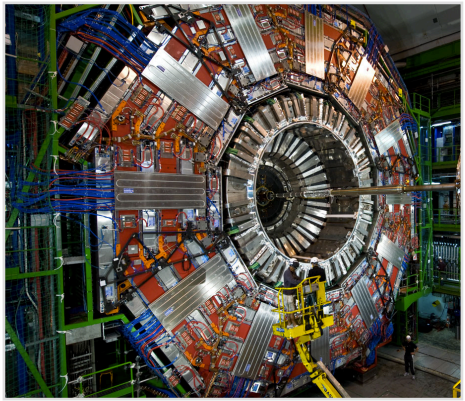
Benchmarks bring innovation

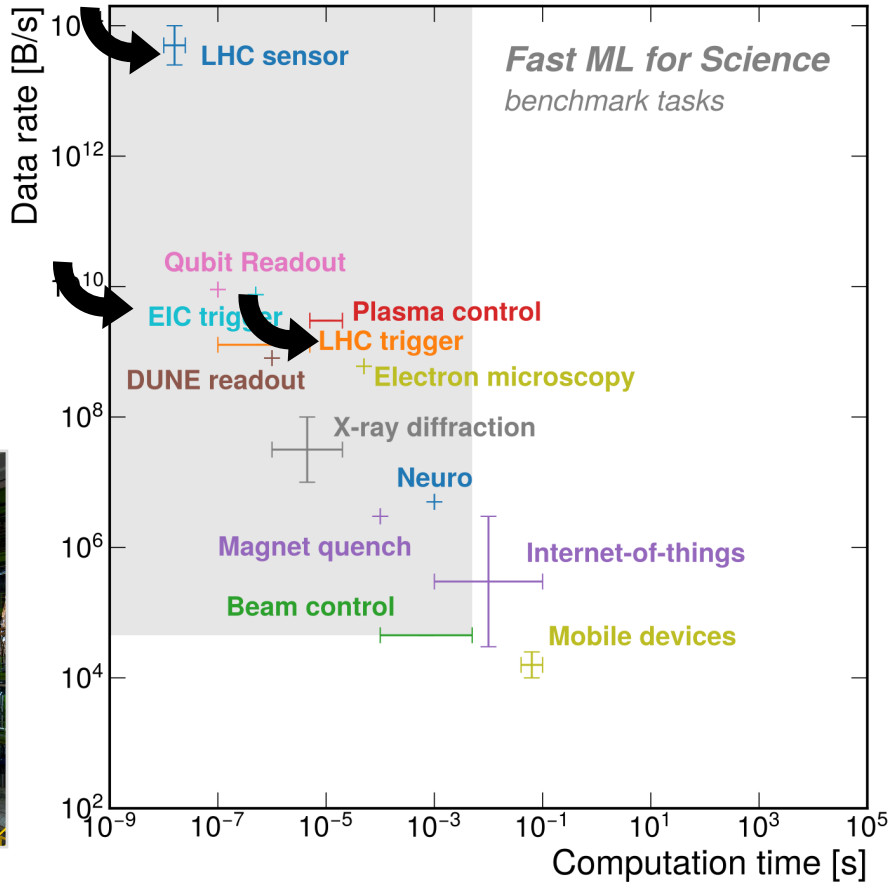
The Fast ML for Science community aims to bring **seemingly different domains** together to develop **techniques, tools, and platforms** for challenges that **far outpace industry.**



The need for speed

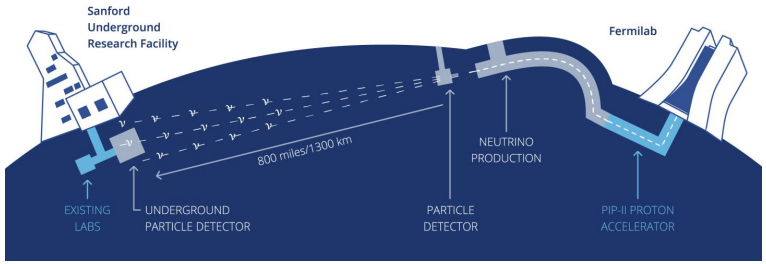






FERMI NATIONAL ACCELERATOR LABORATORY

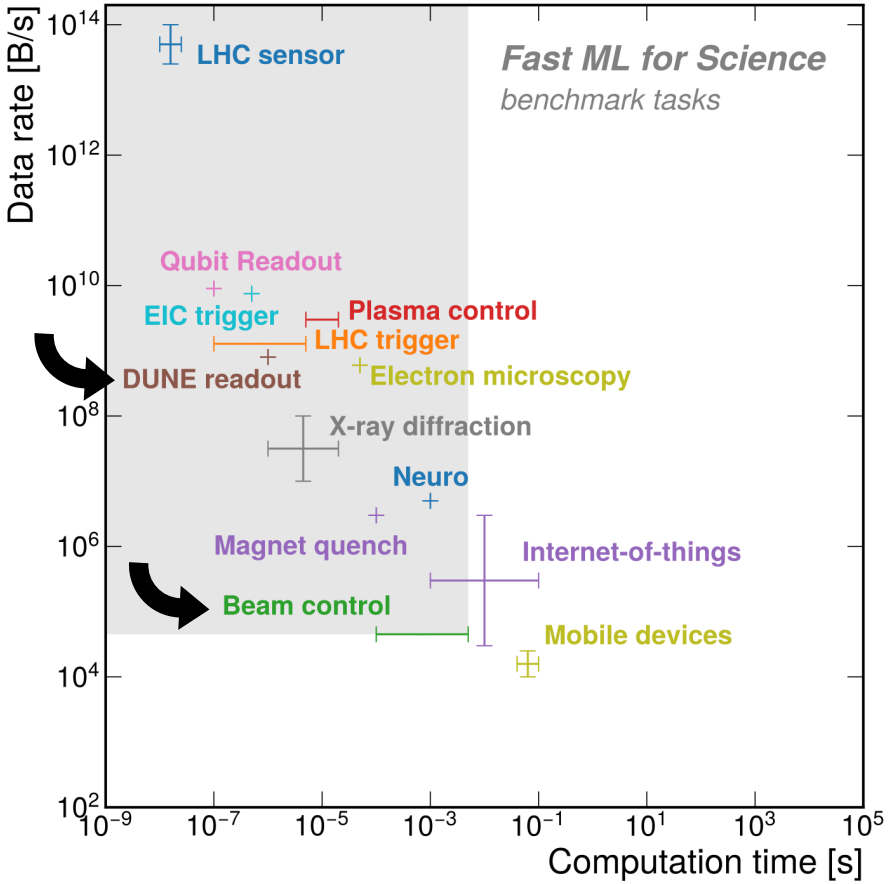
The need for speed





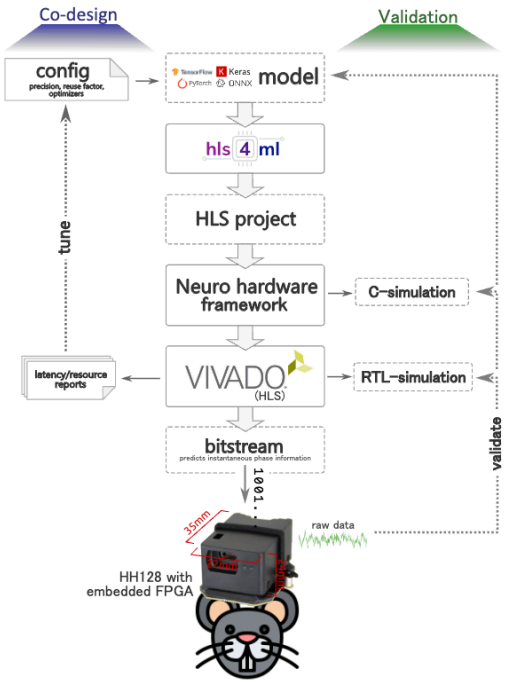


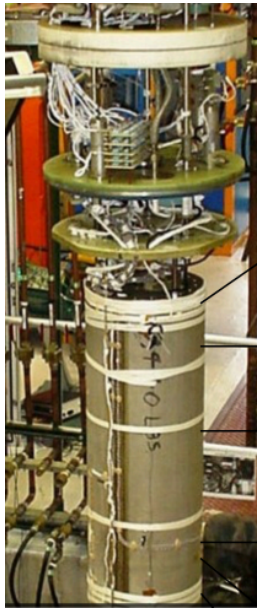
Fermilab accelerator complex

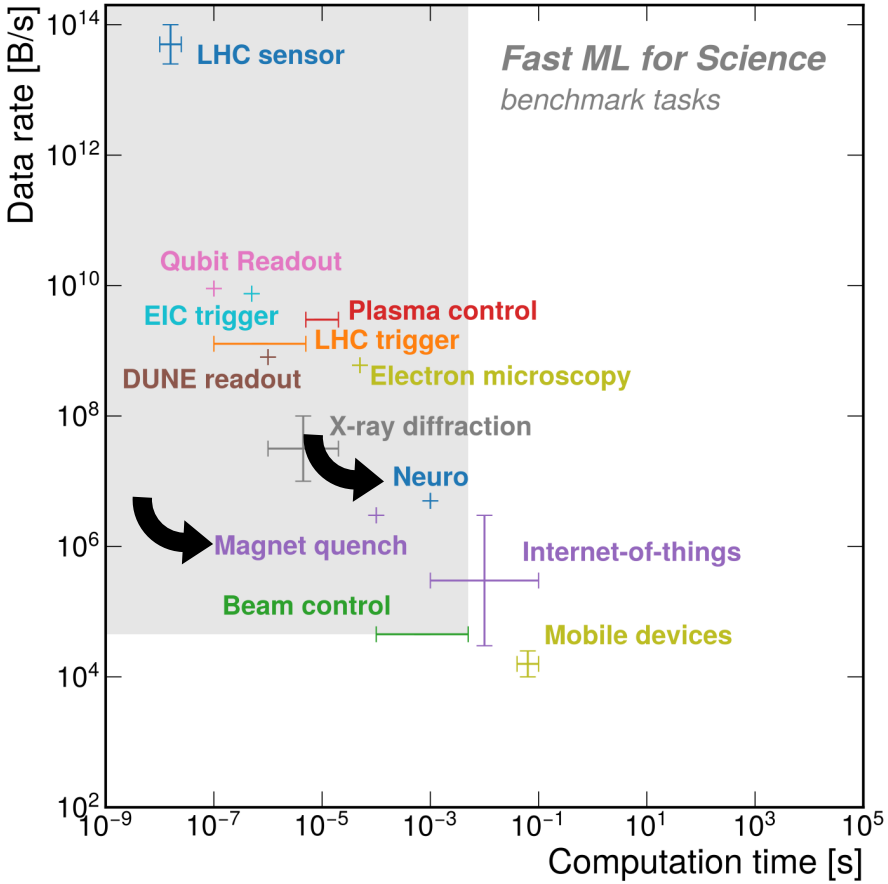


FERMI NATIONAL ACCELERATOR LABORATORY

The need for speed

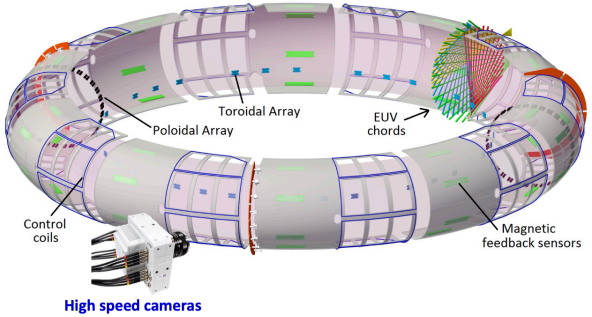


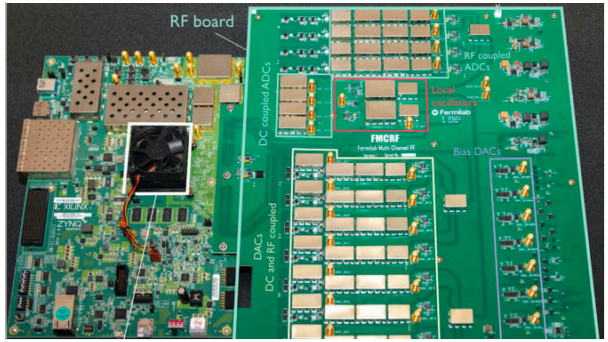


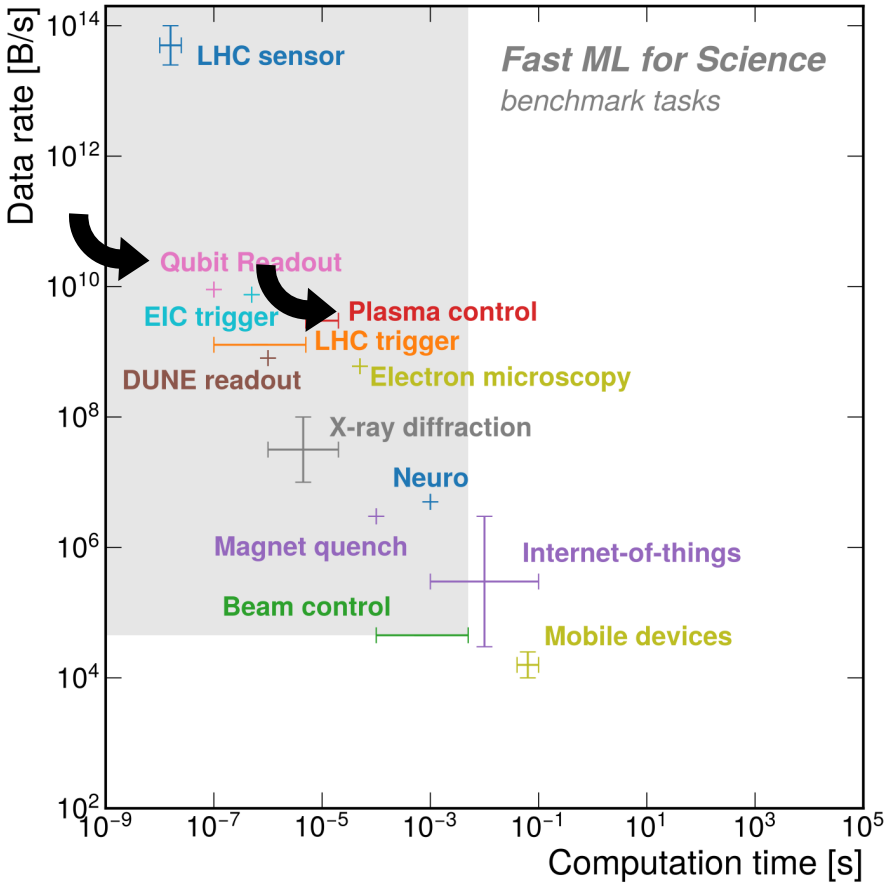


FERMI NATIONAL ACCELERATOR LABORATORY

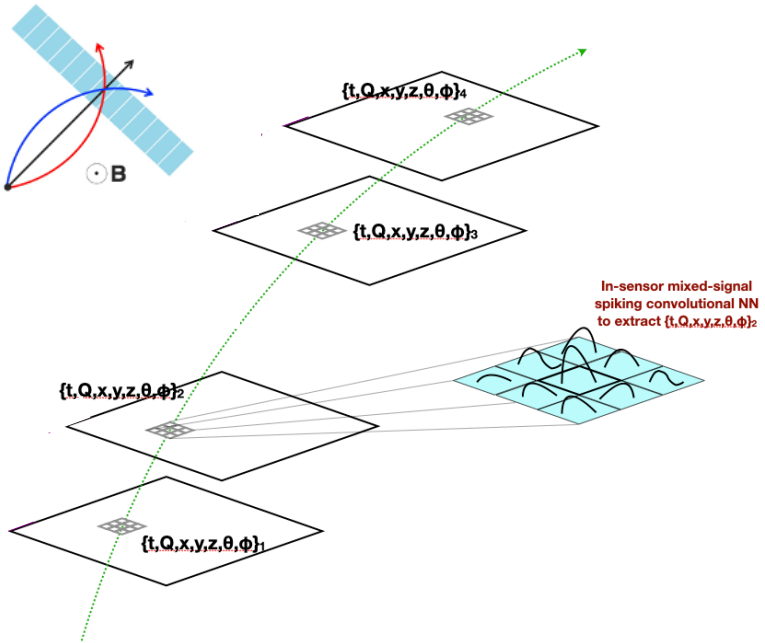
The need for speed



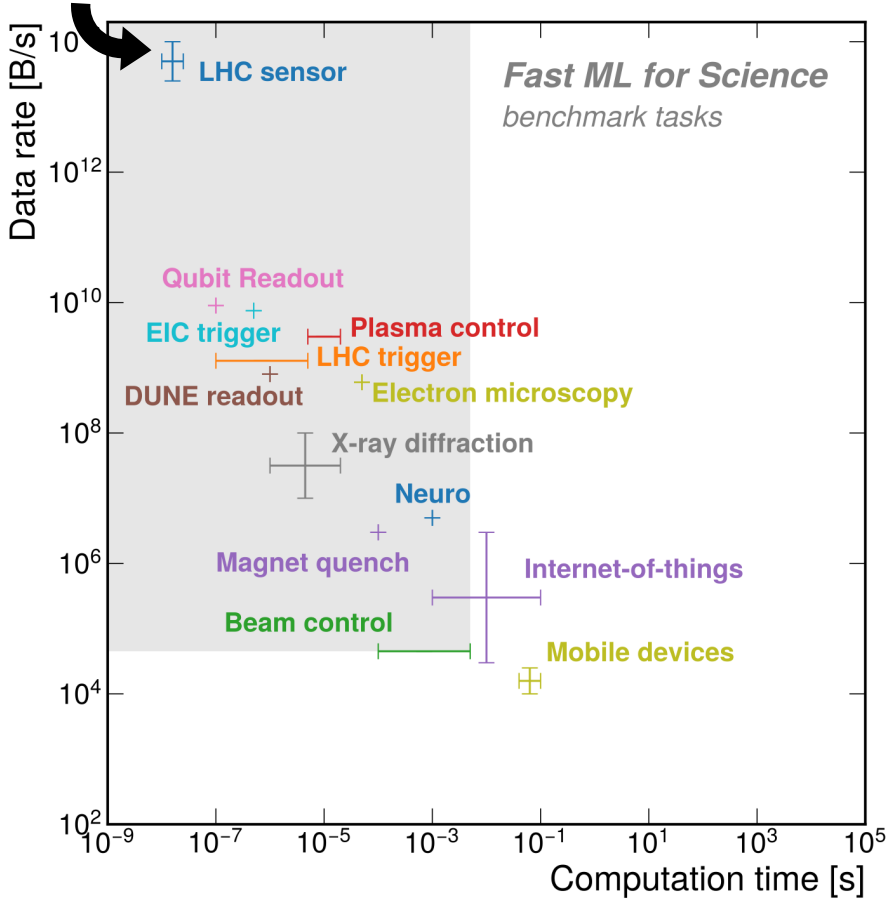




The need for speed

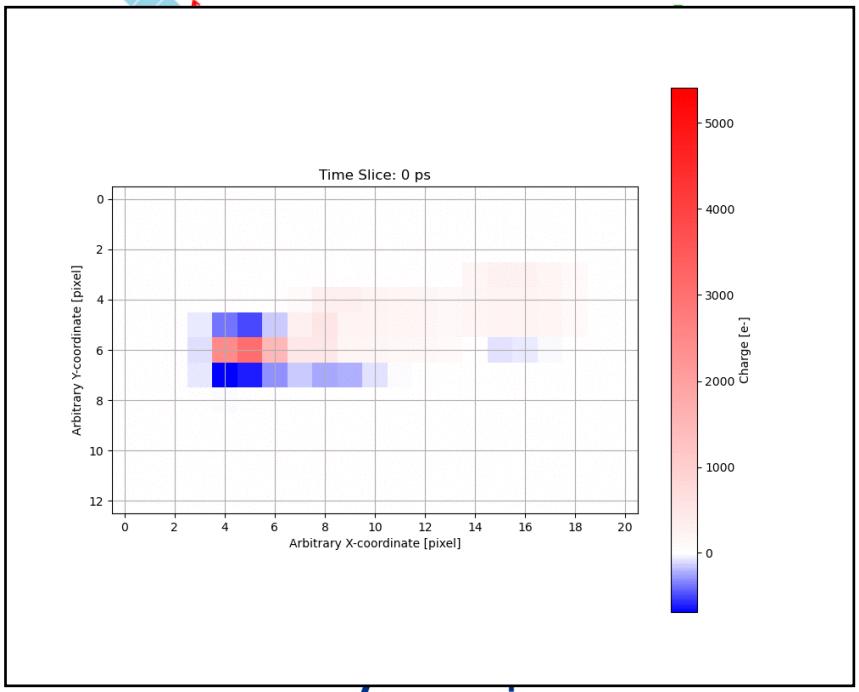


Pixel information in the trigger brings fundamentally new capabilities to LHC experiments - but rates are massive!

