Fast Machine Learning for Science Conference 2025

Monday 1 September 2025 - Friday 5 September 2025

ETH Zurich

Book of Abstracts

Contents

hls4ml . . . 1

Registration . . . 1

hls4ml tutorial . . . 1

Tutorial 2 . . . 1

Coyote v2: Open-source Abstractions and Infrastructure for FPGAs . . . 2

Tutorial 4 . . . 2

Tutorial 5 . . . 2

Tutorial 6 . . . 2

Registration . . . 2

Reasoning Language Models: Overview and Blueprint . . . 3

Real-time mixed-signal electronic circuits for understanding and implementing neural computation . . . 3

The JAX scientific ecosystem . . . 3

Real-time inference at the LHC . . . 3

Real-time inference in gravitational wave astronomy (REMOTE) . . . 4

Welcome . . . 4

Poster Session and Coffee . . . 4

Fast ML foundation discussion . . . 4

Coyote v2: Open-source Abstractions and Infrastructure for FPGAs . . . 4

Optimizing Reciprocal Human-Machine Learning Decisions . . . 5

Radiation-Hard, ML-Based, Low-Latency Compression for the LHCb ECAL Upgrade . . . 5

End-to-End Neural Network Compression and Deployment for Hardware Acceleration Using PQuant and hls4ml . . . 6

Data-Driven Optimization of Chemical Vapor Deposition Using Machine Learning and Surrogate Modeling . . . 6

Tracking for the next ATLAS event filter with GNNs on GPUs . . . 7

Jet finding in real-time using an object detection CNN . . . 7

Graph Neural Networks for Online Track Reconstruction using FPGAs at the Event Filter for Phase-II Upgrades for the ATLAS Experiment . . . 8

High Throughput FPGA Deployment of Distilled Deep Sets Networks for Jet Preselection in the High-Level Trigger . . . 8

AMD AI-Engines in fixed latency environments . . . 9

Go small then go home - hyperparameter transfer for ML in HEP . . . 9

Planetary systems architecture based on a conditional generative model . . . 10

Advancing the CMS Level-1 Trigger: Jet Tagging with DeepSets at the HL-LHC . . . 10

Easing the path to deployment in ML4Sys through FPGAs . . . 11

Pushing Matrix-Vector Multiplication Performance on AMD AI Engines for Low-Latency Edge Inference . . . 11

When Less is More: Optimizing Graph Neural Networks and Knowledge Distillation for Efficient Particle Reconstruction and Identification in LHCb's Next-Generation Calorimeter . . . 12

Towards Fast and Interpretable Physics-Informed Learning: Second-Order Neurons and Mixed-Activation Networks . . . 12

Towards a Self-Driving Trigger: Adaptive Response in Real Time . . . 13

AI-Enhanced Dual-Coated QCM Sensor with FPGA-Based Impedance Analysis for Real-Time Multi-Parameter Environmental Monitoring . . . 14

Real-Time GPU Kalman-Filter Tracking via Kernel Refactoring and INT8 Surrogates for High-Luminosity Colliders . . . 15

Efficient data movement for Machine Learning inference in heterogeneous CMS software 16

Improving On-Chip Compression of High-Granularity Calorimeter Data with Conditional Autoencoders . . . 16

Efficient and Interpretable Transformers for Particle Physics . . . 17

Chisel4ml: Using Chisel For Direct Circuit Implementation of Deeply Quantized Neural Networks . . . 17

Chisel4ml: Direct Circuit Implementation of Deeply Quantized Neural Networks . . . 17

Designing and Deploying Low-Latency Neural Networks on FPGAs with HGQ and da4ml 18

A Real-Time GNN-based Clustering Algorithm for the Level 1 Calorimeter Trigger at Belle II . . . 19

FloatQuant: Arbitrary-Precision Minifloats in QONNX . . . 19

Evolution of the oneAPI backend for hls4ml 20

Experiences Deploying a Hybrid PVFinder Algorithm for Primary Vertex Reconstruction in LHCb's GPU-Resident HLT1 20

Electron/Positron - Proton Classification with AMS ECAL using Convolutional Vision Transformers 21

High-Throughput Ghost Track Rejection with Deep Learning at LHCb 21

Real-Time Anomaly Detection in the CMS Level-1 Trigger with AXOL1TL 22

FPGA-Optimized ML for Fast electron identification and pT regression with the CMS Phase-2 L1 trigger 22

Highly Granular Quantization for CICADA 23

EveNet: Towards a Generalist Event Transformer for Unified Understanding and Generation of Collider Data 23

Flexibly Equivariant Generative framework for Stochastic sub-grid Turbulence modeling 23

NeuraLUT-Assemble: Hardware-aware Assembling of Sub-Neural Networks for Efficient LUT Inference 25

FPGA-accelerated ML for real-time RHEED inference 25

Fast Synthetic X-Ray Generation for AI Diagnostics: DALL·E vs. Stable Diffusion 26

Enhancing Transit Detection to Uncover Long-Period Exoplanets 27

SparsePixels: Efficient Convolution for Sparse Data on FPGAs 27

Super Neural Architecture Codesign Package (SNAC-Pack) 28

Low-Latency On-Chip Tau Event Selection with Machine Learning for the Belle II Level-1 Trigger 28

GELATO: A Generic Event-Level Anomaly detection Trigger for ATLAS 29

Hierarchical Dataflow Accelerator of Interaction Networks for Large-Scale Particle Tracking on FPGA 29

Toward a Ultra-Fast, Energy-Efficient Readout of Calorimeters with Neuromorphic Processing 30

Scientific Machine Learning for Symbolic Recovery of Relativistic Effects in Black Hole Orbits 31

Embedding domain knowledge: Inductive biases for algorithmic alignment in Machine Learning 32

TinyML-Based Early Detection of Pak Choi Leaf Diseases Using FPGA in Vertical Farming Environments 33

FPGA-Based Quantized Inception-Style Neural Networks for Enhancing Spatial Resolution in Electron Microscopy 33

Accelerated Graph Neural Network Inference on FPGAs for Real-Time Muon Triggering at the HL-LHC . . . 34

Loss Landscape Analysis for Reliable Quantized ML Models for Scientific Sensing . . . 34

Porting MADGRAPH to FPGA Using High-Level Synthesis (HLS) . . . 35

Quantum-Inspired Tensor Network Models for Ultrafast Jet Tagging on FPGAs . . . 35

PrioriFI: Efficient Fault Injection for Edge Neural Networks . . . 36

QONNX Birds-of-a-Feather (BoF) Session . . . 36

Low-Latency Resource-Efficient GNNs for Jet Tagging on FPGAs . . . 37

FINN+: Towards Hassle-Free Co-Design of FPGA DNN Inference Accelerators . . . 37

Dynamic Scheduling Support for Faster ML Inference in hls4ml . . . 38

TDSCAN : Trigger Distributed Spatial Convolution Area Network . . . 39

ENABOL: Enabling Neural Backpropagation On-chip Learning for Edge AI Systems . . . 39

SuperSONIC: Cloud-Native Infrastructure for ML Inferencing . . . 39

Acceleration of a Quantized LeNet-based IDS on FPGA using FINN . . . 40

Locality-Sensitive Hashing-Based Efficient Point Transformer for Charged Particles Reconstruction . . . 41

Arbolta: A Fault Tolerance Study using Minimal Hardware Simulation . . . 42

Real-Time Andean Cryo-Hydrology Intelligence: A Heterogeneous Computing Framework for Glacial Hazard Early Warning . . . 42

MLOps Pipeline for Continuous Deployment of Machine Learning Algorithms for HEP . . . 43

Smartpixels: Intelligent pixel detectors: Towards a radiation hard ASIC with on-chip machine learning in 28nm CMOS . . . 43