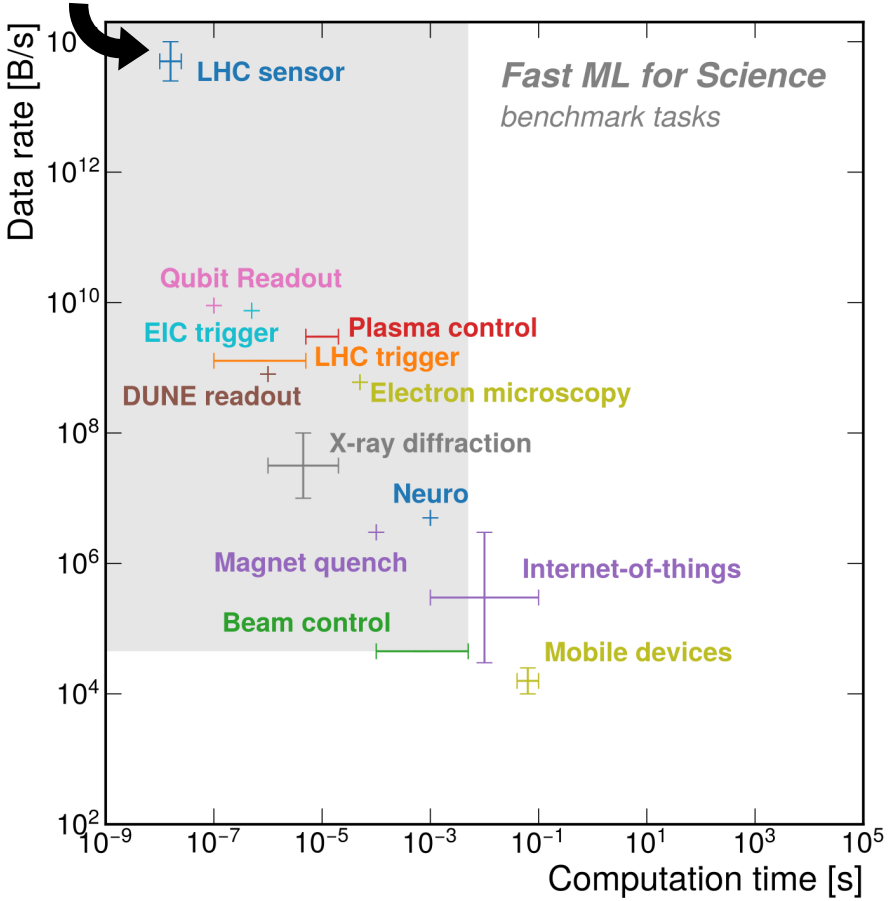


FERMI NATIONAL ACCELERATOR LABORATORY

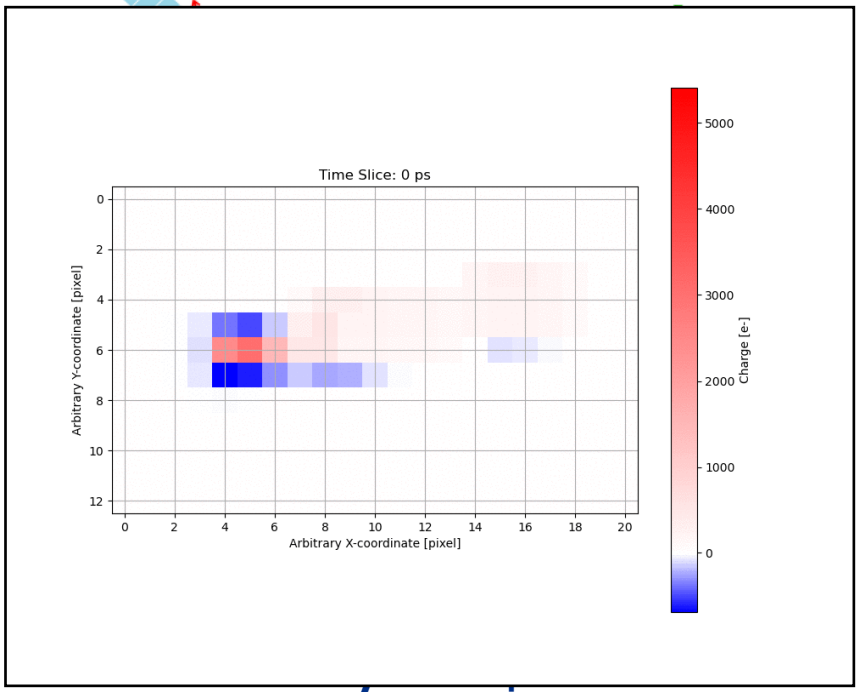
The need for speed



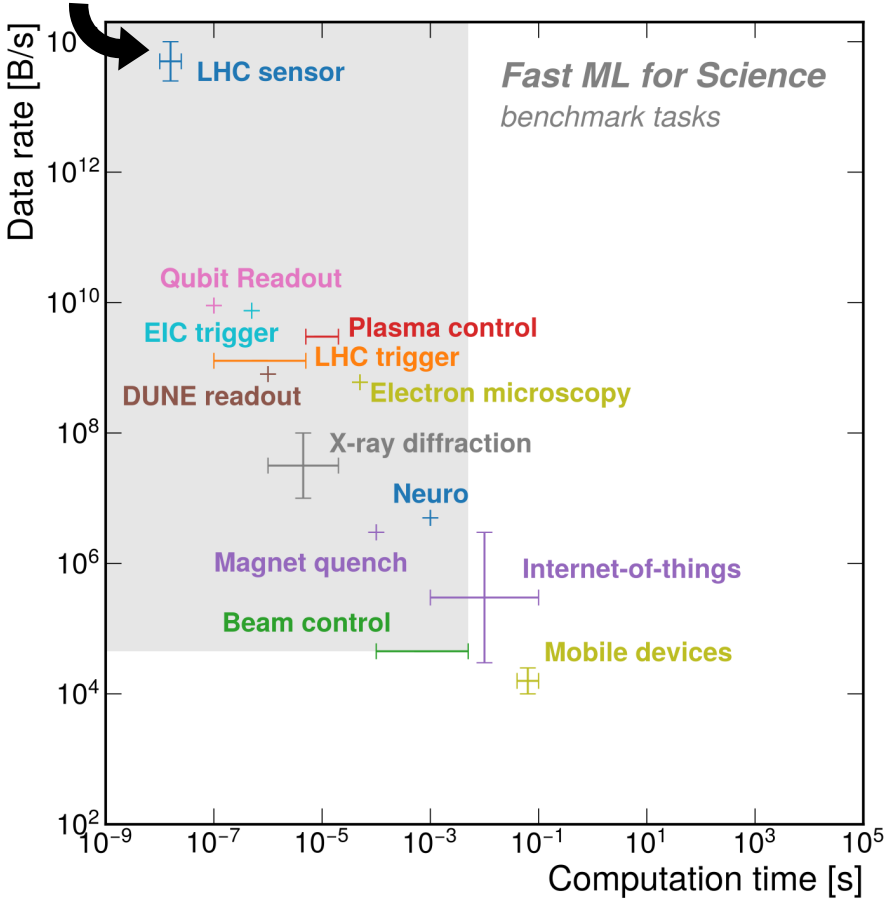
experiments - but rates are massive!



The need for speed



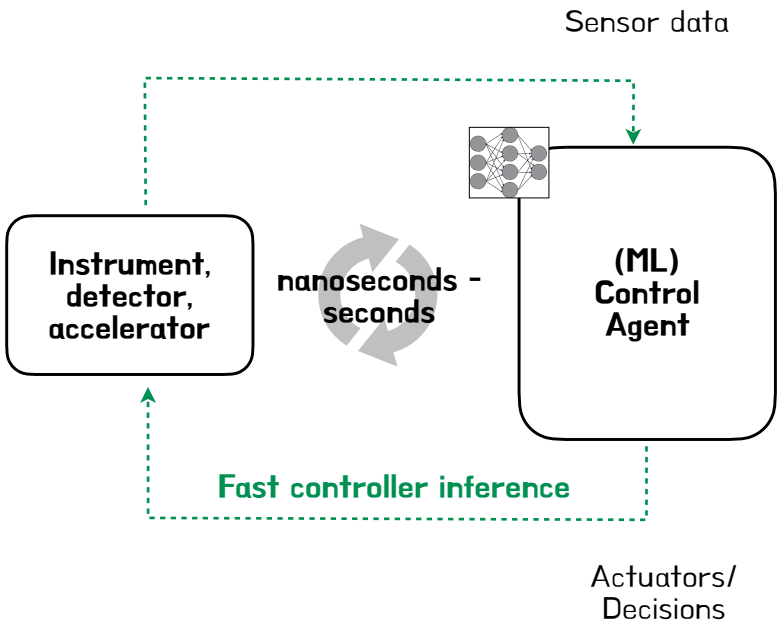
experiments - but rates are massive!



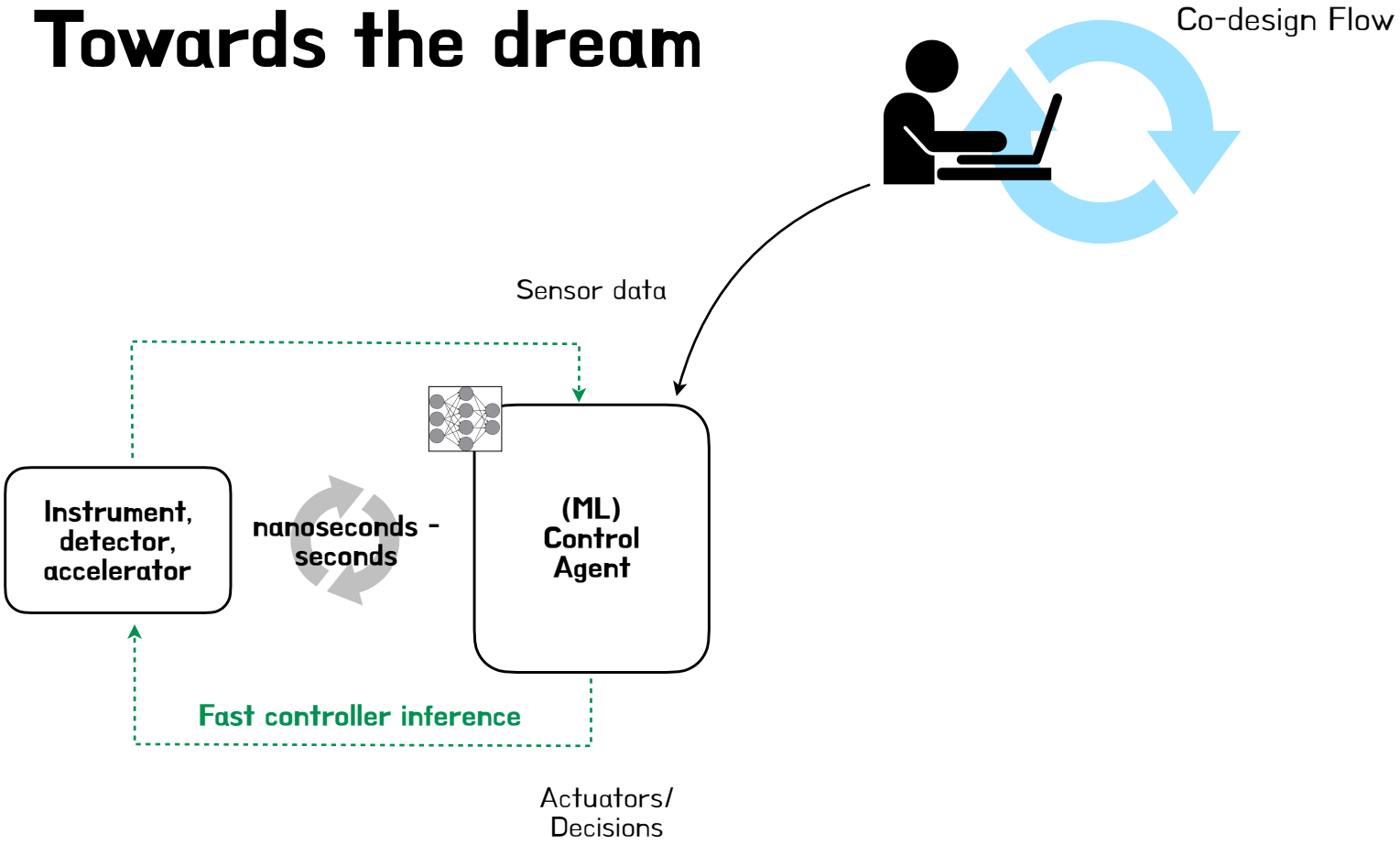
Outline

- Fast Control
- **Fast Neural Control**
- Fast Adaptive Neural Control

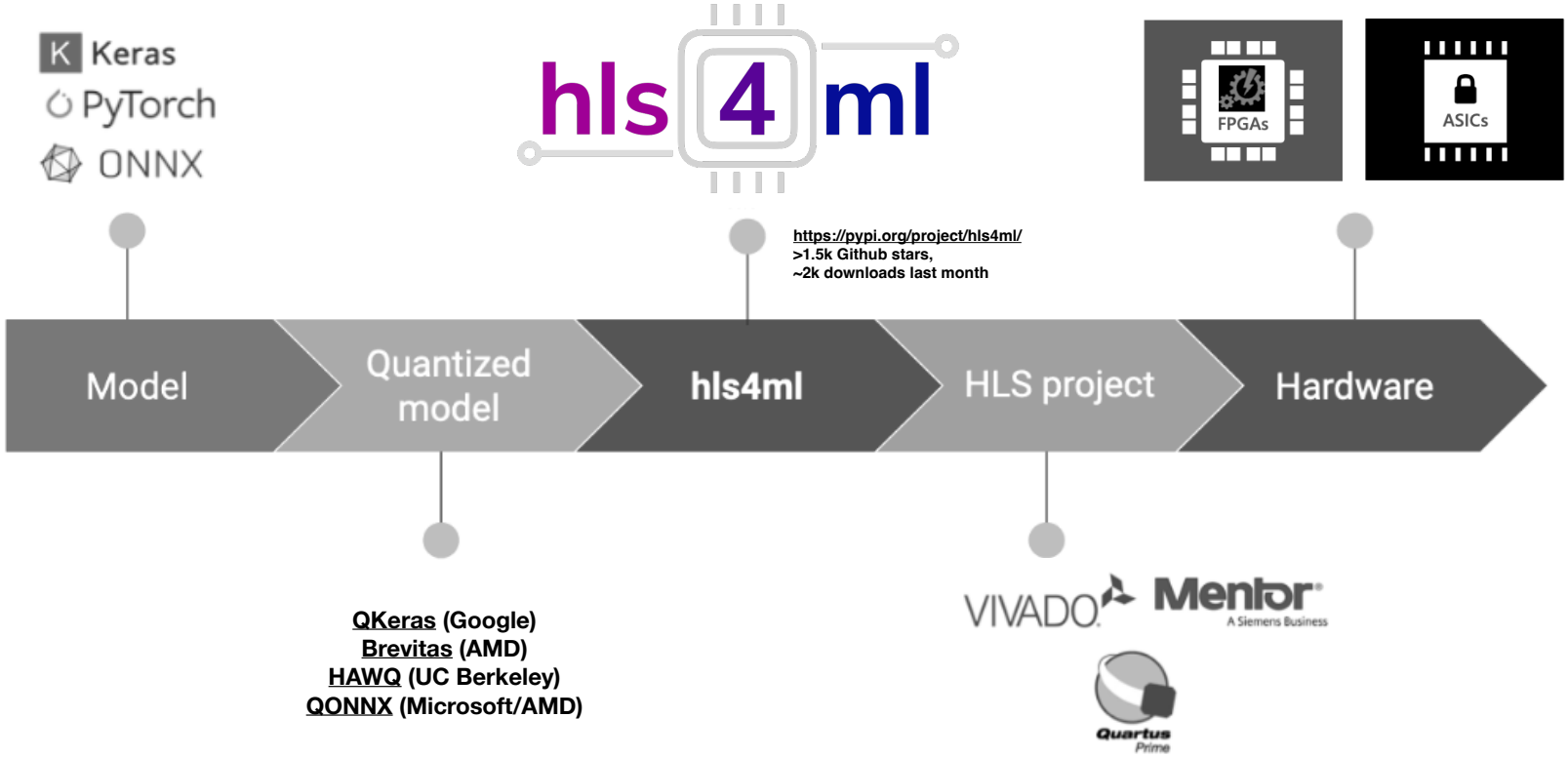
Towards the dream



Towards the dream

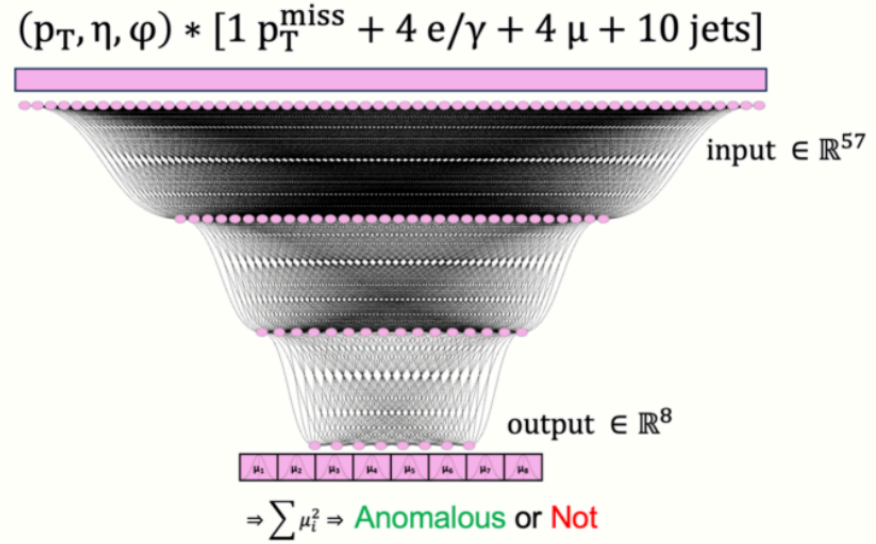


Efficient tools for efficient, embedded AI



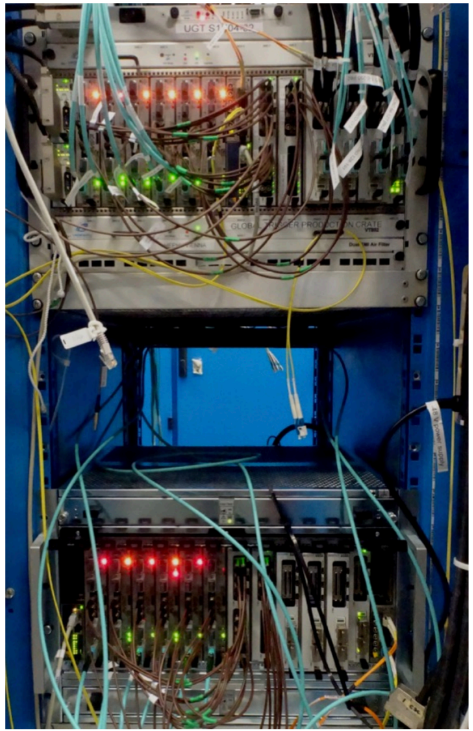
Anomaly detection in 50 ns

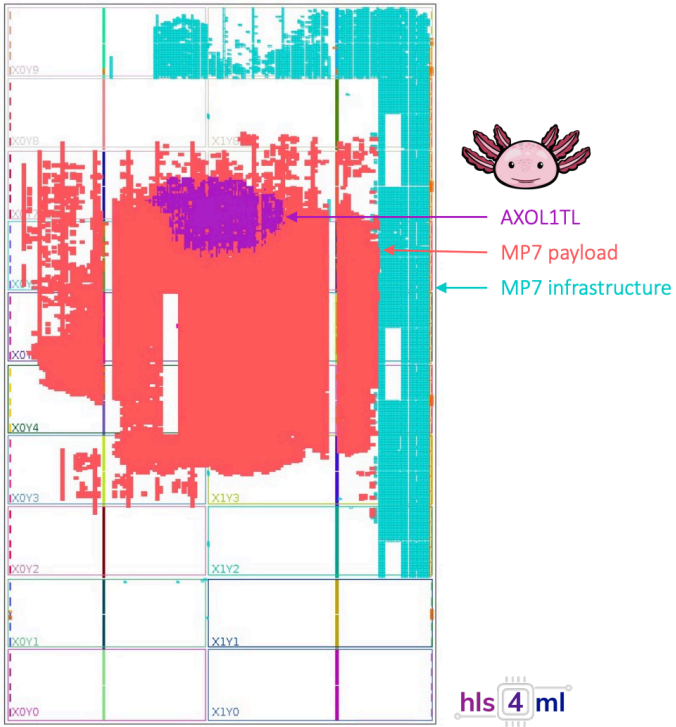
https://cds.cern.ch/record/2904695



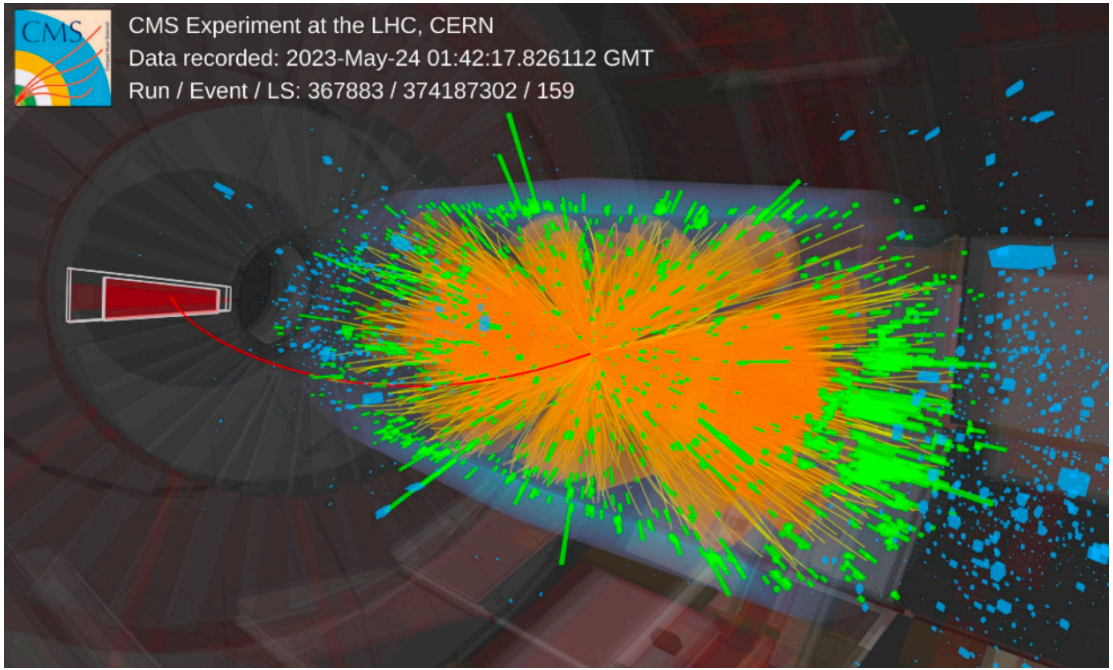
	Latency	LUTs	FFs	DSPs	BRAMs
AXOL1TL	2 ticks 50 ns	2.1%	~0	0	0

Anomaly detection in 50 ns



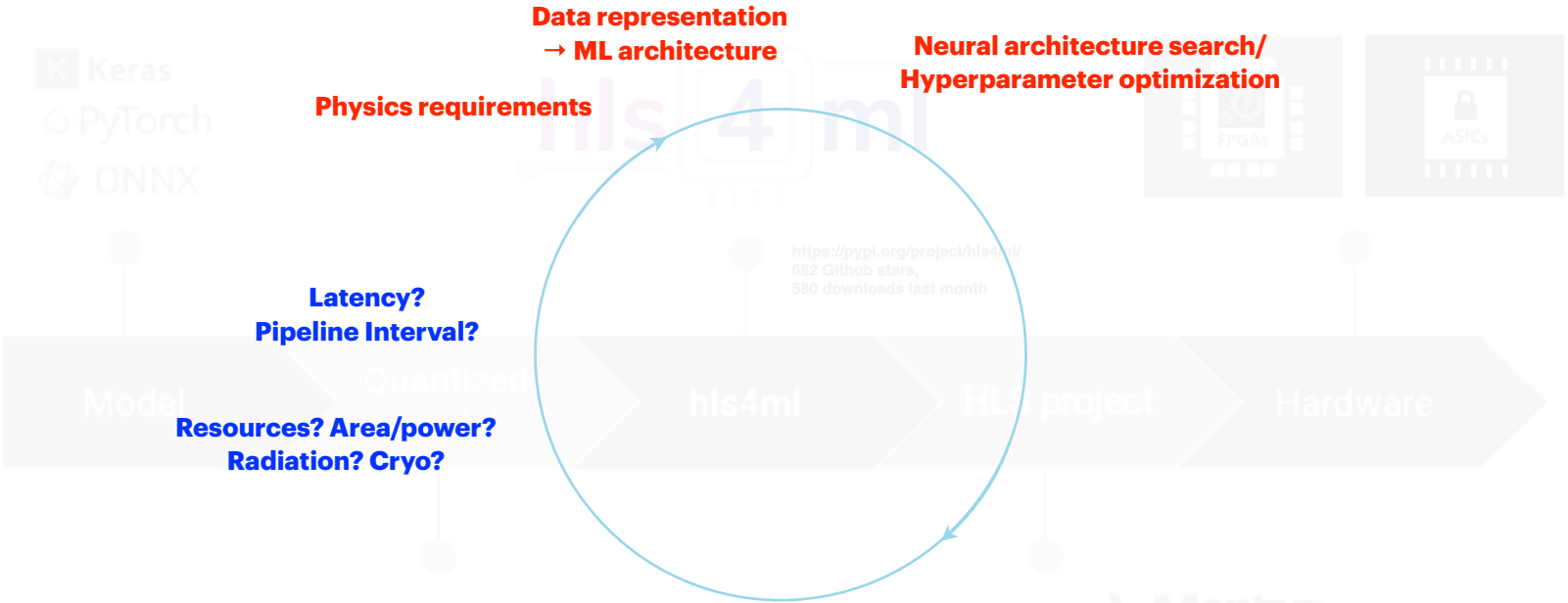


Anomaly detection in 50 ns



Up to factor of 10 increase in signal efficiency!

Hardware - algorithm codesign



Hardware - algorithm codesign

Data representation
→ ML architecture

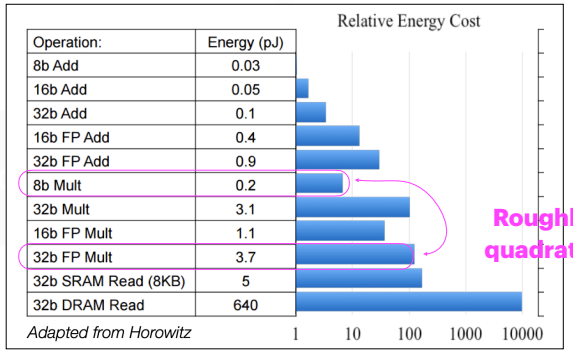
Neural architecture search/
Hyperparameter optimization

Physics requirements

Quantize network

Latency?
Pipeline Interval?

Resources? Area/power?
Radiation? Cryo?



Hardware - algorithm codesign

Data representation
→ ML architecture

Neural architecture search/
Hyperparameter optimization

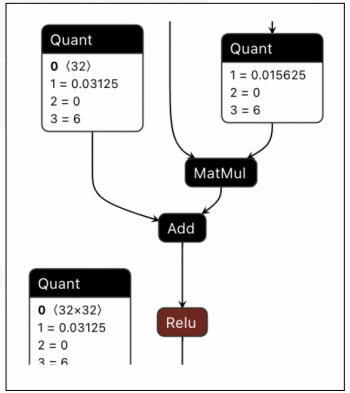
Physics requirements

Quantize network

Latency?
Pipeline Interval?

Intermediate (quantized)
representations

Resources? Area/power?
Radiation? Cryo?



Hardware - algorithm codesign

Data representation
→ ML architecture

Neural architecture search/
Hyperparameter optimization

Physics requirements

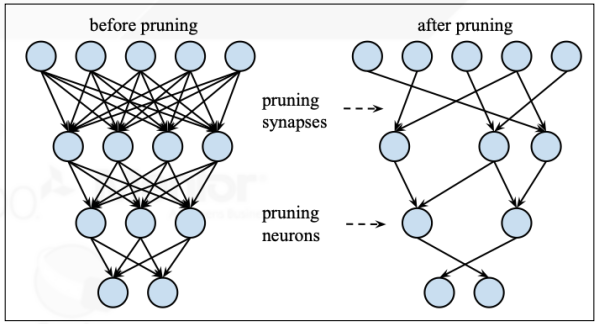
Quantize network

Latency?
Pipeline Interval?

Intermediate (quantized)
representations

Resources? Area/power?
Radiation? Cryo?

Pruning/sparsity?



FERMI NATIONAL ACCELERATOR LABORATORY

Hardware - algorithm codesign

Data representation
→ ML architecture

Neural architecture search/
Hyperparameter optimization

Physics requirements

Quantize network

Intermediate (quantized)
representations

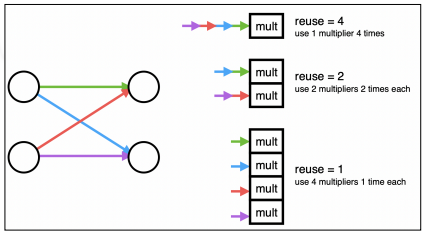
Latency?
Pipeline Interval?

Pruning/sparsity?

Resources? Area/power?
Radiation? Cryo?

Microarchitecture

Parallelization

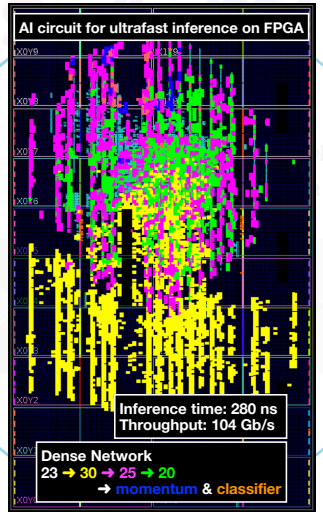


Hardware - algorithm codesign

Data representation
→ ML architecture

Neural architecture search/
Hyperparameter optimization

Physics requirements



Quantize network

Latency?
Pipeline Interval?

Intermediate (quantized)
representations

Resources? Area/power?
Radiation? Cryo?

Pruning/sparsity?

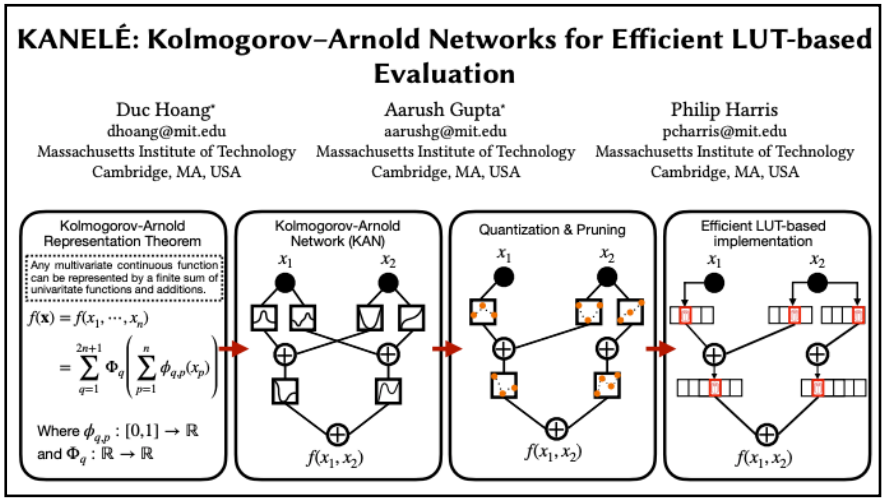
Microarchitecture

Synthesize, validate design,
satisfy design rules/timing

Parallelization

FERMI NATIONAL ACCELERATOR LABORATORY

Efficient codesign - emerging methods





Dataset	Model	Accuracy (%)	LUT	FF	DSP	BRAM	F_{max} (MHz)	Latency (ns)	Area×Delay (LUT×ns)
JSC CERNBox	**KANELÉ**	**75.1**	**5034**	1917	0	0	**870**	8.1	$\mathbf{4.1 \times 10^4}$
	NeuraLUT-Assemble [6]	75.0	8539	1332	0	0	352	**5.7**	4.87×10^4
	AmigoLUT-NeuraLUT [47]	74.4	42742	4717	0	0	520	9.6	4.10×10^5
	PolyLUT-Add [30]	75.0	36484	1209	0	0	315	16	5.84×10^5
	NeuraLUT [5]	**75.1**	92357	4885	0	0	368	14	1.29×10^6
	PolyLUT [4]	75.0	246071	12384	0	0	203	25	6.15×10^6
	LogicNets [42]	72.0	37931	**810**	0	0	427	13	4.93×10^5

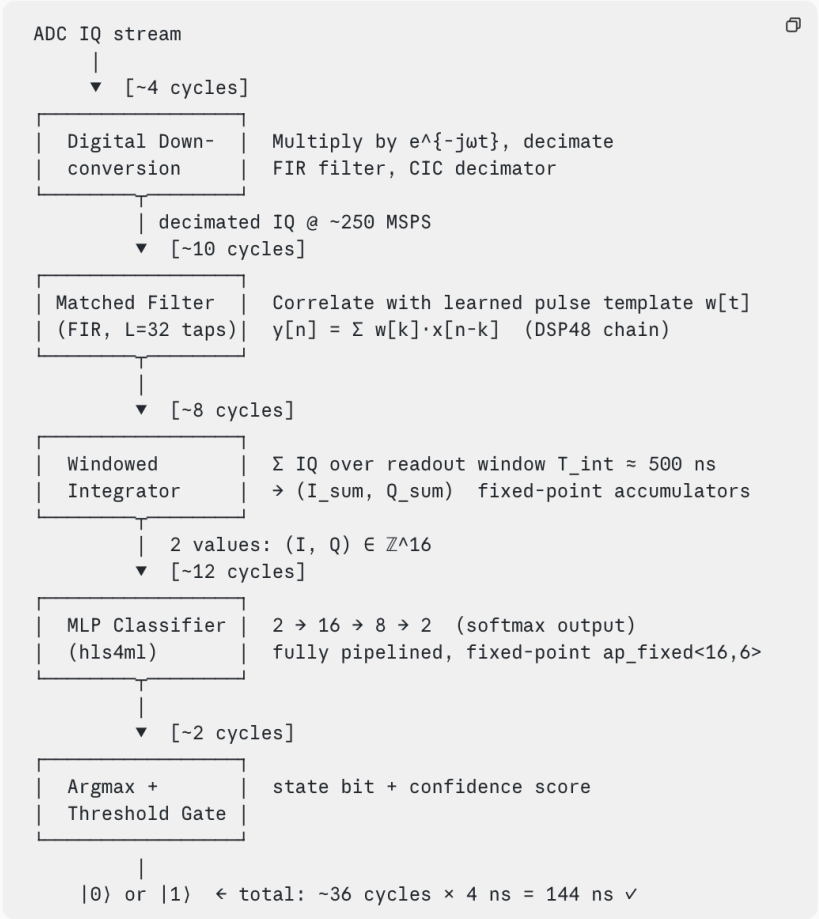
Open-source, application driven

design an ML algorithm for qubit readout on an FPGA < 200 ns

ML Qubit Readout on FPGA < 200 ns

This is a real-time signal discrimination problem. Here's a complete architecture targeting ~150 ns end-to-end latency at 250 MHz (4 ns/cycle → ~37 cycle budget).