Low-latency Jet Tagging for HL-LHC Using Transformer Architectures . . . 44

NomAD: Low-Latency Unsupervised Anomaly Detection for the ATLAS Trigger . . . 45

Part 1: Agile Hardware Design for AI: A Hands-On Tutorial with SODA and Bambu . . . 45

State Space Models for Scientific Time Series Applications . . . 46

Fast Adaptive Neural Control of Resonant Extraction at Fermilab . . . 47

Towards Online Machine Learning in DUNE Data Acquisition . . . 47

Accelerating Efficient Transformer Architectures for Point Cloud Data using hls4ml (REMOTE) . . . 48

da4ml: Distributed Arithmetic for Real-time Neural Networks on FPGAs . . . 49

wa-hls4ml: A Benchmark and Surrogate Models for hls4ml Resource and Latency Estimation . . . 49

FPGA Toolchain Integration for CI/CD workflows . . . 50

MLCommons Science Benchmarks . . . 50

COLLIDE-2V - 750 Million Dual-View LHC Event Dataset for Low-Latency ML . . . 51

KAN-LUT: Efficient LUT-Based Acceleration of Kolomogorov-Arnold Networks (KANs) on FPGAs . . . 52

Integrating Support for Google XLS in hls4ml . . . 53

AutoDeploy-HEP: An Intelligent Toolkit for ML Deployment on Heterogeneous Hardware in High-Energy Physics . . . 53

Inverse Design for Femtosecond-Laser Photonic Surfaces with Direct Gradient Optimization . . . 54

Balancing Prediction Performance, Transparency and Energy Consumption in Machine Learning Models for Data Streams . . . 55

A Hybrid Architecture for Real-Time Experiment Feedback in X-ray Microscopy Enabled by Edge and Cloud Computing . . . 55

Wearable TinyML Platforms for Biosignal Intelligence Across the Body . . . 56

Real-time ML and neuromorphic computing for smart robots . . . 56

The Instrumental Edge: Real-Time and AI-Ready Scientific Discovery . . . 56

AI for next-gen cellular networks (6G) . . . 56

Scaling up Advanced, Near-Sensor AI: An Open Platform Approach . . . 57

Co-Design for Efficient & Adaptive ML . . . 57

Efficient Graph neural networks at Google . . . 57

Fast inference with Decision Forests . . . 57

Next Generation GPU Signal Processing Pipeline for Radio Astronomy . . . 58

State space models for Project 8 event reconstruction . . . 58

Registration . . . 58

Commute to ETH Main Building (by ETH link 17:34 or 17:54). Meet Polyterrasse . . . 59

Part 2: Agile Hardware Design for AI: A Hands-On Tutorial with SODA and Bambu . . . 59

Conference summary . . . 60

Poster prizes and closing . . . 60

Poster & Coffee . . . 60

Satellite event: hls4ml dev meeting 60

An ML Pipeline for Real-time Gravitational Wave Alerts 60

Edge Deep Learning for Particle Physics (EPIGRAPHY) 61

Conference photo outside of auditorium! 61

Map to Tessin Grotto 61

Get your poster! 61

Tutorials / **1**

hls4ml

Authors: Benjamin Ramhorst[1]; Fast Machine Learning Collaboration[None]

[1] *ETH Zurich*

FPGAs provide unique advantages in the realm of machine learning acceleration. Unlike CPUs and GPUs, FPGAs allow for custom parallelism, data type precision and dataflow tailored specifically to the workload. Their reconfigurability enables the design of optimised hardware circuits that can reduce latency, power consumption, and improve throughput. Some common examples of FPGA-accelerated neural networks include particle classification, in-network traffic sniffing, and image segmentation for autonomous vehicles.

In this tutorial, we will introduce and hold a hands-on demo on hls4ml, an open-source library for real-time deployment of neural networks on FPGAs. hls4ml allows a seamless conversion from high-level models (e.g., from Keras or PyTorch) to low-latency, low-power FPGA designs. In this tutorial, we will cover the design choices behind hls4ml, from deeply pipelined dataflow architectures to model quantization and pruning. The hands-on demo will allow participants to experiment with hls4ml's Python API and try out the following concepts:

- Quantization-aware training with QKeras
- Model conversion with hls4ml
- Analysis of model latency and resource utilisation
- Tuning of model resources and latency

Finally, the tutorial will conclude with a live demo of the model inference on a real FPGA.

2

Registration

Tutorials / **3**

hls4ml tutorial

In this tutorial, you will get familiar with the hls4ml library. This library converts pre-trained Machine Learning models into FPGA firmware, targeting extreme low-latency inference. You will learn techniques for model compression, including how to reduce the footprint of your model using state-of-the-art techniques such as quantization. Finally, you will learn how to synthesize your model for implementation on the chip. Familiarity with Machine Learning using Python and Keras is beneficial for participating in this tutorial but not required. https://github.com/fastmachinelearning/hls4ml-tutorial

Tutorials / **4**

Tutorial 2

Tutorials / 5

Coyote v2: Open-source Abstractions and Infrastructure for FPGAs

Authors:
Gustavo Alonso, Maximilian Jakob Heer, Benjamin Ramhorst

As Moore's Law and Dennard Scaling reach their limits, computing is shifting toward heterogeneous hardware for large-scale data processing. Cloud vendors are deploying accelerators, like GPUs, DPUs, and FPGAs, to meet growing computational demands of ML and big data.

While FPGAs offer great flexibility and performance, practically integrating them in larger systems remains challenging due to the long development cycles and expertise required. To address this, we introduce Coyote v2, an open-source FPGA shell with high-level, OS-like abstractions. Broadly speaking, Coyote v2 strives to simplify the application deployment process and enable developers to solely focus on their application logic and its performance, rather than infrastructure development. By providing clear and simple-to-use interfaces in both hardware and software, Coyote v2 allows everyone to leverage the mentioned abstractions for customized acceleration offloads and build distributed and heterogeneous computer systems, consisting of many FPGAs, GPUs and CPUs. Coyote v2 has been re-engineered for flexibility of use as the basis platform for multi-tenant accelerators, SmartNICs, and near-memory accelerators.

This tutorial will cover Coyote v2's vFPGAs, which enable users to seamlessly deploy arbitrary applications on FPGAs, the built-in networking stacks for distributed applications and, finally, the shared virtual memory model, enabling FPGA interaction with other hardware (CPU, GPU, storage). Additionally, we will showcase Coyote's high-level software API, which enables easy, yet high performance, interaction from C++ with the FPGA. Finally, we will showcase Coyote's integration with hls4ml, performing inference on a PCIe-attached FPGA from a few lines of Python.

Tutorials / 6

Tutorial 4

Tutorials / 7

Tutorial 5

Tutorials / 8

Tutorial 6

Social / 9

Registration

Invited talks / **10**

Reasoning Language Models: Overview and Blueprint

Invited talks / **11**

Real-time mixed-signal electronic circuits for understanding and implementing neural computation

While machine learning has made tremendous progress in recent years, there is still a large gap between artificial and natural intelligence.
Closing this gap requires combining fundamental research in neuroscience with mathematics, physics, and engineering to understand the principles of neural computation and cognition.
Mixed-signal subthreshold analog and asynchronous digital electronic integrated circuits offer an additional means of exploring neural computation, by providing a computational substrate that shares many similarities with the one of biological brains.
In this subthreshold region of operation, transistor channels employ the same physics of carrier transport (diffusion) as the proteic channels of real neurons.
Thus, complex neuromorphic circuits and networks built following this approach share many similarities with real synapses, neurons, and cortical neural circuits.
In this presentation I will demonstrate how to build neuromorphic processors that use the physics of their computational substrate to directly emulate the physics of biological neural processes in real-time. I will demonstrate how to build complex recurrent electronic neural circuits with dynamics and response properties strikingly similar to those measured in real neural networks. I will argue that these systems can be used to complement numerical simulations in basic research and real-world applications.

Invited talks / **12**

The JAX scientific ecosystem

This talk provides an overview of several libraries in the open-source JAX ecosystem (such as Equinox, Diffrax, Optimistix, …) In short, we have been building an "autodifferentiable GPU-capable scipy". These libraries offer the foundational core of tools that have made it possible for us to train neural networks (e.g. score-based diffusions for image generation), solve PDEs, and smoothly handle hybridisations of the two (e.g. fit neural ODEs to scientific data). By the end of the talk, the goal is for you to be able to walk away with a slew of new modelling tools, suitable for tackling problems both in ML and in science.

Invited talks / **13**

Real-time inference at the LHC

The real-time processing of data created by the Large Hadron Collider's (LHC) experiments, amounting to over 10% of worldwide internet traffic, is one of the greatest computing challenges ever attempted. I will discuss the concrete applications of real-time processing in the LHC's main experiments, and the technological innovations in this area over the past decades. I will also reflect on the development of communities focused on real-time data processing in experimental HEP, and

describe the ways in which these communities have expanded the physics reach of their experiments far beyond what had been originally imagined. Finally I will look ahead to the challenges facing the LHC experiments in the high-luminosity era of the LHC, and touch on ways in which the widespread availability of precision timing in our detectors will transform real-time processing in the next decades.

Invited talks / **14**

Real-time inference in gravitational wave astronomy (REMOTE)

Invited talks / **15**

Welcome

16

Poster Session and Coffee

17

Fast ML foundation discussion

Tutorials / **18**

Coyote v2: Open-source Abstractions and Infrastructure for FPGAs

Authors: Benjamin Ramhorst[1]; Gustavo Alonso[1]; Maximilian Jakob Heer[1]

[1] *ETH Zurich*

As Moore's Law and Dennard Scaling reach their limits, computing is shifting toward heterogeneous hardware for large-scale data processing. Cloud vendors are deploying accelerators, like GPUs, DPUs, and FPGAs, to meet growing computational demands of ML and big data.

While FPGAs offer great flexibility and performance, practically integrating them in larger systems remains challenging due to the long development cycles and expertise required. To address this, we introduce Coyote v2, an open-source FPGA shell with high-level, OS-like abstractions. Broadly speaking, Coyote v2 strives to simplify the application deployment process and enable developers to solely focus on their application logic and its performance, rather than infrastructure development. By providing clear and simple-to-use interfaces in both hardware and software, Coyote v2 allows everyone to leverage the mentioned abstractions for customized acceleration offloads and build distributed and heterogeneous computer systems, consisting of many FPGAs, GPUs and CPUs. Coyote

v2 has been re-engineered for flexibility of use as the basis platform for multi-tenant accelerators, SmartNICs, and near-memory accelerators.

This tutorial will cover Coyote v2's vFPGAs, which enable users to seamlessly deploy arbitrary applications on FPGAs, the built-in networking stacks for distributed applications and, finally, the shared virtual memory model, enabling FPGA interaction with other hardware (CPU, GPU, storage). Additionally, we will showcase Coyote's high-level software API, which enables easy, yet high performance, interaction from C++ with the FPGA. Finally, we will showcase Coyote's integration with hls4ml, performing inference on a PCIe-attached FPGA from a few lines of Python.

Posters and coffee / **20**

Optimizing Reciprocal Human-Machine Learning Decisions

Author: Gila Fruchter[None]

Co-author: Dov Te'eni

Recent advances in machine learning have raised ethical concerns in both industry and academia regarding the uncontrollable diffusion of AI and the diminishing human capacity to oversee its impacts. These concerns underscore the need for regulatory and design approaches that maintain human oversight in AI-driven decision-making. Keeping humans in the loop is essential for auditing, fairness, trust, and—critically—ongoing human learning. Without continuous human learning, the knowledge gap between ever-learning machines and static human experts widens, risking the exclusion of humans from meaningful oversight. We focus on future human-AI configurations for high-stakes environments, particularly those that support mutual learning. One such configuration is Reciprocal Human-Machine Learning (RHML), in which humans and machines learn both independently and from each other to reduce their knowledge gap. While RHML holds promise for improving decision quality, it also entails significant operational costs. Its effectiveness depends on the learning capacities of both agents and the allocation of effort across different learning modes.

To formalize RHML, we develop an optimal control model that maximizes a performance objective —balancing decision accuracy against the cost of learning efforts. In this model, decision accuracy evolves as a function of effort allocated to independent and reciprocal learning. We assume that learning contributes to knowledge, which in turn drives accuracy—modulated by knowledge complexity, explainability, and decay. We derive both an open-loop and a feedback solution, where the latter is characterized by an optimal allocation of learning efforts that directly depends on the system's current level of accuracy.

Our analysis explores how system parameters influence learning effort, decision accuracy, model complexity, and maximized performance. The model also identifies optimal stopping times, where continued learning is no longer cost-effective. We provide insights on inter-channel allocation and inter-agent allocation. We conclude with implications for AI design, policy, and management in decision-critical settings.

Contributed talks / **21**

Radiation-Hard, ML-Based, Low-Latency Compression for the LHCb ECAL Upgrade

Authors: Eluned Anne Smith[1]; Julian Garcia Pardinas[1]; Katya Govorkova[1]; Vladimir Loncar[2]

[1] *Massachusetts Inst. of Technology (US)*

[2] *CERN*

The LHCb Upgrade II will operate at a data rate of 200 Tb/s, requiring efficient real-time data reduction. A major challenge of this pipeline is the transfer of full timing information from the

frontend Electromagnetic Calorimeter (ECAL) to the backend for processing, which is critical for resolving pile-up, background suppression, and enhancing energy resolution. Due to the data rate, full timing information cannot be transmitted, requiring compression of data to reduce bandwidth. To address this, we develop a machine-learning-based compression algorithm, capable of learning detector-specific correlations in the data, outperforming generic compression schemes. Central to this effort is the extension of the hls4ml framework to fully support Microchip architectures, enabling the deployment of optimised autoencoder networks on the PolarFire FPGAs. These networks compress high-granularity timing data with minimal latency, achieving O(25 ns) inference times within stringent resource constraints. This development is key to reducing bandwidth while preserving physics performance and represents an essential step toward maintaining the physics reach of LHCb Upgrade II in the high-luminosity era.

Contributed talks / **22**

End-to-End Neural Network Compression and Deployment for Hardware Acceleration Using PQuant and hls4ml

Author: Roope Oskari Niemi[None]

Co-authors: Chang Sun [1]; Anastasiia Petrovych [2]; Enrico Lupi [3]; Dimitrios Danopoulos [2]; Arghya Ranjan Das [4]; Sebastian Dittmeier [5]; Michael Kagan [6]; Miaoyuan Liu [4]; Vladimir Loncar [2]

[1] *California Institute of Technology (US)*

[2] *CERN*

[3] *CERN, INFN Padova (IT)*

[4] *Purdue University (US)*

[5] *Ruprecht-Karls-Universitaet Heidelberg (DE)*

[6] *SLAC National Accelerator Laboratory (US)*

As the demand for efficient machine learning on resource-limited devices grows, model compression techniques like pruning and quantization have become increasingly vital. Despite their importance, these methods are typically developed in isolation, and while some libraries attempt to offer unified interfaces for compression, they often lack support for deployment tools such as hls4ml. To bridge this gap, we developed PQuant, a Python library designed to streamline the process of training and compressing machine learning models. PQuant offers a unified interface for applying a range of pruning and quantization techniques, catering to users with minimal background in compression while still providing detailed configuration options for advanced use. Notably, it features built-in compatibility with hls4ml, enabling seamless deployment of compressed models on FPGA-based accelerators. This makes PQuant a versatile resource for both researchers exploring compression strategies and developers targeting efficient implementation on edge devices or custom hardware platforms. We will present the PQuant library, the performance of several compression algorithms implemented with it, and demonstrate the conversion flow of a neural network model from an uncompressed state to optimized firmware for FPGAs.