Algorithm Architecture



driven

re targeting ~150 ns end-to-end latency at 250 MHz (4 ns/cycle → ~37 cycle budget).

FERMI NATIONAL ACCELERATOR LABORATORY

Algorithm Architecture



FPGA HLS Implementation (hls4ml)

```python
# deploy_hls4ml.py
import hls4ml
from hls4ml.model.optimizer import get_available_passes

def export_to_hls(keras_model, scaler):
    config = hls4ml.utils.config_from_keras_model(
        keras_model,
        granularity='name'
    )

    # Fixed-point precision: <total_bits, integer_bits>
    config['Model']['Precision'] = 'ap_fixed<16,6>'
    config['Model']['ReuseFactor'] = 1   # full unroll = minimum latency

    for layer in config['LayerName']:
        config['LayerName'][layer]['Strategy'] = 'Latency'  # not Resource
        config['LayerName'][layer]['Precision'] = 'ap_fixed<16,6>'

    hls_model = hls4ml.converters.convert_from_keras_model(
        keras_model,
        hls_config=config,
        output_dir='readout_hls',
        backend='Vivado',
        board='ZCU216',       # typical QPU FPGA platform
        part='xczu49dr-ffvf1760-2-e',
    )
    hls_model.compile()

    # Verify latency report
    hls_model.build(csim=False, synth=True, cosim=False, export=True)
    report = hls_model.get_timing_report()
    print(f"Estimated latency: {report['WorstCaseLatency']} cycles")

    return hls_model
```

Revised Honest Budget

```
Stage                    Cycles     Time      Confidence
──────────────────────────────────────────────────────────
DDC + CIC                  9        36 ns     medium
                                              (was 6 → optimistic)
Matched filter             8        32 ns     high
FIR group delay fill      16        64 ns     omitted in original!
  (startup, first pulse only)
Integrator pipeline        2         8 ns     high
  (window duration is T_pulse, not counted here)
MLP (hls4ml, synth TBD)  12–22    48–88 ns   medium
Output register            2         8 ns     high
──────────────────────────────────────────────────────────
Processing latency        33–43   132–172 ns  ✓ (after window ends)
Pulse-start latency       +T_pulse             depends on pulse length
```

The 200 ns budget is achievable for processing latency (from end of integration window to decision). For total latency from pulse start, you need `T_pulse < ~60–70 ns`, which is a tight but real constraint in superconducting qubit readout.

Total Resource Summary

Resource	Used	ZCU216 Available	%
DSP48E2	~200 (16-bit) / ~100 (8-bit)	1968	10% / 5%
CLB LUTs	~600	425,280	<1%
FFs	~1000	850,560	<1%
BRAM	0 (weights in regs)	1080	0%

The design is DSP-bound if using 16-bit precision. Using 8-bit QAT weights (which real hls4ml deployments use) cuts DSP count in half with minimal accuracy loss.

Bottom line on methodology: DDC and FIR estimates follow directly from DSP48E2 pipeline depth + adder tree depth, which are deterministic. The MLP estimate was an informed hls4ml heuristic — `hls_model.build(synth=True)` is the ground truth. The biggest omission was the FIR group delay and the distinction between processing latency vs. pulse-start latency.

Revised Honest Budget

```
Stage                    Cycles     Time      Confidence
──────────────────────────────────────────────────────────
DDC + CIC                  9        36 ns     medium
                                              (was 6 → optimistic)
Matched filter             8        32 ns     high
FIR group delay fill      16        64 ns     omitted in original!
  (startup, first pulse only)
Integrator pipeline        2         8 ns     high
  (window duration is T_pulse, not counted here)
MLP (hls4ml, synth TBD)  12–22    48–88 ns   medium
Output register            2         8 ns     high
──────────────────────────────────────────────────────────
Processing latency        33–43   132–172 ns  ✓ (after window ends)
Pulse-start latency       +T_pulse             depends on pulse length
```

The 200 ns budget is achievable for processing latency (from end of integration window to d ulse < ~60–70 ns , which is a tight but real constraint in superconducting qubit readout.



Total Resource Summary

Resource	Used	
DSP48E2	~200 (16-bit) / ~100 (8-bit)	5%
CLB LUTs	~600	
FFs	~1000	
BRAM	0 (weights in regs)	

The design is DSP-bound if using 16-bit precision. Using 8-bit QAT weights (which real hls4ml deployments use) cuts DSP count in half with minimal accuracy loss.

Bottom line on methodology: DDC and FIR estimates follow directly from DSP48E2 pipeline depth + adder tree depth, which are deterministic. The MLP estimate was an informed hls4ml heuristic — `hls_model.build(synth=True)` is the ground truth. The biggest omission was the FIR group delay and the distinction between processing latency vs. pulse-start latency.

Hardware - algorithm codesign

https://arxiv.org/abs/2501.05515
Tutorial at the Fast ML for Science 2025 in Sept!



Data representation
→ ML architecture

Neural architecture search/
Hyperparameter optimization

Physics requirements

Quantize network

Latency?
Pipeline Interval?

Intermediate (quantized)
representations

Resources? Area/power?
Radiation? Cryo?

Pruning/sparsity?

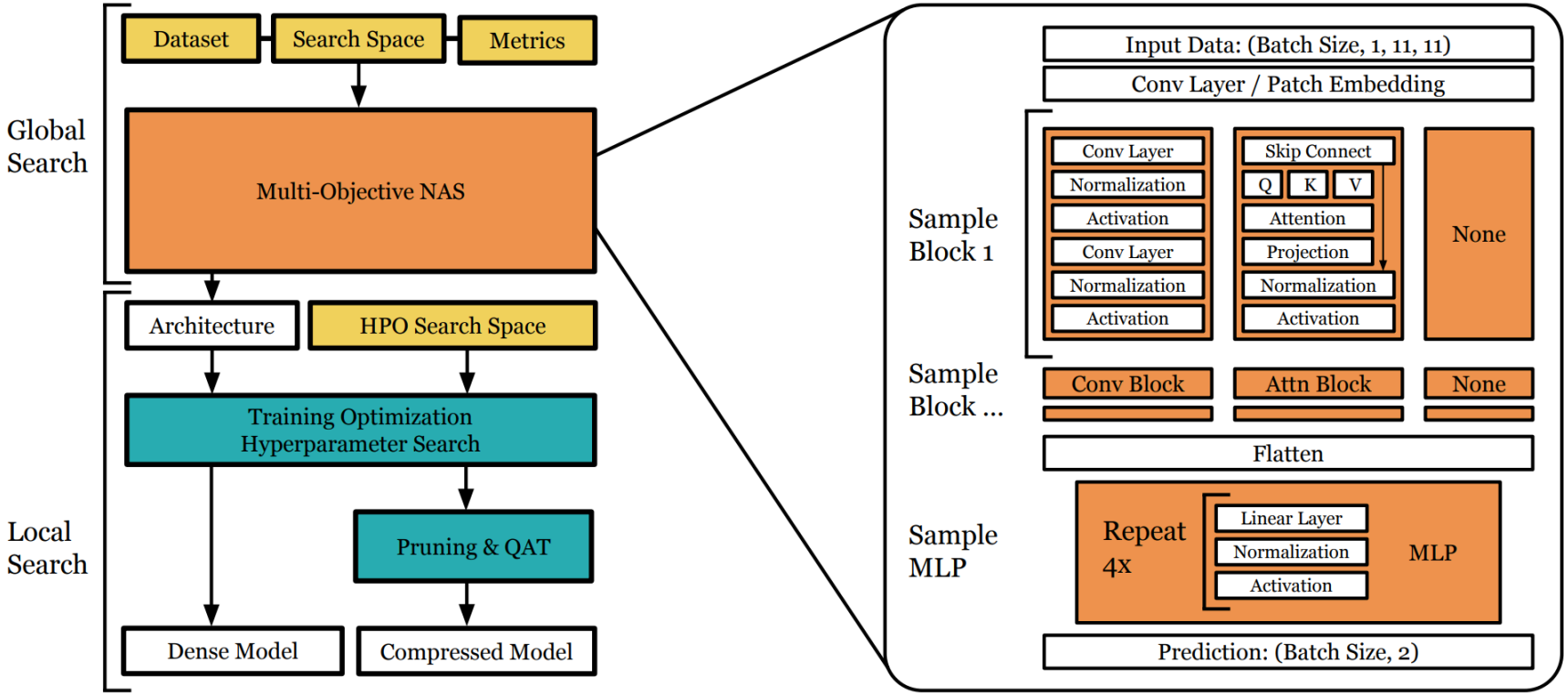
Microarchitecture

Synthesize, validate design,
satisfy design rules/timing

Parallelization

Super Neural Architecture Co-design Package (SNAC-Pack) powered with wa-hls4ml

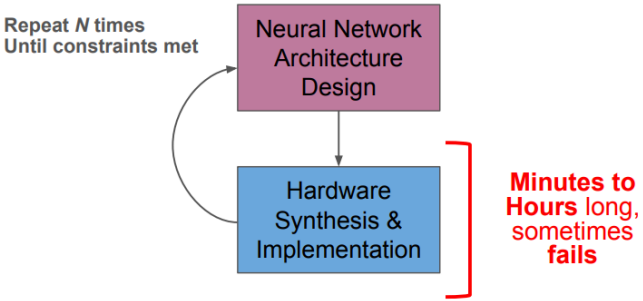
SNAC-pack powered by wa-hls4ml



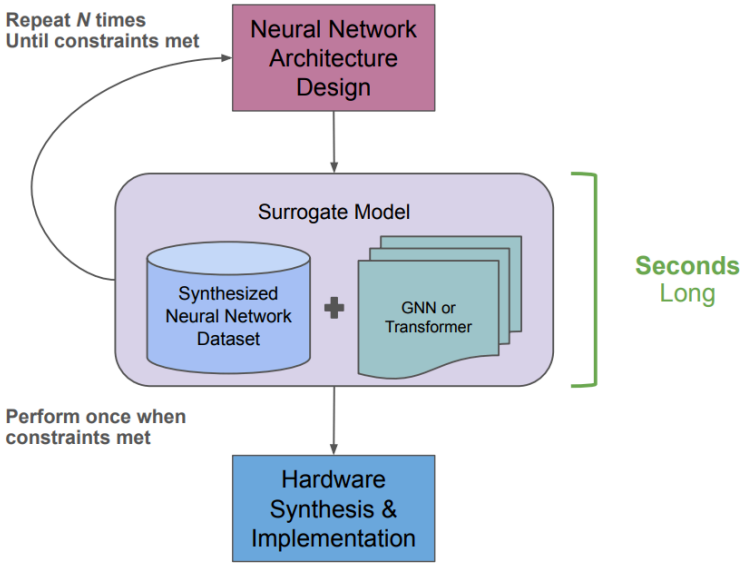
SNAC-pack powered by wa-hls4ml

SNAC-pack powered by wa-hls4ml

Traditional Codesign Workflow



Proposed Codesign Workflow



FERMI NATIONAL ACCELERATOR LABORATORY

Surrogate model

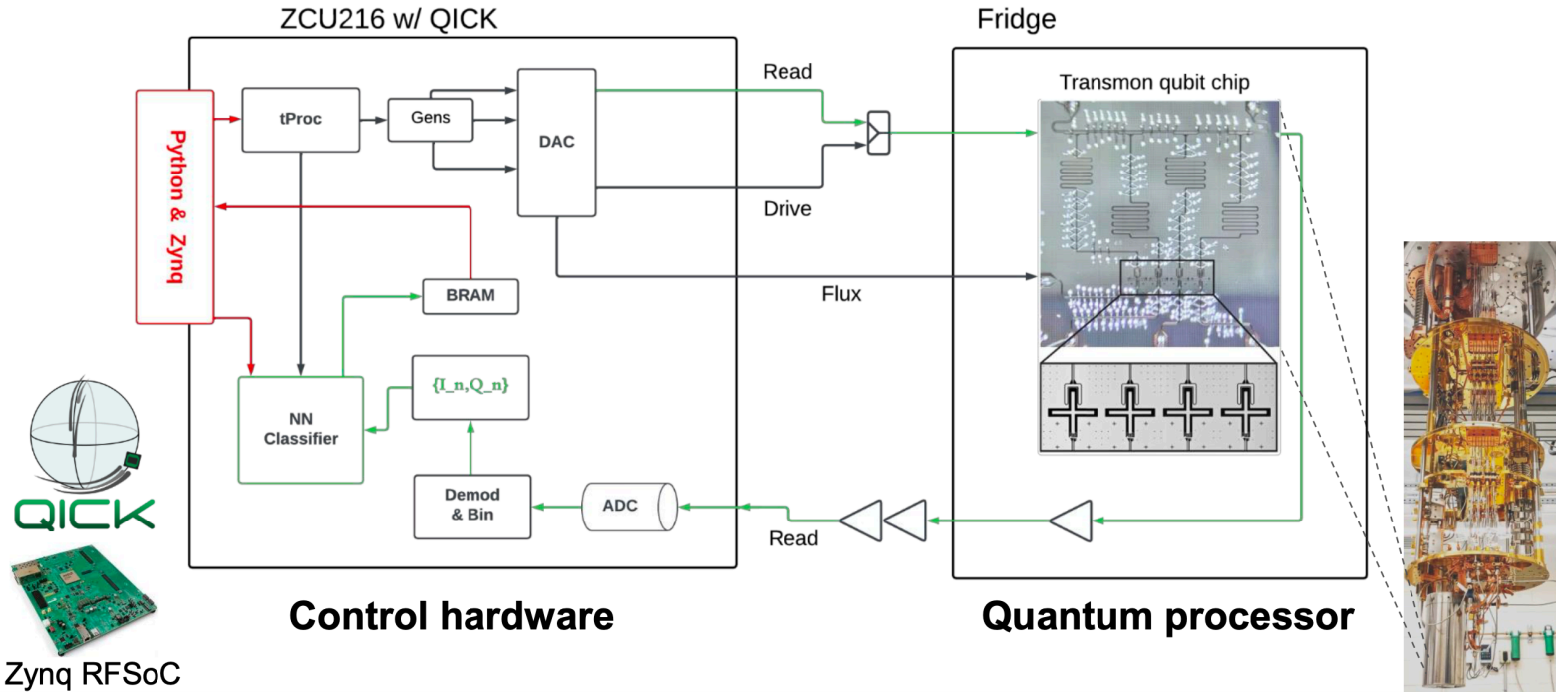
Table 1. Comparison of wa-hls4ml to prior work.

Tool	Open Source	Open Dataset	Dataset Size [samples]	Vendor tools required	Is Bench-mark	Input Abstraction Level
Native HLS Estimate [3]	No	N/A	N/A	Yes	N/A	HLS Code
High-Level Synthesis Performance Prediction using GNNs [38]	**Yes**	**Yes**	40, 000	Yes	**Yes**	HLS/LLVM IR Graph
Machine Learning Aided Hardware Resource Estimation for FPGA DNN Implementation [14]	No	No	N/A	**No**	No	FINN Intermediate Representation
A Graph Neural Network Model for Fast and Accurate Quality of Result Estimation for High-Level Synthesis [22]	No	No	2, 465	Yes	No	HLS/LLVM IR Graph
HLSyn [5]	**Yes**	**Yes**	42, 000	**No**	**Yes**	HLS Code
Fast and Accurate Estimation of Quality of Results in High-Level Synthesis with Machine Learning [13]	No	No	1, 300	Yes	No	HLS Reports
rule4ml [30] (Related Work)	**Yes**	**Yes**	15, 000	**No**	No	hls4ml IR
wa-hls4ml (This work)	**Yes**	**Yes**	683, 176	**No**	**Yes**	hls4ml IR

ML-based qubit readout

https://arxiv.org/abs/2501.14663

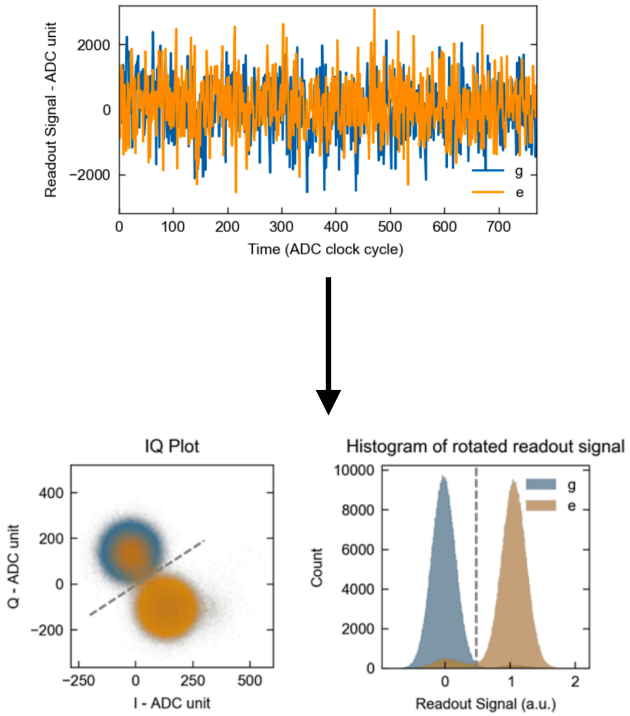
PURDUE UNIVERSITY Fermilab QUANTUM SCIENCE CENTER