Posters and coffee / **23**

Data-Driven Optimization of Chemical Vapor Deposition Using Machine Learning and Surrogate Modeling

Author: Raja Selvam[None]

Co-author: Pradeep George

Chemical Vapor Deposition (CVD) optimization is critical for advancing thin-film quality and process efficiency in semiconductor and optoelectronic applications, yet traditional methods like Computational Fluid Dynamics (CFD) simulations and empirical tuning are often computationally intensive or lack adaptability. To address this challenge, this study presents a data-driven machine

learning (ML) framework that optimizes two key CVD performance metrics—deposition rate and film uniformity—by integrating XGBoost-based predictions with polynomial surrogate modeling for computationally efficient, constraint-aware process improvement. Building on prior work where XGBoost was identified as the most accurate predictor of CVD outcomes compared to other ML models and CFD benchmarks, this research advances the field by constructing surrogate equations from XGBoost-predicted outputs over a test dataset, enabling rapid optimization via the Sequential Least Squares Programming (SLSQP) algorithm. Three optimization scenarios were systematically evaluated: (i) maximizing deposition rate under a uniformity constraint, ensuring high throughput without compromising film quality; (ii) minimizing non-uniformity while meeting a minimum deposition threshold, critical for precision applications; and (iii) maximizing the deposition-to-uniformity ratio, a balanced metric for industrial scalability. The optimized process parameters—susceptor temperature and inlet gas velocity—were derived for each case, revealing distinct operational regimes: high-temperature, low-velocity conditions maximized deposition rate (achieving a 12% improvement over baseline), while moderate-temperature, high-velocity setups minimized non-uniformity (to <5% variation), and an intermediate trade-off regime optimized the deposition-to-uniformity ratio (yielding a 15% gain). Comparative analysis with traditional gradient-based optimization techniques demonstrated that the proposed ML-driven approach not only matched accuracy but also reduced computational time by 40 times, highlighting its suitability for real-time adaptive control in industrial settings. The success of this framework lies in its ability to replace costly CFD iterations with lightweight surrogate models while maintaining fidelity, a breakthrough for intelligent manufacturing and digital twin implementations. Beyond CVD, the methodology's flexibility combining ML with constrained optimization offers broad applicability in other thin-film deposition techniques and complex material synthesis processes. Future directions include embedding this framework with in-situ sensor feedback for closed-loop control and extending it to multi-objective scenarios incorporating additional film properties (e.g., stress, stoichiometry). By bridging the gap between data-driven modeling and industrial process optimization, this work contributes a scalable, efficient paradigm for next-generation manufacturing, aligning with Industry 4.0 advancements in automation and predictive analytics.

Posters and coffee / **26**

Tracking for the next ATLAS event filter with GNNs on GPUs

Authors: ATLAS Collaboration[None]; Benjamin Huth[1]

[1] *CERN*

Graph Neural Networks (GNNs) have been in the focus of machine-learning-based track reconstruction for high-energy physics experiments during the last years. Within ATLAS, the GNN4ITk group has investigated this type of algorithm for track reconstruction at the High-Luminosity LHC (HL-LHC) using the future full-silicon Inner Tracker (ITk).

The Event Filter (EF) is part of the ATLAS Trigger and Data Acquisition (TDAQ) system and will consist of a computing farm that runs a limited set of event reconstruction algorithms to provide the accept/reject decision for offline storage. The decision on the system design choice for the EF farm is scheduled for late 2025.

We are exploring the use of the GNN4ITk track-finding approach alongside other tracking tools from the open-source ACTS (A Common Tracking Software) toolkit to develop a candidate pipeline targeting GPU hardware aimed at meeting the ATLAS Event Filter throughput and physics performance goals.

We will present the implementation strategy, optimizations and computing performance results, as well as the track reconstruction performance for the proposed candidate pipeline.

Contributed talks / **27**

Jet finding in real-time using an object detection CNN

Author: Leon Bozianu[1]

[1] *Universite de Geneve (CH)*

The ATLAS trigger system will undergo a comprehensive upgrade in advance of the HL-LHC programme. In order to deal with the increased data bandwidth trigger algorithms will be required to satisfy stricter latency requirements. We propose a method to speed up the current calorimeter-only preselection step and to aid trigger decisions for hadronic signals containing jets.
We demonstrate the use of a dedicated object-detection Convolutional Neural Network (CNN) for jet finding in the ATLAS calorimeter. The modified computer vision model is employed in the task of jet detection to identify and localise jets within the central calorimeter acceptance and to subsequently estimate their transverse momenta. A custom architecture is introduced to reduce the number of learnable parameters required for improved inference speed. The model performance is evaluated on a set of simulated particle interactions in the ATLAS detector with up to 200 concurrent pile-up interactions.

Posters and coffee / 28

Graph Neural Networks for Online Track Reconstruction using FPGAs at the Event Filter for Phase-II Upgrades for the ATLAS Experiment

Authors: ATLAS Collaboration[None]; Jared Burleson[1]

[1] *University of Illinois at Urbana-Champaign*

The High-Luminosity LHC (HL-LHC) will provide an order of magnitude increase in integrated luminosity and enhance the discovery reach for new phenomena. The increased pile-up foreseen during the HL-LHC necessitates major upgrades to the ATLAS detector and trigger. The Phase-II trigger will consist of two levels, a hardware-based Level-0 trigger and an Event Filter (EF) with tracking capabilities. Within the Trigger and Data Acquisition group, a heterogeneous computing farm consisting of CPUs and potentially GPUs and/or FPGAs is under study, together with the use of modern machine learning algorithms such as Graph Neural Networks (GNNs). GNNs are a powerful class of geometric deep learning methods for modeling spatial dependencies via message passing over graphs. They are well-suited for track reconstruction tasks by learning on an expressive structured graph representation of hit data and considerable speedup over CPU-based execution is possible on FPGAs. In this talk, we will provide novel results of fast track finding within a single FPGA using GNN for pattern recognition as part of the larger project for Phase-II EF system tracking on a FPGA. We present physics performance results and technical measurements related to power, throughput and latency aimed at motivating the viability of GNNs for pattern recognition as part of a fast tracking finding pipeline on an FPGA for the ATLAS Phase-II EF tracking system. We will highlight specific techniques employed to achieve our results, such as sequential processing of the detector, interaction network model hyperparameter tuning, model pruning and quantization-aware training.

Posters and coffee / 30

High Throughput FPGA Deployment of Distilled Deep Sets Networks for Jet Preselection in the High-Level Trigger

Authors: ATLAS TDAQ collaboration[None]; Lucas Bezio[1]

[1] *Universite de Geneve (CH)*

Deep Sets-based neural networks are well-suited to learning from unordered, variable-length inputs such as particle tracks associated with jets. Their permutation-invariant structure makes them attractive for high-energy physics (HEP) applications where input ordering is ambiguous and throughput is a critical constraint. In this work, we explore the use of such architectures on Field-Programmable Gate Arrays (FPGAs) to enable fast and resource-efficient inference within the High-Level Trigger (HLT), where track-level features become available and event processing must be optimized for rate rather than latency.