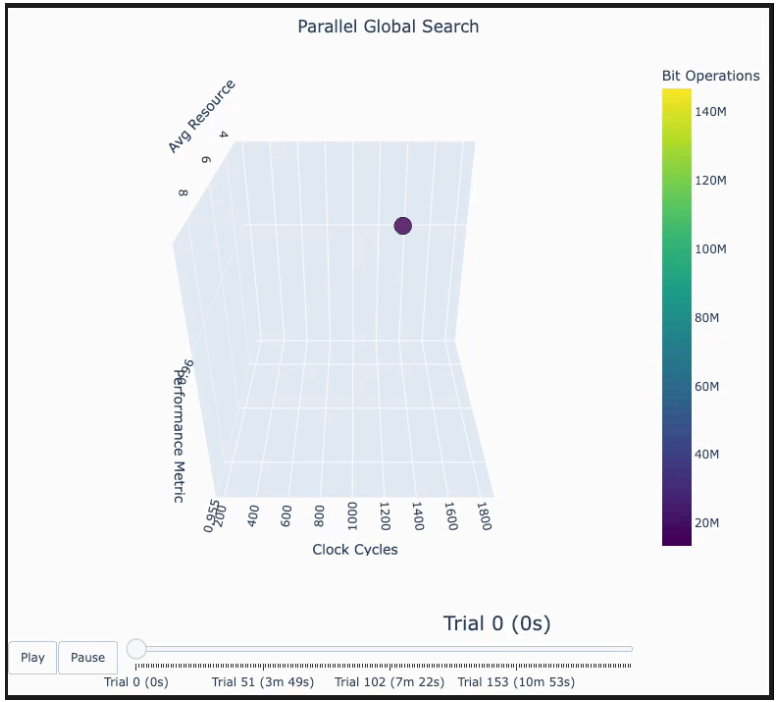


Readout frequencies in the RF, ~few GHz
Control latency < 1 μs

FERMI NATIONAL ACCELERATOR LABORATORY

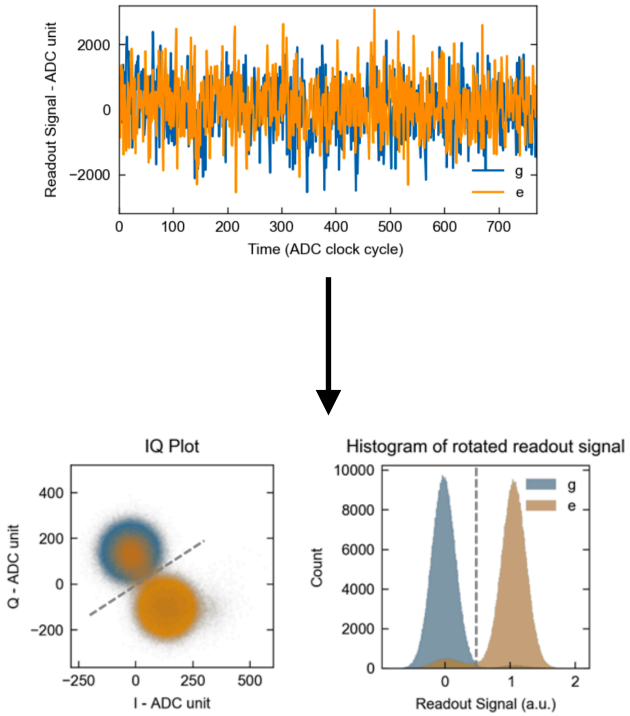
Qubit codesign with SNAC-pack at NERSC

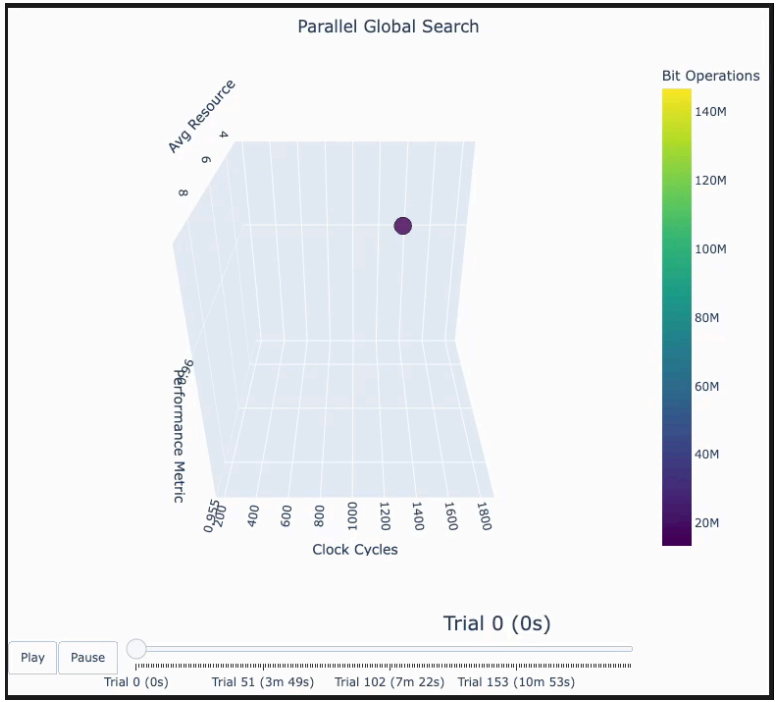




FERMI NATIONAL ACCELERATOR LABORATORY

Qubit codesign with SNAC-pack at NERSC





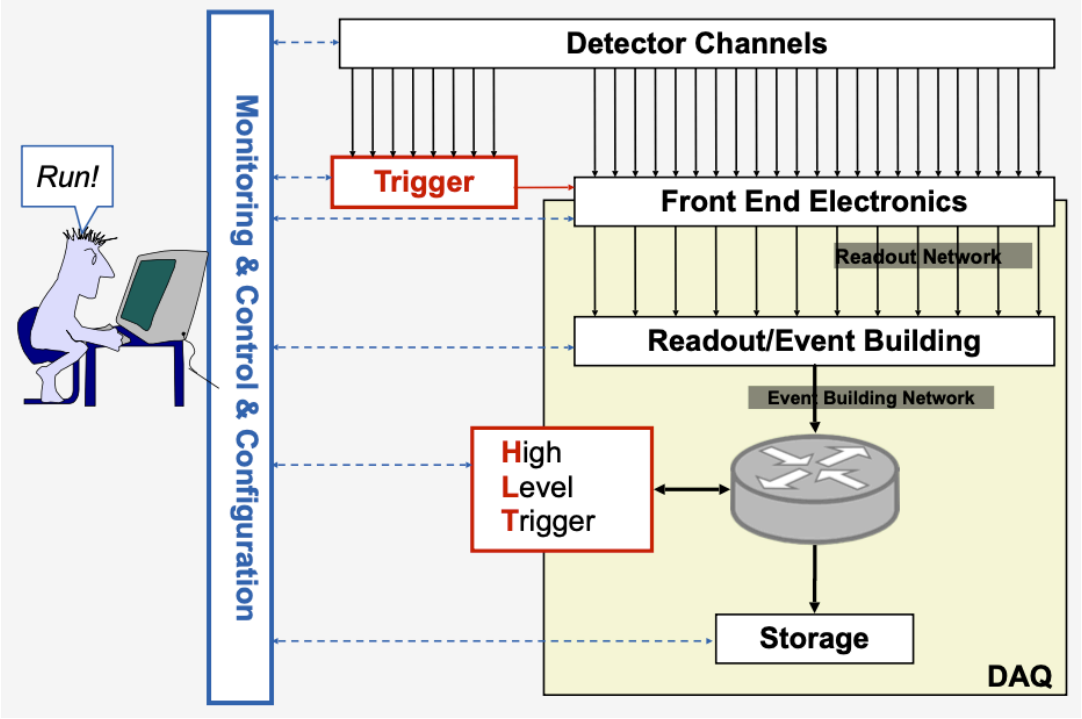
FERMI NATIONAL ACCELERATOR LABORATORY

Outline

- Fast Control
- Fast Neural Control
- **Fast Adaptive Neural Control**

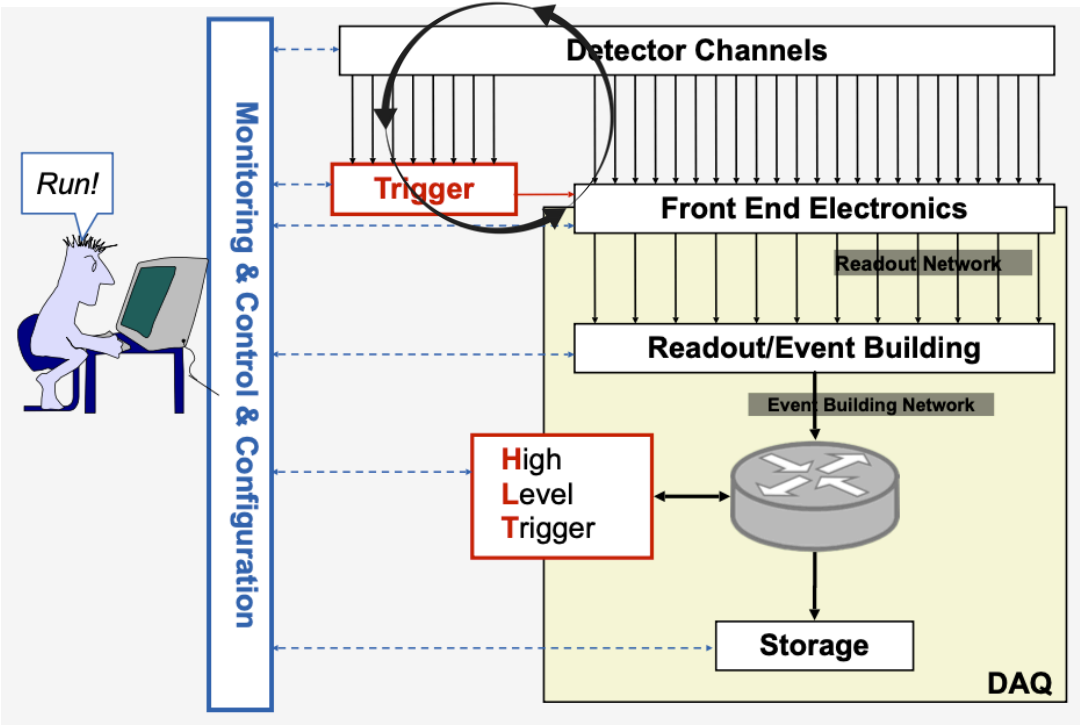
Fast and slow control

Image credit: A. Thea



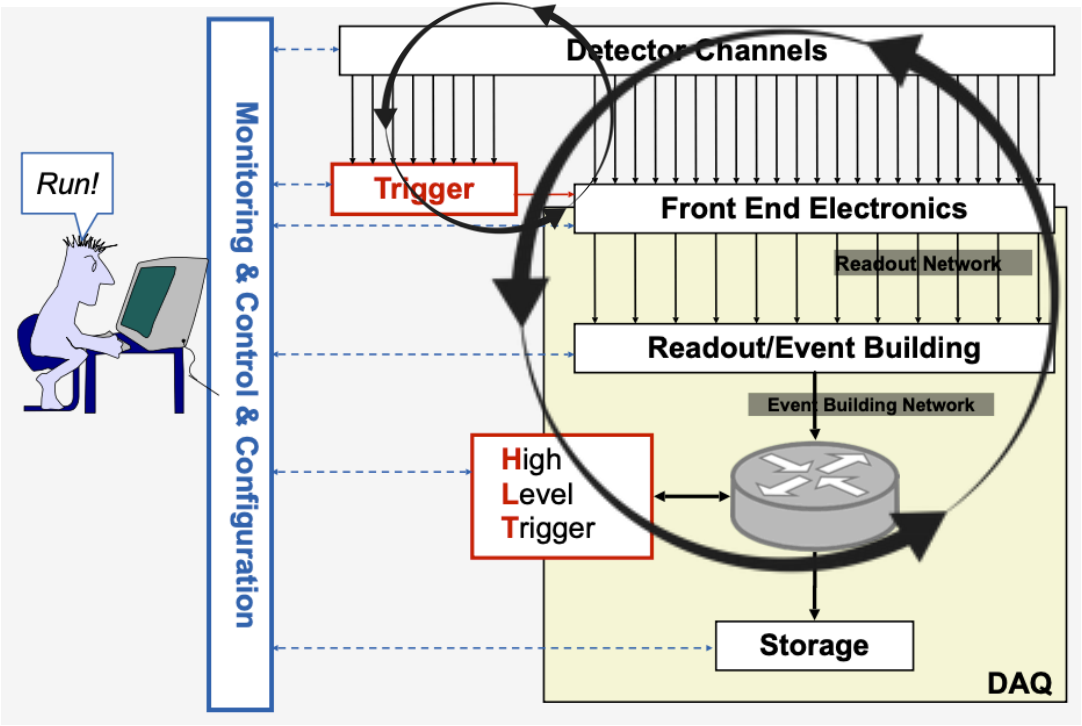
Fast and slow control

Image credit: A. Thea



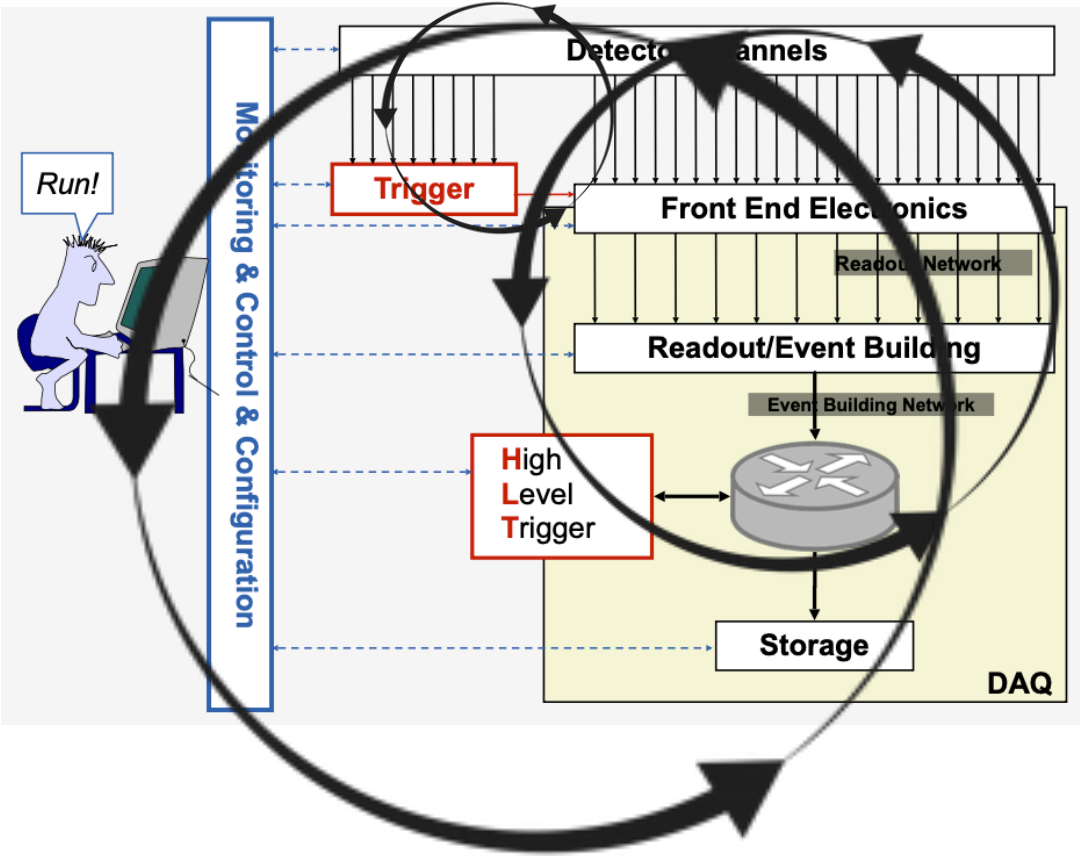
Fast and slow control

Image credit: A. Thea



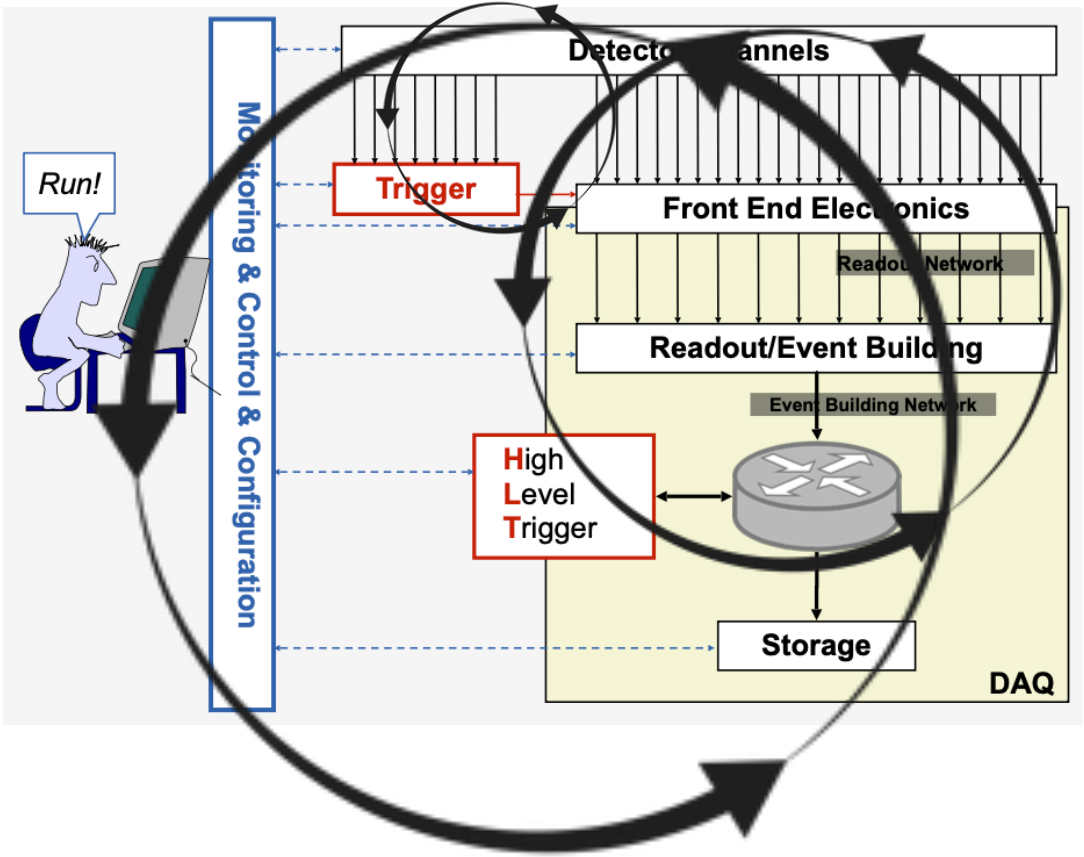
Fast and slow control

Image credit: A. Thea

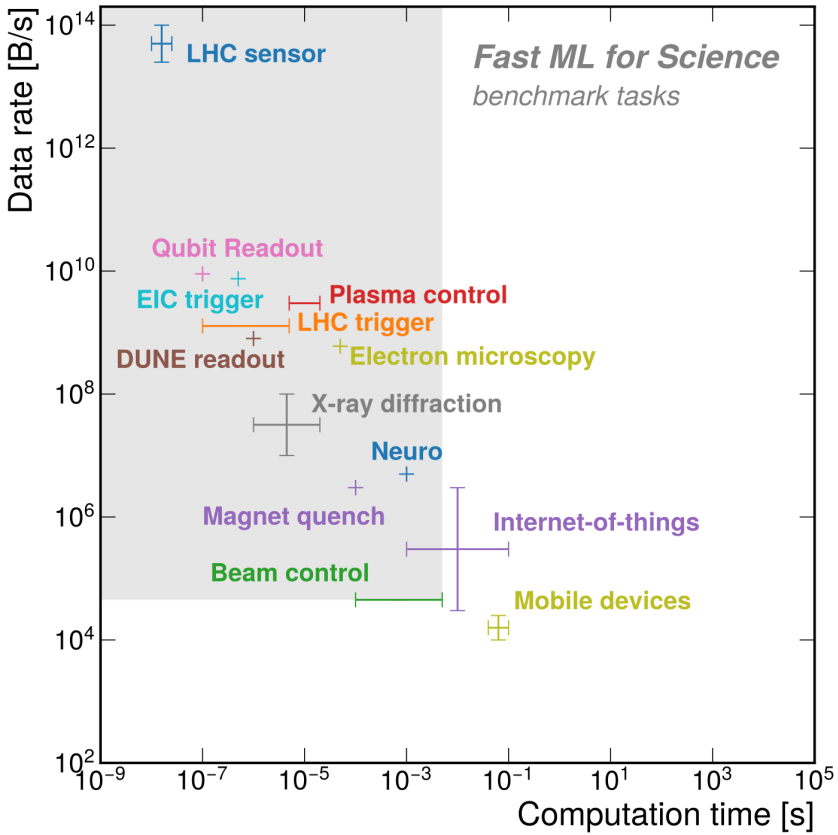


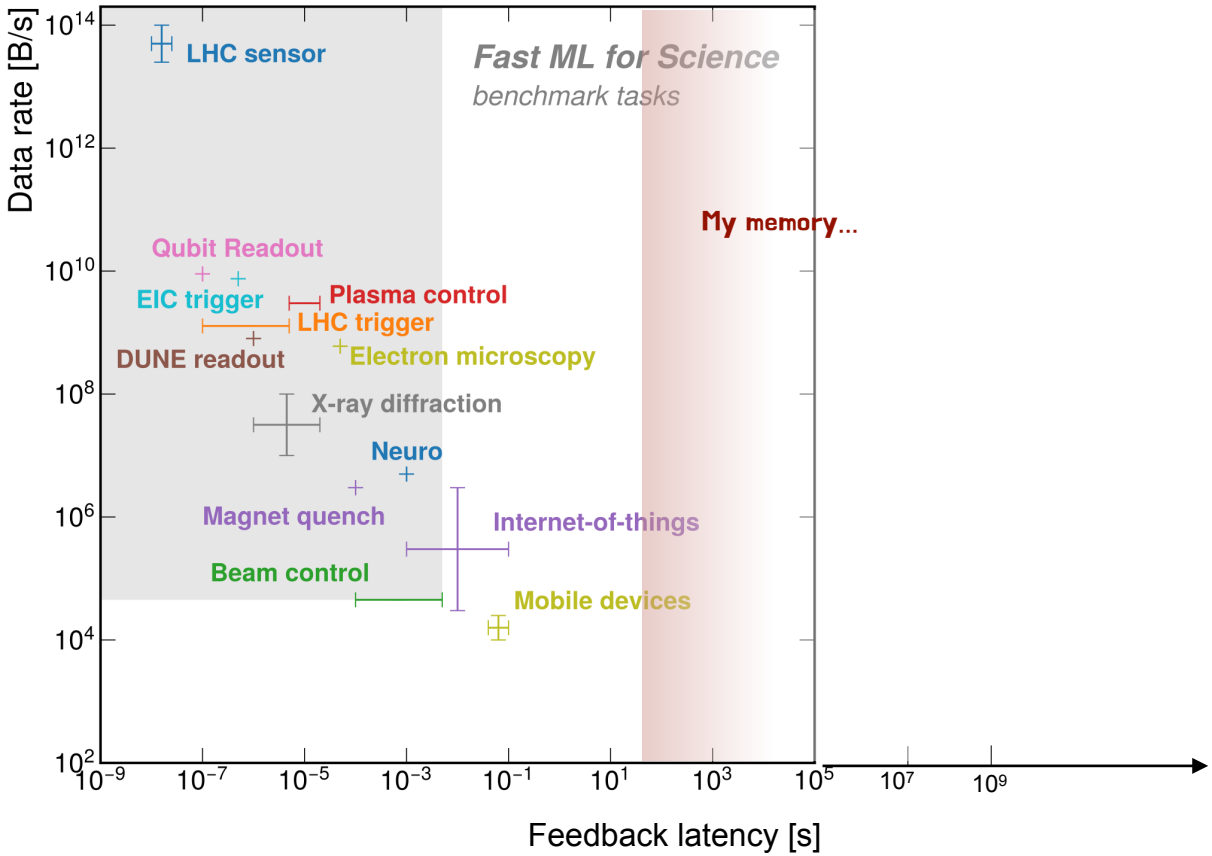
Fast and slow control

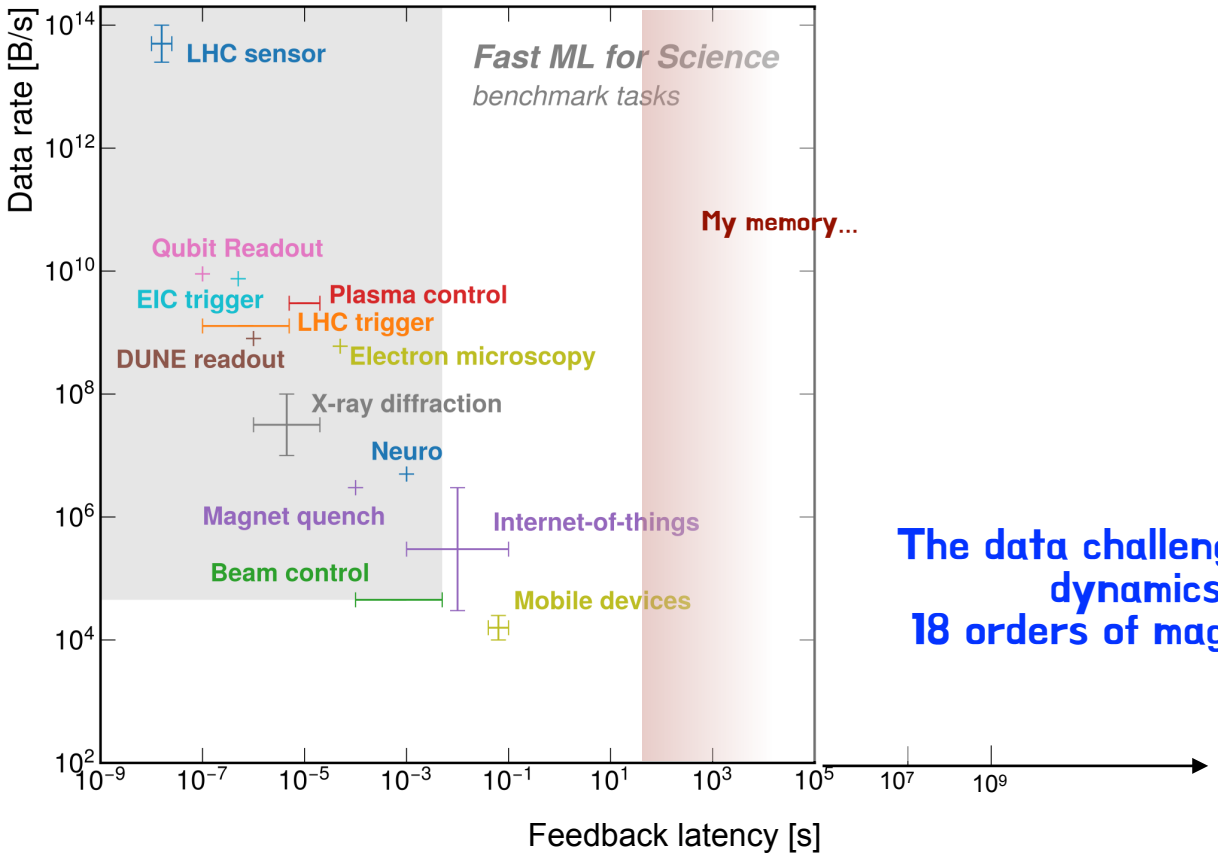
Image credit: A. Thea



- Fast control
  - Immediate response to dynamics of the experiment and data readout