We build on the DIPS (Deep Impact Parameter Sets) algorithm, a Deep Sets-based flavour-tagging model previously used in ATLAS, and adapt it for quantization-aware training and efficient FPGA implementation using QKeras and hls4ml. To maximize throughput without sacrificing classification performance, we introduce knowledge distillation: a high-capacity teacher network guides the training of a compact student model optimized for FPGA deployment. This approach allows us to retain strong discriminative power while drastically reducing inference cost.

Our implementation demonstrates that Deep Sets networks can be effectively mapped to FPGA hardware, including mechanisms for handling variable-length, permutation-invariant inputs. We report detailed results from real FPGA deployments, including resource usage, measured latency, and especially sustained throughput under different configurations.

Rather than competing directly with more sophisticated architectures such as graph neural networks (GNNs), our goal is to provide a high-throughput preselection stage that can reduce the number of events requiring costly downstream inference. In doing so, we enable a more efficient use of compute resources in the HLT pipeline, providing early rejection capabilities with minimal impact on tagging performance.

This work highlights a complementary strategy for deploying ML on FPGAs in HEP: prioritizing throughput to enable scalable inference pipelines and alleviate computational bottlenecks, especially in data-rich environments such as the upgraded ATLAS trigger.

Contributed talks / 31

AMD AI-Engines in fixed latency environments

Author: Ioannis Xiotidis[1]

[1] *CERN*

The ATLAS Level-0 Global Trigger is a mission critical system opting to take advantage of the full calorimeter granularity during Run-4 and beyond. Level-0 Global will be executing a cascade of trigger algorithms combined both the calorimeter information and the muons. Within the Next Generation Trigger project at CERN there is a dedicated work package (WP2.1) exploring large deployment of Machine Learning based algorithms to further enhance selections within the Global system. Given the tight latency and throughput conditions that Global is operating at, any solution developed at WP2.1 has to be deployed in custom hardware solution using FPGAs. The cutting edge FPGA technologies include within the same package dedicated co-processing chiplets optimised for Machine Learning applications. Such a device is the Adaptive Interface Engines provided in the Versal Premium packages. In this talk we will present the work performed within the scope of the NGT WP2.1 aiming to characterise the performance of those devices, provide some generic implementations for specific ML models and explore the feasibility of deploying them in the harsh environment of a mission critical system (L0-Global).

Contributed talks / 32

Go small then go home - hyperparameter transfer for ML in HEP

Authors: Gage DeZoort[1]; Liv Helen Vage[1]

[1] *Princeton University (US)*

Tuning hyperparameters of ML models, especially large ML models, can be time consuming and computationally expensive. As a potential solution, several recent papers have explored hyperparameter transfer. Under certain conditions, the optimal hyperparameters of a small model are also optimal for larger models. One can therefore tune only the small model and transfer the hyperparameters to the larger model, saving a large amount of time and effort. This work explores how well the idea holds up in high-energy physics by applying it to three existing ML pipelines: metric learning for particle tracking, autoencoders for anomaly detection, and particle transformers for jet tagging. These cover several common ML architectures and reflect models currently used or in development at CMS and other experiments. We show that with a few changes to the models, hyperparameters can often be transferred across both neural net depth and width. We focus on learning rate transfer, but also show results on a few other hyperparameters. A few guidelines are introduced, encouraging the use of hyperparameter transfer in future HEP ML models.

Posters and coffee / **34**

Planetary systems architecture based on a conditional generative model

Author: Sara Marques[1]

Co-author: Yann Daniel Pierre Alibert [2]

[1] *UniBe*

[2] *University of Bern*

Understanding the diversity and structure of planetary systems requires capturing not only the properties of individual planets but also the statistical relationships between planets within the same system and their interaction with the host star.

Traditional population synthesis models, such as the Bern model, provide physically motivated insights into these correlations, but their computational cost limits the applications. To address this, we introduce a conditional generative model that produces synthetic planetary systems with high fidelity and efficiency, while explicitly incorporating host star and protoplanetary disk properties such
as the stellar metallicity, disk lifetime, mass and so on.

The model architecture builds on a previous transformer framework (Alibert, Davoult and Marques in press, see http://ai4exoplanets.com). Through conditioning, the new model is expected to capture system-level features such as planet multiplicity, orbital distribution, but also to link such properties to the host Star and disk. The training is performed using Bern model synthetic simulations to avoid observational bias and the model is capable of producing rapidly new planetary systems that remain statistically consistent with those from the simulations.

Contributed talks / **35**

Advancing the CMS Level-1 Trigger: Jet Tagging with DeepSets at the HL-LHC

Authors: Christopher Edward Brown[1]; Duc Hoang[2]; Sebastian Wuchterl[1]; Sioni Paris Summers[1]; Stella Felice Schaefer[3]

[1] *CERN*

[2] *Massachusetts Inst. of Technology (US)*

[3] *Hamburg University (DE)*

At the Phase-2 Upgrade of the CMS Level-1 Trigger (L1T), particles will be reconstructed by linking charged particle tracks with clusters in the calorimeters and muon tracks from the muon station. The 200 pileup interactions will be mitigated using primary vertex reconstruction for charged particles and a weighting for neutral particles based on the distribution of energy in a small area. Jets will be reconstructed from these pileup-subtracted particles using a fast cone algorithm. For the first time at the CMS L1T, the particle constituents of jets will be available for jet tagging. In this work we present a new multi-class jet tagging neural network (NN). Targeting the L1T, the NN is a small DeepSets architecture, and trained with Quantization Aware Training. The model predicts the classes: light jet (uds), gluon, b, c, τ_h^+, τ_h^-, electron, muon. The model additionally predicts the p_T of the object. The new model enhances the selection power of the L1T for important processes for CMS at the High Luminosity LHC such as di-Higgs and Higgs production via Vector Boson Fusion. We present the model including its performance at object tagging and deployment into the L1T FPGA processors, and showcase the improved trigger capabilities enabled by the new tagger.

Contributed talks / 37

Easing the path to deployment in ML4Sys through FPGAs

Author: Maximilian Heer[1]

Co-authors: Benjamin Ramhorst [1]; Gustavo Alonso [1]

[1] *ETH Zurich*

Machine Learning (ML) techniques are increasingly applied for the optimization of complex computing systems, but their integration into core low-level system mechanisms remains limited. A key barrier is the lack of accessible, high- performance interfaces at the boundary between software and hardware as well as hardware-offloaded ML-inference at full systems speed. In this presentation, we show how Field Programmable Gate Arrays (FPGAs) can be a key enabler for closing this very gap: The combination of mature FPGA shells, thus full-fledged computer systems on reconfigurable fabric, and modern ML compilers for hardware-accelerated inference enables rapid prototyping and deployment of be-spoke ML models directly at the interfaces of key system mechanisms. This approach allows for ultra-low latency, real-time decision making in system components such as memory management, scheduling logic and network control. We outline a vision of the fully ML-optimized FPGA-SmartHub, serving as a research platform for system optimization in both classic computer systems and next-generation accelerators.

Contributed talks / 38

Pushing Matrix-Vector Multiplication Performance on AMD AI Engines for Low-Latency Edge Inference

Author: Dimitrios Danopoulos[1]

Co-authors: Roope Oskari Niemi ; Enrico Lupi [2]; Anastasiia Petrovych [1]; Sebastian Dittmeier [3]; Michael Kagan [4]; Vladimir Loncar [1]

[1] *CERN*

[2] *CERN, INFN Padova (IT)*

[3] *Ruprecht-Karls-Universitaet Heidelberg (DE)*

[4] *SLAC National Accelerator Laboratory (US)*

Matrix-vector (GEMV) operations are a common building block in many deep learning models, particularly for large dense layers found in convolutional neural networks (CNNs) and multi-layer perceptrons (MLPs). Despite their importance, GEMV kernels have historically underperformed compared to matrix-matrix (GEMM) operations due to their lower arithmetic intensity and limited data reuse, making them harder to scale efficiently. This work presents the first comprehensive analysis and optimization of matrix-vector operations using AMD's AI Engines on the latest AIE-ML architecture. It addresses key bottlenecks in deploying AI models that rely on such operations targetting low-latency edge inference, such as meeting the tight real-time requirements of the CERN trigger system. Our proposed GEMV kernel achieves high throughput and low latency through exploitation of the AI Engine array, scaling efficiently across tiles both horizontally and vertically through a custom placement strategy. Furthermore, we introduce a novel graph connection mechanism that enables efficient pipelining across multiple layers. The design is modular and can be easily integrated with other frameworks such as hls4ml in a straightforward manner. Our multi-layer implementation achieves close to microsecond-level latency, demonstrating its suitability for ultra-low-latency applications. These results make AMD's AI engines a realistic middle ground solution that can offer the scalability that FPGAs struggle to reach for large models, while maintaining the ultra-low latency that GPUs typically cannot provide.

Posters and coffee / **39**

When Less is More: Optimizing Graph Neural Networks and Knowledge Distillation for Efficient Particle Reconstruction and Identification in LHCb's Next-Generation Calorimeter

Authors: Andrey Ustyuzhanin[1]; Cilicia Uzziel Perez[2]; Elisabet Golobardes Ribe[2]; Felipe Luan Souza De Almeida[3]; Jonas Eschle[None]; Julian Garcia Pardinas[4]; Justin Bartz[3]; Katya Govorkova[4]; Ke Wei[5]; Matthew Scott Rudolph[3]; Miriam Calvo Gomez[2]; Rafael Silva Coutinho[6]; Xavier Vilasis Cardona[2]

[1] *Constructor University (DE)*

[2] *La Salle, Ramon Llull University (ES)*

[3] *Syracuse University (US)*

[4] *Massachusetts Inst. of Technology (US)*

[5] *Wuhan University (CN)*

[6] *CBPF - Brazilian Center for Physics Research (BR)*

Graph Neural Networks (GNNs) have become promising candidates for particle reconstruction and identification in high-energy physics, but their computational complexity makes them challenging to deploy in real-time data processing pipelines. In the next-generation LHCb calorimeter, detector hits—characterized by energy, position, and timing—can be naturally encoded as node features, with spatial and energy-based relationships captured through edge features. This study investigates strategies to reduce both the structural complexity and numerical precision of GNNs to meet stringent real-time processing and resource constraints. We demonstrate that omitting explicit edge features and replacing conventional full message passing with learnable, permutation-invariant aggregation functions results in up to an 8× reduction in CPU inference time, while maintaining or even surpassing the energy resolution and classification performance of baseline methods. Furthermore, we explore post-training quantization, reducing model weights from 32-bit floating point (FP32) to 16-bit or 8-bit integers. While quantization could potentially offer additional efficiency gains, lightweight GNNs with approximately ˜100k parameters exhibit minor inference time performance degradation under aggressive precision reduction. We also present our knowledge distillation experiment, where we train a compact student model to mimic the performance of a larger, more complex teacher network. Our findings provide practical design guidelines for developing fast, efficient, and high-performing GNNs for real-time particle reconstruction in LHCb's upgraded calorimeter, while also highlighting the limitations of quantization in small neural network architectures.

Posters and coffee / **40**

Towards Fast and Interpretable Physics-Informed Learning: Second-Order Neurons and Mixed-Activation Networks

Author: João Paulo De Souza Böger[None]

Co-authors: Francisco Camara Pereira [1]; Guido Cantelmo [1]; Serio Agriesti [1]

[1] *DTU*

Complex simulators are central to scientific research, forecasting, and real-world applications. However, they often require intensive computational resources and suffer from scalability issues —challenges amplified in the big data era. The APEX project addresses these limitations by designing novel efficient architectures for scientific simulators, exploring inductive biases, causal relations, and architectural innovations that learn and generalize simulator dynamics in a fast, scalable, uncertainty-aware, and interpretable way.

Our goal is to build surrogate models (or metamodels) that generalize beyond their training data, even when conditions change —a common challenge where standard machine learning (ML) models often fail. Many simulators, like those used in transport, climate modeling, epidemiology, and hydrodynamics, are governed by ordinary and partial differential equations (ODEs/PDEs). Efficiently learning these dynamics is crucial for effective surrogate modeling. To address the limitations of traditional ML approaches, we introduce inductive biases that guide learning based on known constraints of theses systems. While traditional PDE solvers and vanilla ML models struggle with scalability, determining which biases best help models generalize ODE/PDE dynamics remains an open challenge. This calls for model compression and reduced-parameter architectures that effectively blend expert knowledge with ML to create fast ML solutions for dynamical system simulators.

Quadratic neurons and domain-specific activation functions (MixFunn) \[1\], motivated by the solutions'analytic forms to differential equations, provide a method for integrating domain insights into a neural network architecture. Coupled with soft constraints incorporated in the loss function as suggested by Physics-Informed Neural Networks (PINNs) \[2\], this innovative design matches or surpasses the performance of vanilla PINNs, yet utilizes considerably fewer parameters —facilitating closed-form solutions and enhancing model efficiency, which in turn can accelerate scientific discovery.

Building on this idea, we hypothesize that polynomial neurons and tailored non-linearities are key to creating scalable, generalizable architectures for scientific simulators. Our contributions include an extensive evaluation of MixFunn, along with PINNs, neuralODEs \[3\], and Equation Learners \[4\], in four benchmark ODE-based simulators. We use two recent proposed metrics \[5\] better suited for assessing learning quality OOD in surrogate models and test their robustness to noise in data. Furthermore, we introduce a test-time adaptation module \[6\], which allows models to refine parameters when limited trajectory data is available, improving accelerated inference for scientific applications.

We evaluate all approaches on widely used benchmarks: the SIR epidemic model, the Lotka–Volterra predator-prey system, Duffing oscillator, and Van der Pol oscillator. These systems span diverse tasks and dynamics, providing a robust testbed for comparing inductive biases, OOD generalization, and computational efficiency.

This work highlights the importance of architectural choices and domain-informed priors to closing the gap between robustness, speed, and generalization in scientific surrogates. It paves the way for compact, interpretable, and fast metamodels that encode scientific structure for real-world simulation tasks.

Contributed talks / 43

Towards a Self-Driving Trigger: Adaptive Response in Real Time

Authors: Abhijith Gandrakota[1]; Cecilia Tosciri[2]; Christian Herwig[3]; David Miller[2]; Giovanna Salvi[3]; Jennifer Ngadiuba[4]; Nhan Tran[1]; Shaghayegh Emami[3]; Yuxin Chen[None]; Zixin Ding[None]

[1] *Fermi National Accelerator Lab. (US)*

[2] *University of Chicago (US)*

[3] *University of Michigan (US)*

[4] *FNAL*

The trigger systems of ATLAS and CMS currently reject vast numbers of potentially valuable collision events due to their conservative, static designs, a limitation that directly hampers discovery potential. We propose an alternative to these rigid, hand-tuned menus with an autonomous controller capable of dynamically optimizing trigger performance in real time.
In this work, we demonstrate that, by continuously adapting trigger thresholds and resource allocations in response to evolving experimental conditions (such as pileup, beam-induced backgrounds, or detector drifts), our self-driving trigger system maintains peak performance across multiple axes: signal efficiency, rate of unusual events, and computational cost.
Crucially, we validate our approach through playback of zero-bias L1 trigger data from real LHC collision events. Anomaly detection triggers serve as a natural testbed, where the gains from dynamic prioritization are especially clear. More broadly, this framework is particularly powerful for triggers that rely on subtle correlations and are inherently more sensitive to changes in detector conditions. Our architecture supports both lightweight feedback controllers and more powerful reinforcement learning approaches, laying the foundation for truly adaptive, intelligent triggering at the LHC and beyond.