  - Event timing, triggering, etc.
- Slow control
  - Detector stability over minutes, days, weeks, months,…
  - Monitoring and controlling operational parameters: electronics gains, pedestals, calibrations, etc.

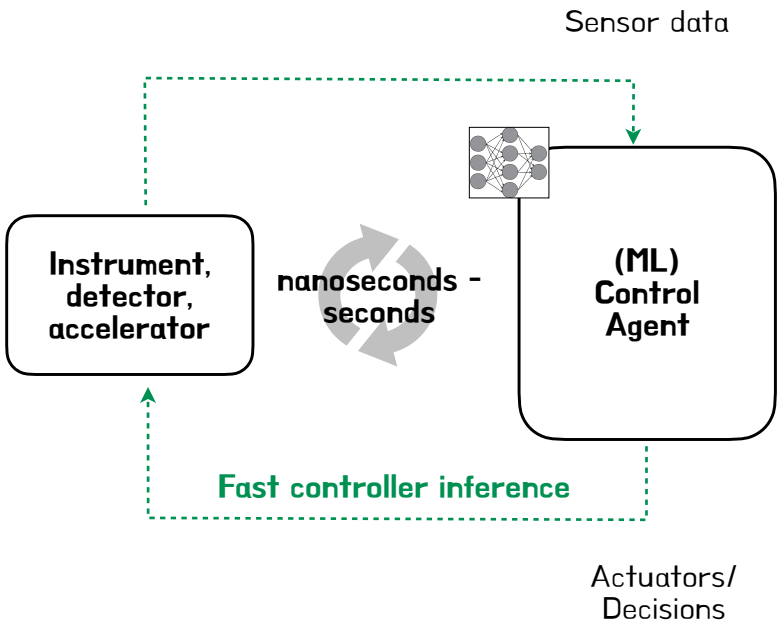

Fast ML for Science
benchmark tasks
LHC sensor
Qubit Readout
EIC trigger
Plasma control
LHC trigger
DUNE readout
Electron microscopy
X-ray diffraction
Neuro
Magnet quench
Internet-of-things
Beam control
Mobile devices
Data rate [B/s]
Computation time [s]
10^{14}
10^{12}
10^{10}
10^{8}
10^{6}
10^{4}
10^{2}
10^{-9}
10^{-7}
10^{-5}
10^{-3}
10^{-1}
10^{1}
10^{3}
10^{5}


Data rate [B/s]
10^{14}
10^{12}
10^{10}
10^{8}
10^{6}
10^{4}
10^{2}
LHC sensor
Fast ML for Science
benchmark tasks
My memory...
Qubit Readout
EIC trigger
Plasma control
LHC trigger
DUNE readout
Electron microscopy
X-ray diffraction
Neuro
Magnet quench
Internet-of-things
Beam control
Mobile devices
10^{-9}
10^{-7}
10^{-5}
10^{-3}
10^{-1}
10^{1}
10^{3}
10^{5}
10^{7}
10^{9}
Feedback latency [s]

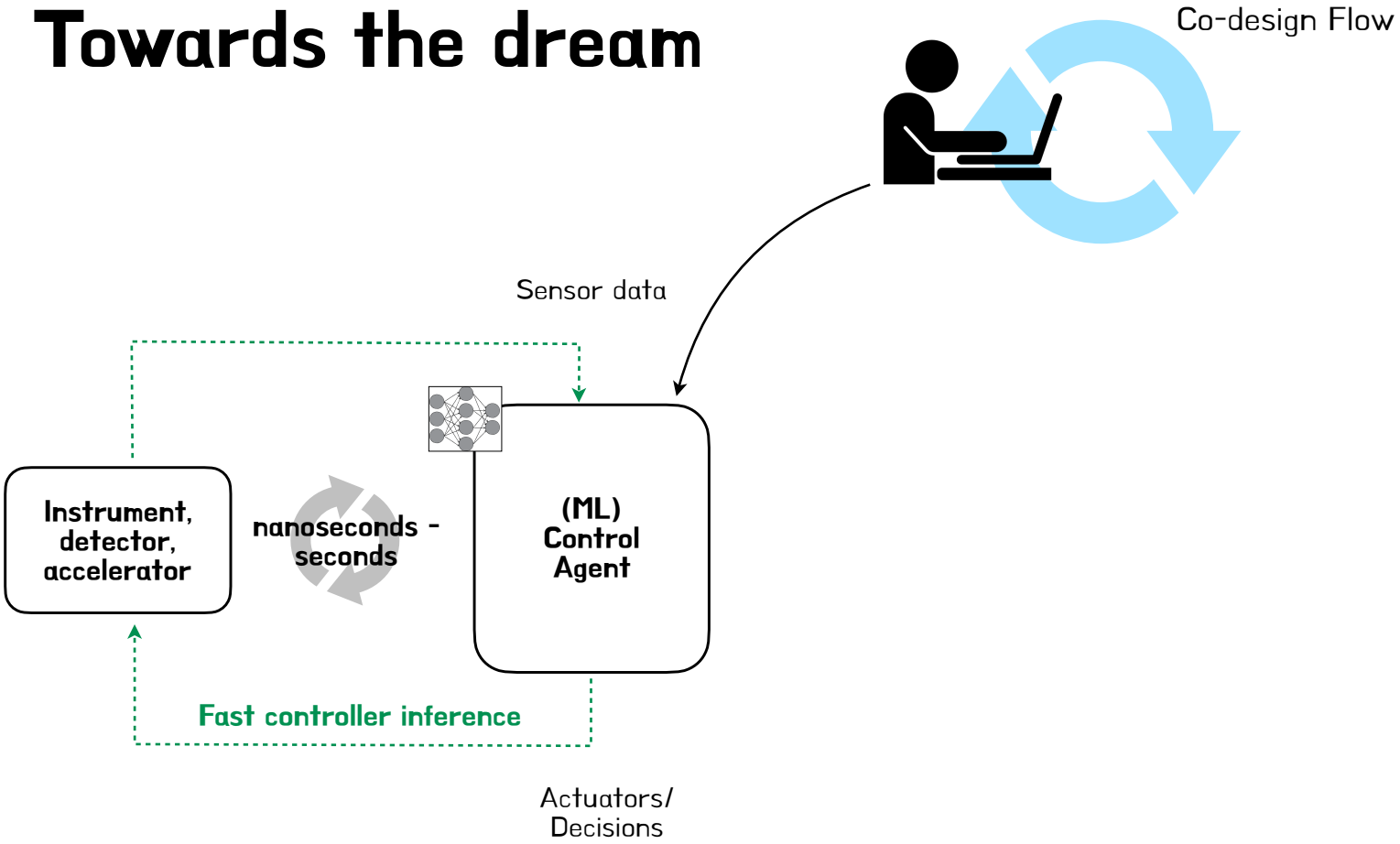


The data challenge is to capture dynamics across 18 orders of magnitude in time!

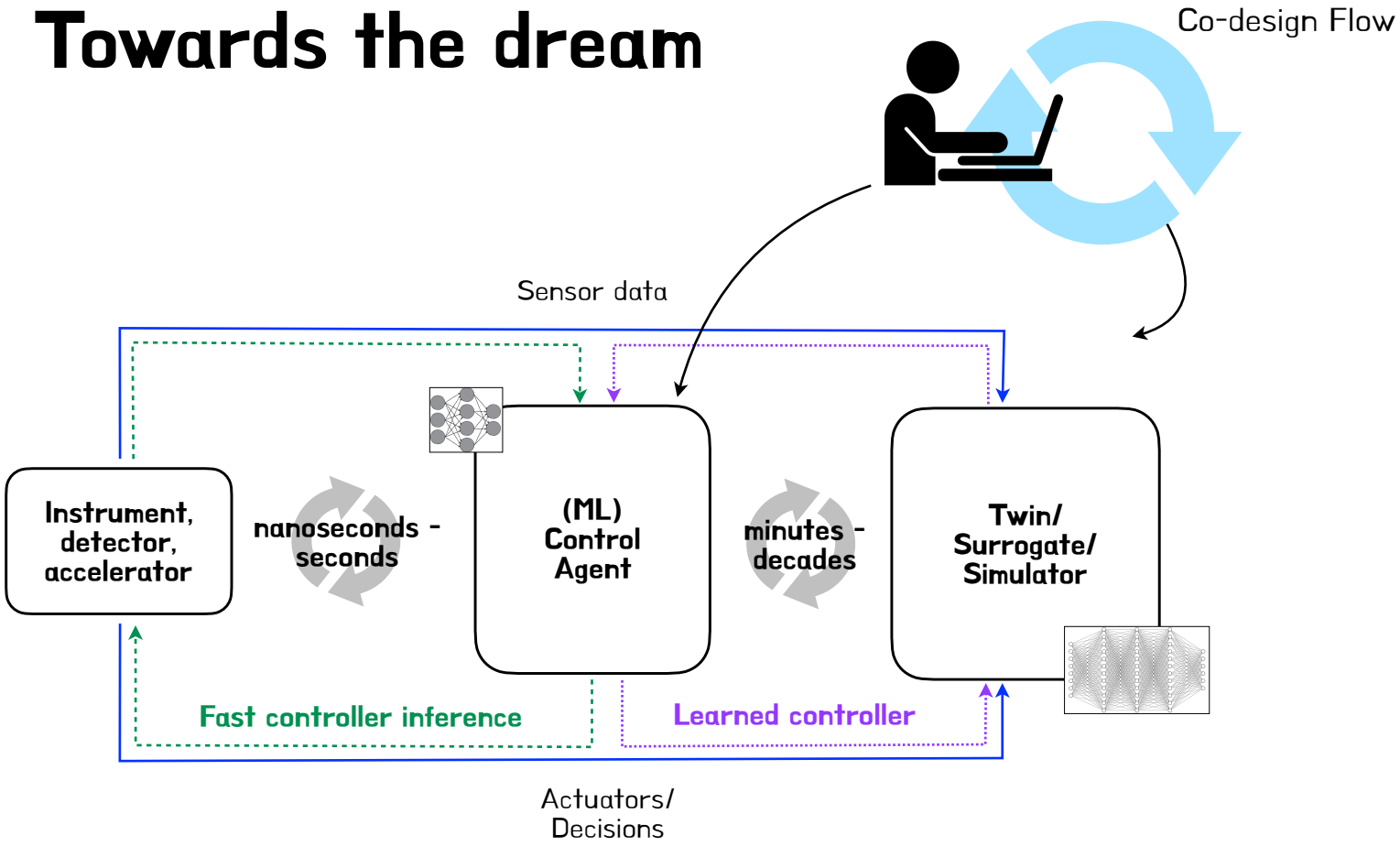
Towards the dream

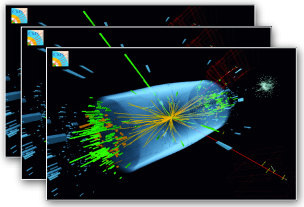


Towards the dream



Towards the dream

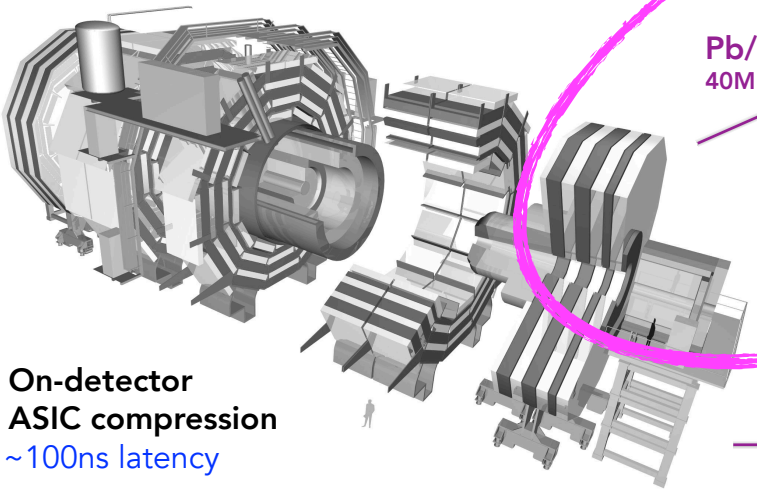




CMS Experiment

40MHz collision rate

~1B detector channels

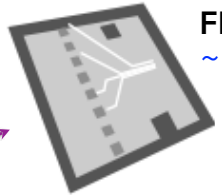


On-detector ASIC compression

~100ns latency

Pb/s

40MHz

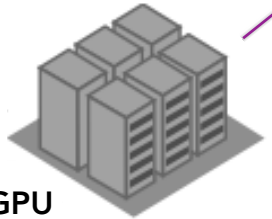


FPGA filter stack

~us latency

10s Tb/s

100s kHz

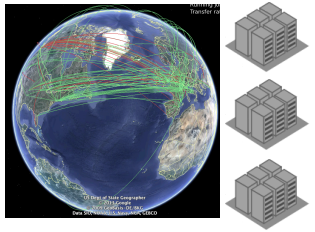


On-prem CPU/GPU filter farm

~100 ms latency

10s Gb/s

~5 kHz



Worldwide computing grid

Exabyte-scale datasets

credit: David Miller

The Hidden Cost: Operational Burden

Despite how we might talk about it, the trigger menu is not a one-time design, it is a living system requiring continuous expert attention throughout every run, every fill, and every luminosity evolution.

Trigger Menu Coordinators	A dedicated team designs, validates, and maintains hundreds of trigger chains — a multi-month effort per data-taking period.
On-Call Trigger Experts	Round-the-clock expert coverage to monitor rates, respond to unexpected changes, and authorize manual adjustments during fills.
Shift Crew Overhead	Shifters must recognize trigger anomalies and interpret rate plots — a steep, recurring learning curve for every new crew.
Prescale Adjustment Cycles	Every luminosity change may need manual prescale updates to stay within bandwidth — repeated many times per fill, every fill.
Menu Validation Campaigns	New algorithms require extensive offline validation: rate estimation, efficiency checks, and multi-group sign-off before any deployment.
Knowledge Concentration Risk	Critical know-how lives in few experts. Attrition and long shutdown periods create dangerous institutional memory gaps.



What if the data processing and reduction pipeline could continuously learn to determine what data to save on its own?



IOP Publishing Mach. Learn.: Sci. Technol. 7 (2026) 030501 https://doi.org/10.1088/2632-2153/ae631f

MACHINE LEARNING
Science and Technology

Check for updates

OPEN ACCESS

RECEIVED
23 January 2026

REVISED
30 March 2026

ACCEPTED FOR PUBLICATION
20 April 2026

PUBLISHED
11 May 2026

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

BENCHMARK

Towards a self-driving trigger at the LHC: adaptive response in real time

Shaghayegh Emami[1,*], Cecilia Tosciri[2], Giovanna Salvi[1], Zixin Ding[3], Yuxin Chen[3], Abhijith Gandrakota[4], Christian Herwig[4], David W Miller[2,5], Jennifer Ngadiuba[4] and Nhan Tran[4]

[1] University of Michigan, Ann Arbor, MI 48109, United States of America
[2] Enrico Fermi Institute, University of Chicago, Chicago, IL 60637, United States of America
[3] Department of Computer Science, University of Chicago, Chicago, IL 60637, United States of America
[4] Fermi National Accelerator Laboratory, Batavia, IL 60510, United States of America
[5] Department of Physics, Kavli Institute for Cosmological Physics, University of Chicago, Chicago, IL 60637, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: shghygh@umich.edu

Keywords: LHC trigger system, real-time control, anomaly detection, CMS open data

Abstract

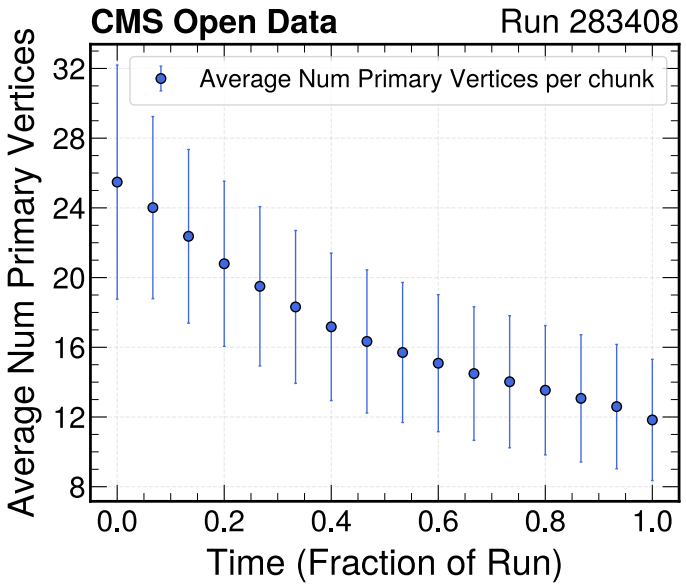
Real-time data filtering and selection—or *trigger*—systems at high-throughput scientific facilities such as the experiments at the Large Hadron Collider must process extremely high-rate data streams under stringent bandwidth, latency, and storage constraints. Yet these systems are typically designed as static, hand-tuned menus of selection criteria grounded in prior knowledge and simulation. In this work, we further explore the concept of a *self-driving trigger*, an autonomous data-filtering framework that reallocates resources and adjusts thresholds dynamically in real-time to optimize signal efficiency, rate stability, and computational cost as instrumentation and environmental conditions evolve. We introduce a benchmark ecosystem to emulate realistic collider scenarios and demonstrate real-time optimization of a menu including canonical energy sum triggers as well as modern anomaly-detection algorithms that target non-standard event topologies using machine learning. Using simulated data streams and publicly available collision data from the Compact Muon Solenoid experiment, we demonstrate the capability to dynamically and automatically optimize trigger performance under specific cost objectives without manual retuning. Our adaptive strategy shifts trigger design from static menus with heuristic tuning to intelligent, automated, data-driven control, unlocking greater flexibility and discovery potential in future high-energy physics analyses.

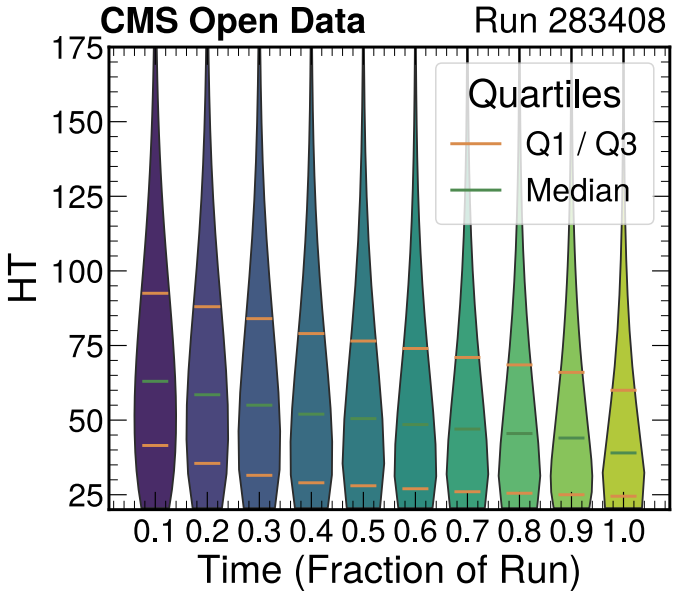
1. Introduction

In scientific domains characterized by high data throughput, algorithms tasked with filtering and selecting data for storage and analysis—frequently referred to as *trigger algorithms*—must operate under

https://iopscience.iop.org/article/10.1088/2632-2153/ae631f/pdf
LinkedIn Post!

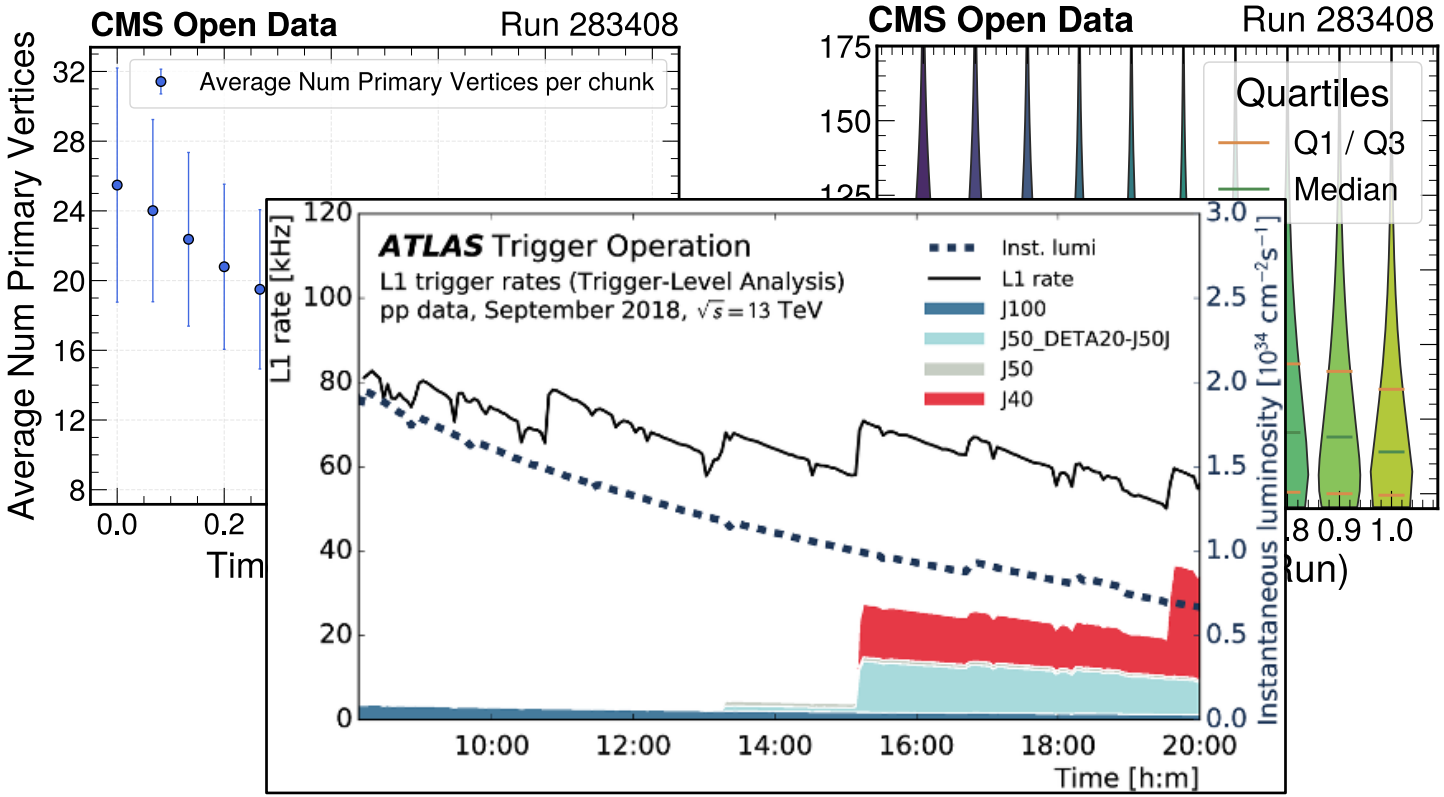
Dynamic system



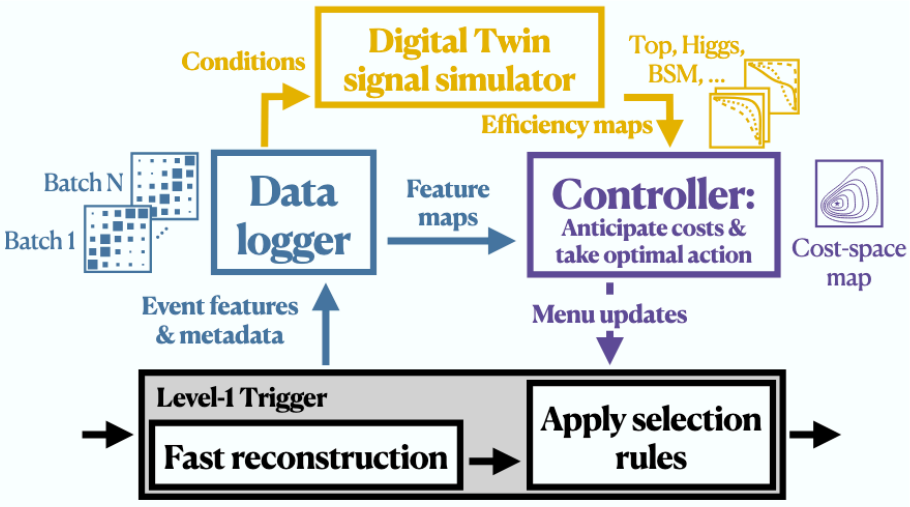


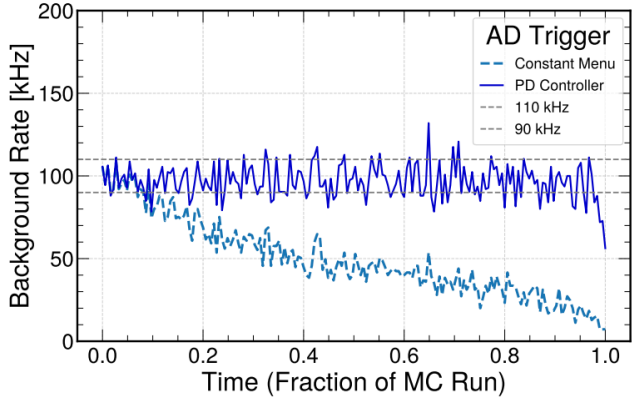
FERMI NATIONAL ACCELERATOR LABORATORY

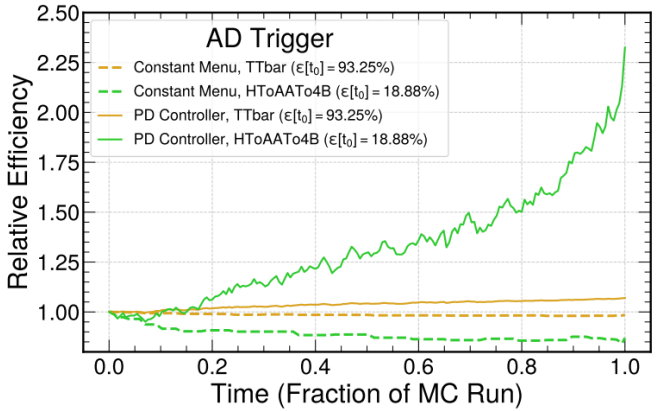
Dynamic system



Towards a self-driving LHC trigger







Setup:

- MinBias
- Ttbar
- H > aa > bbbb

Trigger paths

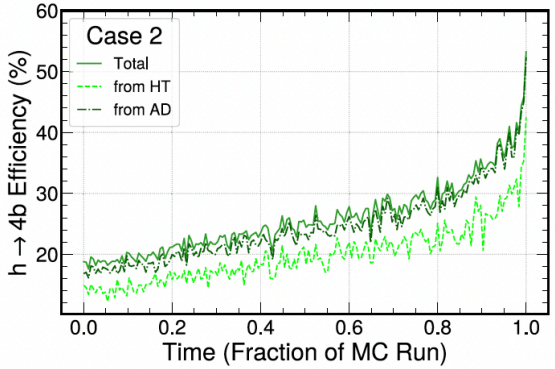
- HT
- Anomaly Detection (AD)

Towards a self-driving LHC trigger

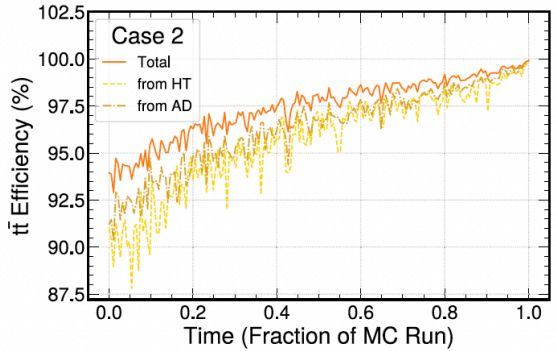
Multi-trigger scenarios:

- Case 1: Fixed background
- Case 2: Allocate exploratory path without sacrificing ttbar
- Case 3: Considering computational cost

$$C_2 = \frac{|r_b - r_t|}{\sigma_b} + \frac{|\epsilon_{t\bar{t}} - 0.9|}{\sigma_{t\bar{t}}} + \frac{|r_{AD}^{(ex)} - p \cdot r_t|}{\sigma_{AD}}$$

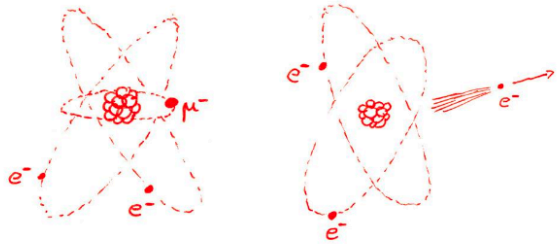


(c) BSM signal $h \to 4b$ efficiency breakdown by path, showing increased AD contribution due to bandwidth allocation.