Posters and coffee / 44

AI-Enhanced Dual-Coated QCM Sensor with FPGA-Based Impedance Analysis for Real-Time Multi-Parameter Environmental Monitoring

Author: N Ramakrishnan[1]

Co-authors: Chun Leon Chung [2]; Wei Yin Lim [2]; You Jin Lam [2]

[1] *Associate Professor, Monash University Malaysia*

[2] *Monash University Malaysia*

Quartz Crystal Microbalance (QCM) sensors are renowned for their high sensitivity to mass changes, making them ideal for detecting environmental parameters such as relative humidity (RH) and ultraviolet (UV) radiation. In this work, we present an AI-driven, dual-sided coated QCM sensor integrated with advanced machine learning (ML) and implemented on a real-time hardware platform. This sensor employs a dual-coating strategy with zinc oxide (ZnO) for UV detection and graphene oxide (GO) for RH sensitivity. These materials were selectively coated on opposite sides of the QCM via RF sputtering and electrophoretic deposition, respectively, enabling simultaneous multi-parameter sensing. Experimental data was collected under a controlled chamber setup with environmental conditions modulated and recorded using standard instrumentation.
To overcome the limitations of traditional analytical techniques in handling multi-signal responses, various deep learning models—Convolutional Neural Networks (CNN), Gated Recurrent Units (GRU), and Long Short-Term Memory (LSTM)—were employed. The GRU model outperformed others for RH prediction with a mean squared error (MSE) of 27.433 and R^2 of 0.798, while the LSTM model excelled for UV prediction with an MSE of 2.176 and R^2 of 0.843. These results indicate strong correlations between QCM response parameters (Fs, Fp, Zs, Zp) and environmental measurands, allowing robust and accurate modeling. In contrast, a regression-based AutoGluon ensemble ML model, although useful as a baseline, demonstrated limited performance in capturing non-linear dependencies in the sensor data.
To transition from laboratory proof-of-concept to real-time application, the developed model is currently being implemented on an Altera Cyclone III FPGA platform. A custom impedance analyzer daughter board was developed to interface with the QCM sensor, enabling precise measurement of resonance and impedance characteristics. This miniaturized hardware implementation addresses

key limitations of conventional impedance analysis methods—such as bulkiness, slow data acquisition, and high cost—by offering a compact, faster alternative without significant loss in performance. In the initial hardware prototype evaluation, the impedance analyzer demonstrated functionality across the 6 MHz range with a dynamic impedance detection range from 10 Ω to 10^6 Ω, a frequency resolution of 0.55 Hz, and a data acquisition speed of 1.5 ms per point. Experimental comparison with commercial analyzers showed a ±7.5% deviation in impedance magnitude, which was significantly improved with calibration techniques. However, noise in calibration parameters suggests further optimization is necessary.
This integrated QCM sensing platform, enhanced with AI, presents a novel approach to multi-parameter environmental monitoring. It holds significant promise for deployment in real-time applications such as air quality surveillance, industrial safety systems, and smart environmental sensing. Future work will focus on refining model robustness, improving calibration consistency, and expanding sensing capabilities to additional environmental parameters. The results and hardware will be demonstrated in the poster session, providing a valuable platform for discussion on applying real-time ML in acoustic wave sensor systems.

Posters and coffee / **45**

Real-Time GPU Kalman-Filter Tracking via Kernel Refactoring and INT8 Surrogates for High-Luminosity Colliders

Author: Hao-Chun Liang[1]

Co-authors: Bo-Cheng Lai [2]; Yuan-Tang Chou [3]

[1] *Institute of Pioneer Semiconductor Innovation, National Yang Ming Chiao Tung University*

[2] *Institute of Electronics, National Yang Ming Chiao Tung University*

[3] *University of Washington (US)*

As the era of the High-Luminosity Large Hadron Collider (HL-LHC) approaches, the GPU-accelerated High-Level Trigger (HLT) of the CMS experiment faces a stringent requirement to reduce the Level-1 readout stream from 100 kHz to 5 kHz, a twenty-fold decrease essential to adhere to archival bandwidth constraints [1], [2]. Meeting this demand necessitates highly efficient real-time charged-particle tracking.

In the recent release of the ACTS Traccc pipeline, the Kalman-filter Fit kernel has emerged as the predominant latency bottleneck, primarily due to excessive register pressure and serialization of matrix inversion operations within GPU warp execution [3]. To address these performance limitations, we propose two synergistic GPU optimizations to increase the kernel throughput.

First, we introduce a kernel refactoring method to restructure the Fit kernel into three distinct computational phases—Predict, Update, and Finalize. Each of the phases is clearly delineated by a single __syncthreads() synchronization barrier. This refactoring method facilitates more efficient compiler-driven register allocation, significantly shortens register lifetimes, and effectively reduces register spill traffic, thereby enhancing kernel throughput.

Second, we substitute the computationally intensive analytic matrix inversion operations with a quantization-aware INT8 multilayer perceptron (MLP). This surrogate model, using a network architecture with three hidden layers, is explicitly trained to approximate the 6×2 Kalman-gain matrix. By leveraging NVIDIA's __dp4a integer dot-product instruction together with extensive compile-time optimizations using C++ constexpr, we can perform the majority of INT8 operations directly in registers, dramatically reducing shared-memory traffic and further cutting overall latency.

Evaluations conducted on Geant4-simulated Open-Data-Detector events demonstrate substantial performance gains [4]. Kernel refactoring alone results in a throughput enhancement of approximately 15%. Furthermore, the MLP surrogate achieves high precision, replicating the analytic Kalman-gain matrix with a mean-squared error less than 8×10^{-5}. Comprehensive testing on an NVIDIA RTX 2080 Ti GPU illustrates the combined efficacy of these optimizations, improving the Fit-kernel reciprocal throughput by 5.22x (from an initial 11.5 ms to 2.2 ms). Concurrently, the overall end-to-end pipeline reciprocal throughput decreases from 23.5 ms to 8.23 ms. This yields an

overall speed-up factor of 2.86×, effectively transitioning the kernel execution from being memory- and special function unit (SFU)-bound to predominantly compute-bound.

In conclusion, we propose two methods: lightweight, quantization-aware MLP surrogates and meticulous kernel refactoring, to significantly optimize the GPU-based track fitting. This optimization successfully approaches the stringent budgetary constraints in terms of computational latency imposed by the HL-LHC HLT, marking a critical advancement toward efficient and effective real-time particle tracking at future high-luminosity collider experiments.

Posters and coffee / **46**

Efficient data movement for Machine Learning inference in heterogeneous CMS software

Author: Davide Valsecchi[1]

[1] *ETH Zurich (CH)*

Efficient data processing using machine learning relies on heterogeneous computing approaches, but optimizing input and output data movements remains a challenge. In GPU-based workflows data already resides on GPU memory, but machine learning models requires the input and output data to be provided in specific tensor format, often requiring unnecessary copying outside of the GPU device and conversion steps. To address this, we present an interface that allows seamless conversion of Structure of Arrays (SoA) data into lists of PyTorch tensors without explicit data movement. Our approach computes the necessary strides for various data types, including scalars and rows of vectors, matrices, allowing PyTorch tensors to directly access the data on the GPU memory. The introduced metadata structure provides a flexible mechanism for defining the columns to be used and specifying the order of the resulting tensor list. This user-friendly interface minimizes the amount of code required, allowing direct integration with machine learning models. Implemented within the CMS computing framework and using the Alpaka library for heterogeneous applications, this solution significantly improves GPU efficiency. By avoiding unnecessary CPU-GPU transfers, it accelerates model execution while maintaining flexibility and ease of use.