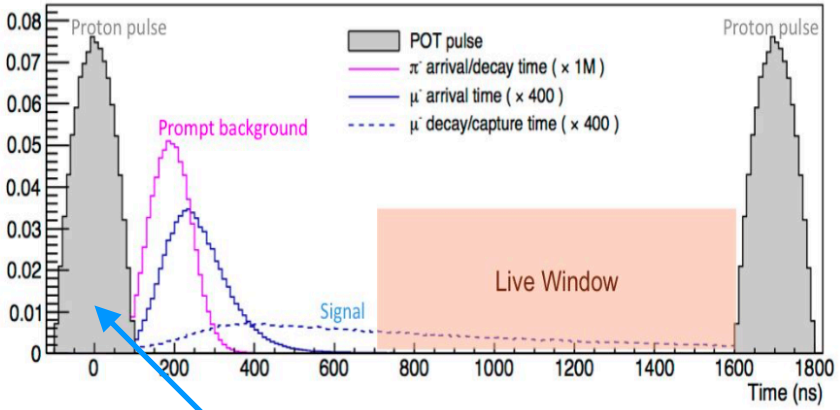
(d) $t\bar{t}$ signal rate by path, dominated by AD but nearly failing the 90% efficiency criterion imposed by the cost.

Application example: slow spill regulation for mu2e



μ^- captured inside Al atom

μ^- converting directly to e^-



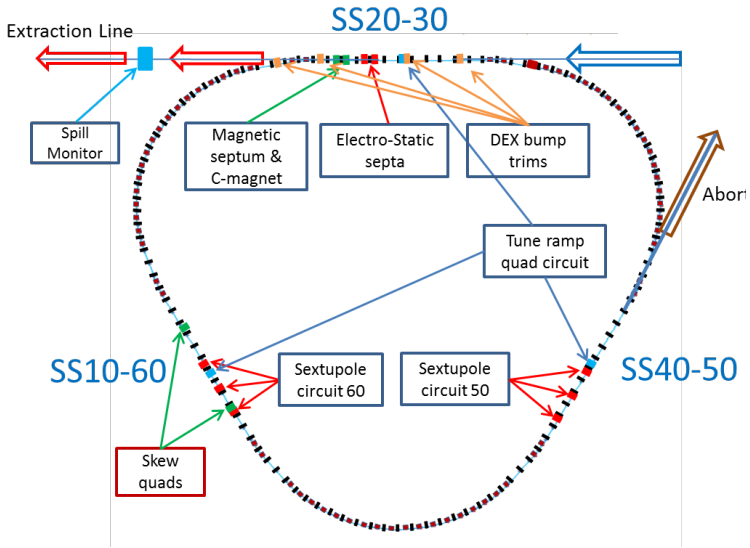
Detect: $R_{\mu^- \to e^-} = \dfrac{\mu^- N \to e^- N}{\mu^- N \to all\ muon\ captures}$

Pulsed beam required to avoid backgrounds.

As uniform as possible

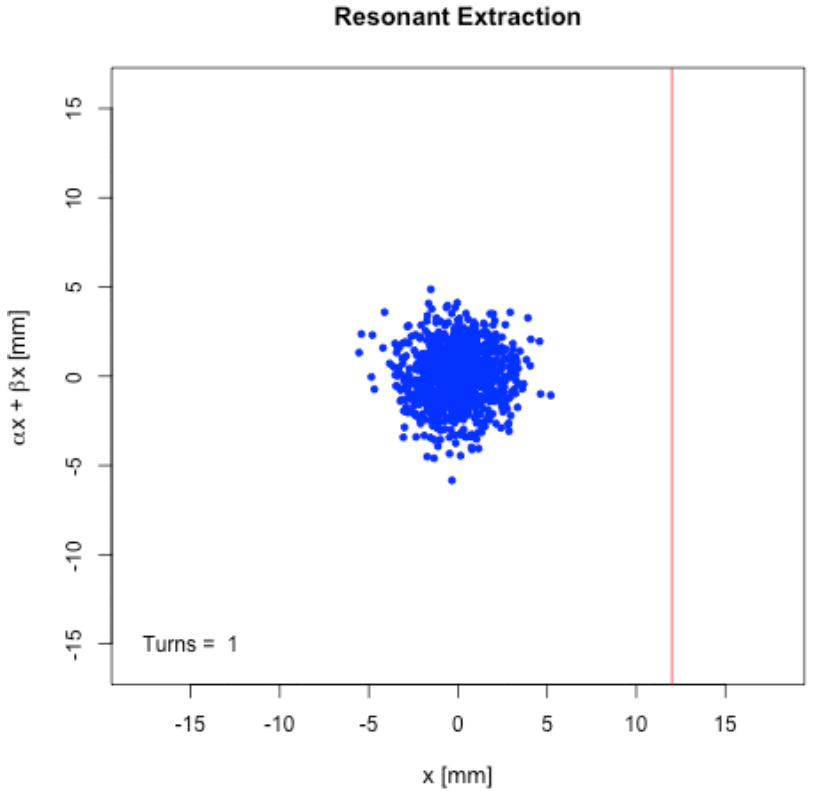
Beam control - resonant extraction

Credit: Aakaash Narayan



$$SDF = \frac{1}{1 + \sigma^2_{\{ext.rate\}}}$$

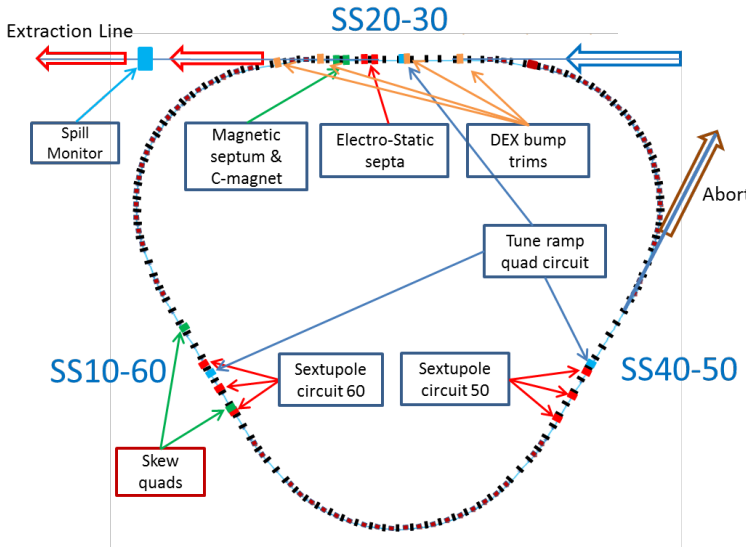
🎯 **Target SDF > 0.6**



FERMI NATIONAL ACCELERATOR LABORATORY

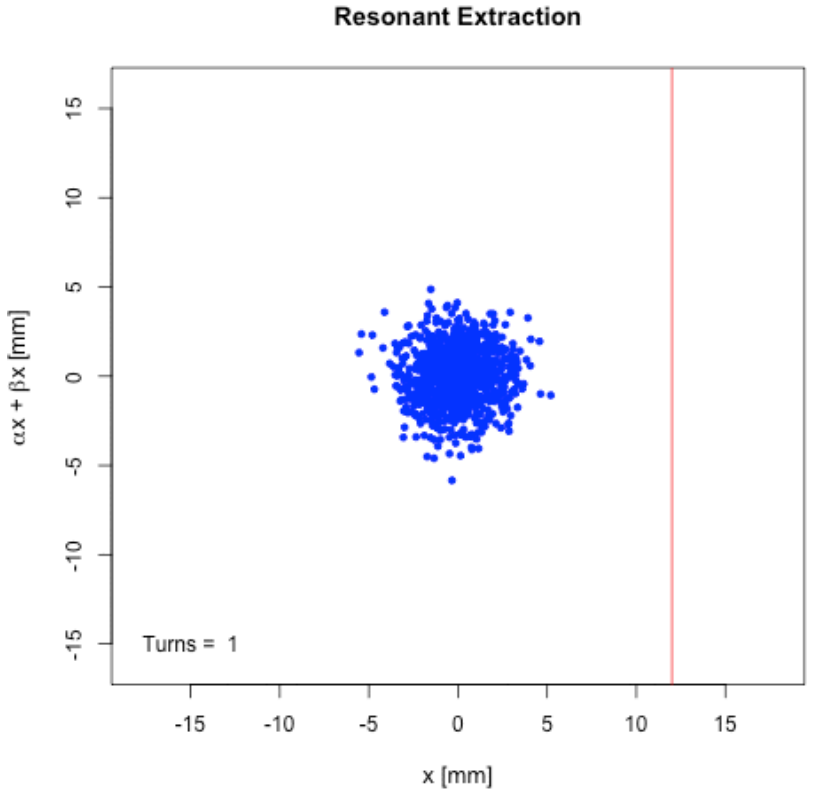
Beam control - resonant extraction

Credit: Aakaash Narayan



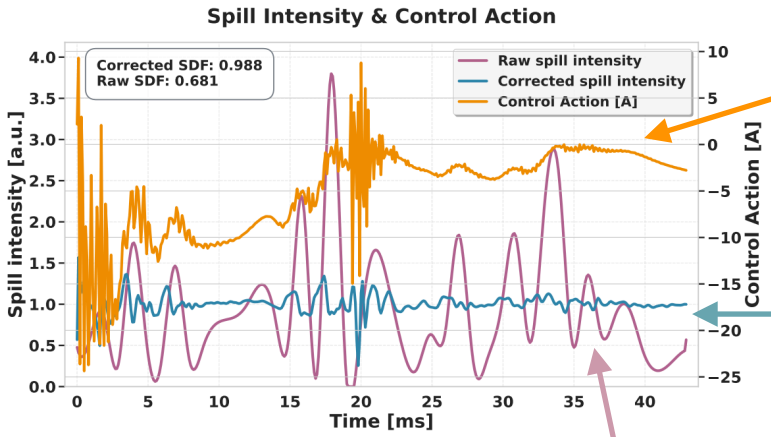
$$SDF = \frac{1}{1 + \sigma^2_{\{ext.rate\}}}$$

🎯 **Target SDF > 0.6**



FERMI NATIONAL ACCELERATOR LABORATORY

Beam control - resonant extraction



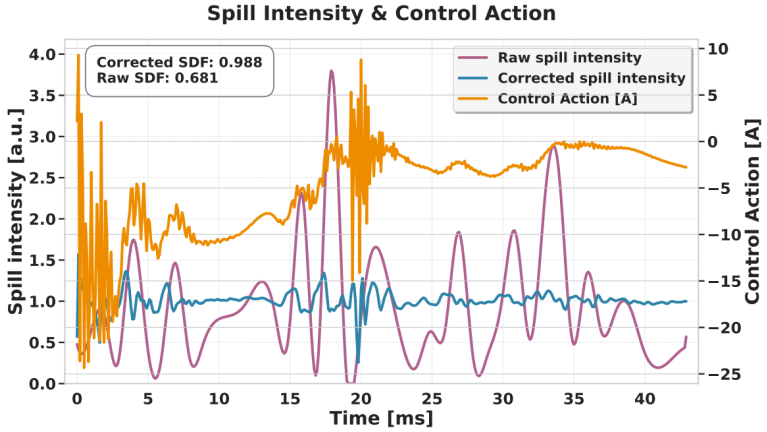
Control actions

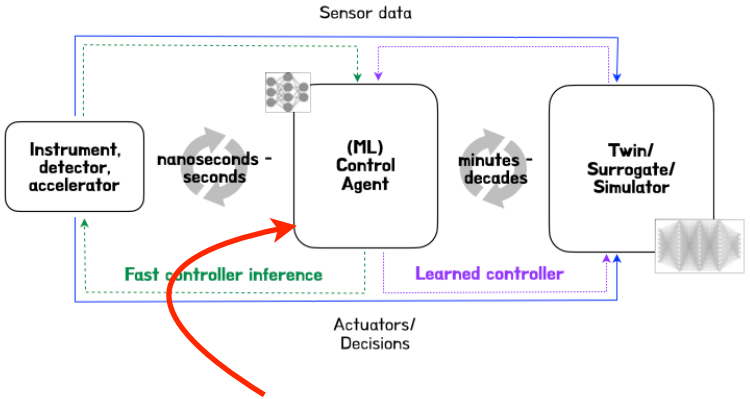
Corrected spill

Raw spill

https://arxiv.org/pdf/2511.19099

Beam control - resonant extraction

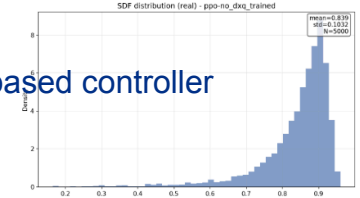


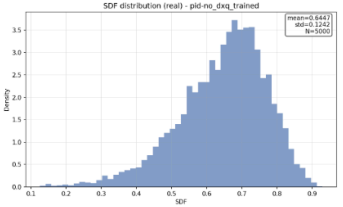


RL Controller (0.839)

PID Controller (0.645)

GRU-based controller

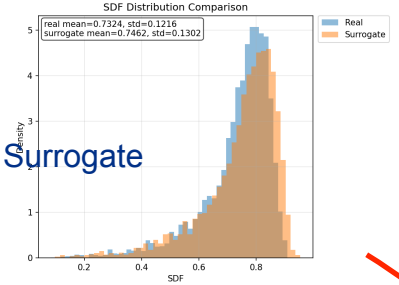


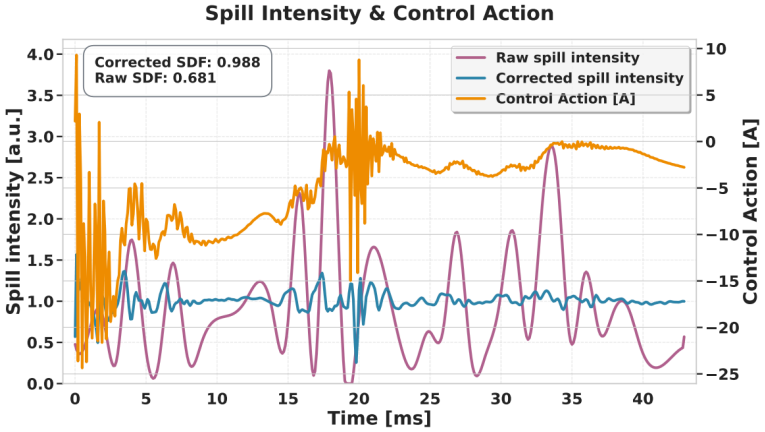


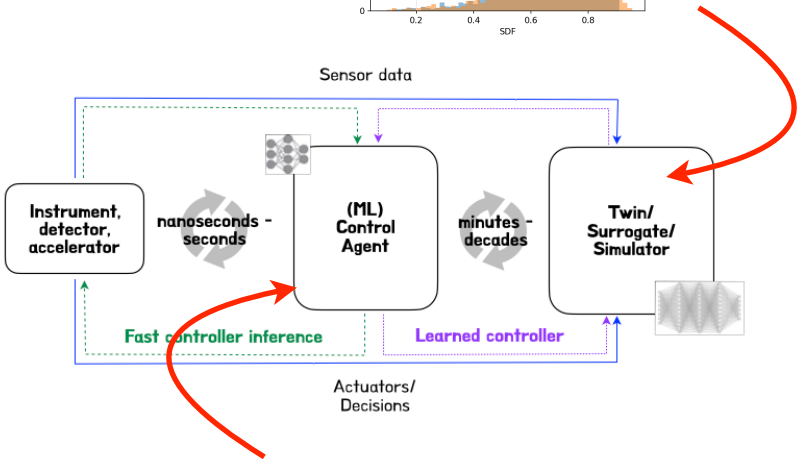
https://arxiv.org/pdf/2511.19099

Beam control - resonant extraction

Diffusion + GRU for Surrogate



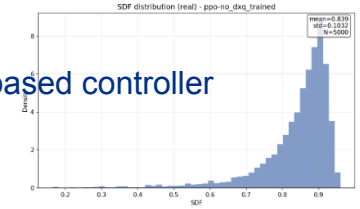


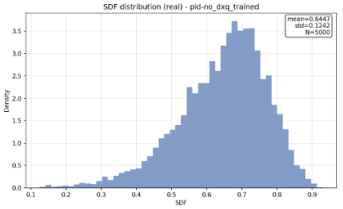


RL Controller (0.839)

PID Controller (0.645)

GRU-based controller



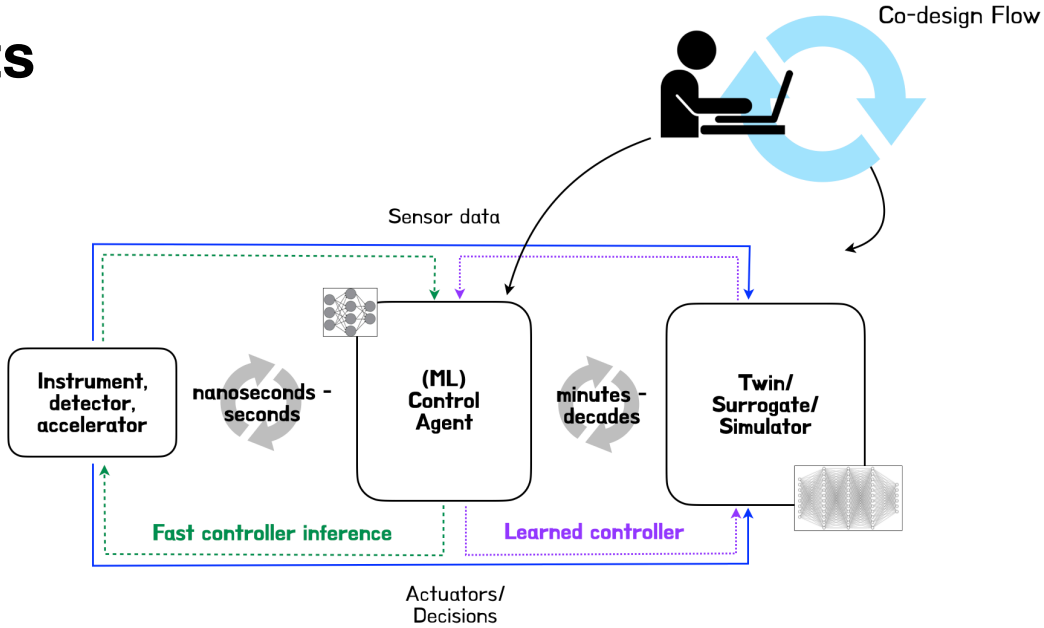


https://arxiv.org/pdf/2511.19099



Outlook

FAN-C Feats



- Edge AI and hardware codesign
  - Enable fundamentally new capabilities & robust adaptive systems
  - 500x codesign acceleration through surrogate-guided design space exploration
- Fast and Slow together
  - Towards powerful autonomous instruments enabled through physics-inspired surrogate models and robust, real-time controllers