Posters and coffee / **49**

Improving On-Chip Compression of High-Granularity Calorimeter Data with Conditional Autoencoders

Authors: Danny Noonan[1]; Erdem Yigit Ertorer[2]; Jim Hirschauer[1]; Mariel Peczak[None]; Matteo Cremonesi[2]; Nate Woodward[None]; Nhan Tran[1]; Peter Eduard Meiring[2]; Philip Coleman Harris[3]; Simon Rothman[3]; Zachary Baldwin[4]

[1] *Fermi National Accelerator Lab. (US)*

[2] *Carnegie-Mellon University (US)*

[3] *Massachusetts Inst. of Technology (US)*

[4] *Carnegie Mellon University*

The Large Hadron Collider (LHC) will soon undergo a high-luminosity (HL) upgrade to improve future searches for new particles and to measure particle properties with increased precision. The upgrade is expected to provide a dataset ten times larger than the one currently available by the end of its data-taking period. The increased beam intensity will also increase the number of simultaneous interactions per bunch-crossing (pileup), presenting greater challenges in particle reconstruction. To address this, the CMS Collaboration will replace its current endcap calorimeters with a radiation-hard high-granularity calorimeter (HGCAL). Containing six million readout channels, HGCAL will measure the energy and position of particles with unprecedented precision. However,

this level of granularity will result in a significant increase in the data rates, of the order of 5 Pb/s. These extreme rates must be reduced by several orders of magnitude within a few microseconds by the CMS trigger system in order to be feasibly processed. Front-end ECON-T application-specific integrated circuits (ASICs) perform the first-stage reduction. The current baseline applies a variable threshold that keeps only the trigger cells (TC) that surpasses it, discarding the rest. Although this baseline algorithm successfully reduces the data rates, it fails to utilize the full granularity of HGCAL. We introduce an ASIC implementation of a physics-aware conditional autoencoder (CAE) that compresses the full granularity online. The network differs from a traditional autoencoder architecture by conditioning on physically significant information within the latent space. Through this approach, we are able to efficiently encode the data before transmission off-detector where decoding can take place on dedicated FPGAs.

Posters and coffee / **50**

Efficient and Interpretable Transformers for Particle Physics

Authors: Aaron Wang[1]; Abhijith Gandrakota[2]; Elham Khoda[3]; Javier Mauricio Duarte[4]; Jennifer Ngadiuba[5]; Vivekanand Gyanchand Sahu[6]; Zihan Zhao[7]

[1] *University of Illinois Chicago (US)*

[2] *Fermi National Accelerator Lab. (US)*

[3] *University of Washington (US)*

[4] *Univ. of California San Diego (US)*

[5] *FNAL*

[6] *University of California San Diego*

[7] *UCSD*

Transformers excel at modeling correlations in LHC collisions but incur high costs from quadratic attention. We analyze the Particle Transformer using attention maps and pair correlations on the (η,ϕ) plane, revealing that Particle Transformer attention maps learn traditional jet substructure observables. To improve efficiency we benchmark linear attention variants on JetClass and find that the Linformer matches ParT accuracy with far fewer resources. We then develop a novel small linear model informed by physics that partitions particles by kinematics and uses convolutional layers to capture local features. This model outperforms the standard Linformer on HLS4ML and has much lower latency than full attention models. Sequence ordering guided by physics and embedding analysis further improve accuracy and transparency for real time collider applications.

Contributed talks / **51**

Chisel4ml: Using Chisel For Direct Circuit Implementation of Deeply Quantized Neural Networks

Author: Jure Vreča[None]

We give an introduction to chisel4ml, a tool for generating direct circuit implementations of deeply quantized neural networks. It uses structural descriptions of deeply quantized neural networks in the form of Chisel generators. Chisel is a domain-specific language for designing synchronous digital circuits. It is a language embedded in Scala that offers a wealth of powerful features, such as: functional programming, object-oriented programming, and static type safety. We will introduce you to the basics of the Chisel language and show you how chisel4ml can be used to create implementations of deeply quantized neural networks.

Posters and coffee / 52

Chisel4ml: Direct Circuit Implementation of Deeply Quantized Neural Networks

Author: Jure Vreča[None]

Chisel4ml is a tool we developed for generating fast implementations of deeply quantized neural networks. The tool has a Python frontend and a Chisel backend. The Python frontend serves as an interface to the Python ecosystem for training neural networks. The Chisel backend consists of hardware generators written in the Chisel Hardware Construction Language. This is a language embedded in Scala that offers a wealth of powerful features, such as functional programming, object-oriented programming and static type safety. The tool is capable of processing QONNX-based, deeply quantized neural networks and generating a bit-exact circuit in the form of a Verilog file.

Tutorials / 53

Designing and Deploying Low-Latency Neural Networks on FPGAs with HGQ and da4ml

Author: Chang Sun[1]

Co-authors: Jennifer Ngadiuba [2]; Maria Spiropulu [1]; Thea Aarrestad [3]; Vladimir Loncar [4]; Wayne Luk ; Zhiqiang (Walkie) Que [5]

[1] *California Institute of Technology (US)*

[2] *FNAL*

[3] *ETH Zurich (CH)*

[4] *CERN*

[5] *Imperial College London*

Neural networks with a latency requirement on the order of microseconds are widely used at the CERN Large Hadron Collider, particularly in the low-level trigger system. To satisfy this latency requirement, these neural networks are often deployed on FPGAs.

This tutorial aims to provide a practical, hands-on guide of a software-hardware co-design workflow using the HGQ2 and da4ml libraries. Comparing with existing workflows, this approach has shown to reduce the resource consumption of the resulting hardware designs by up to two orders of magnitude while maintaining the same accuracy. In particular, the following topics are covered:

1. Setup and Basic Concepts
2. Environment: We will cover installing the HGQ2 and da4ml packages via pip, configuring Keras v3 backends, and understanding the basics of numba JIT-compilation used in da4ml to avoid common pitfalls.
3. HGQ Methodology: The key concepts of HGQ will be introduced, including the use of a surrogate gradient for differentiable bit-widths and the construction of a differentiable hardware resource estimate incorporated into the loss function for efficient model training.
4. da4ml Methodology: An overview of da4ml's two-stage hybrid algorithm will be provided, including the coarse-grained graph-based reduction and the fine-grained common subexpression elimination to create multiplier-free designs. We will explain how this process aligns with HGQ's training goal by effectively reducing the number of non-zero digits in the weight matrix.
5. The Co-Design Workflow

6. Training with HGQ: We will define and train neural networks from scratch in HGQ, covering the basics of configuring fixed-point quantizers and applying HGQ to architectures ranging from simple DNNs to MLP-Mixers. Best practices for defining models that can be converted to FPGA firmware with bit-exactness will be discussed. In addition, guidance will be given on how to emulate QKeras behavior in HGQ2 when necessary.
7. Synthesis with hls4ml and da4ml: We will demonstrate how to convert an HGQ-trained model using hls4ml for bit-exact firmware generation, and explain how this is achieved in the background through a model-wise symbolic precision propagation. We will also show how to enable and configure da4ml using the distributed_arithmetic strategy in hls4ml.
8. Analysis and Advanced Techniques
9. RTL Generation: For compatible network architectures, we will explore da4ml's ability to generate fully pipelined Verilog directly from a trained model. We will also demonstrate how to verify the design's correctness with streamlined Verilator emulation.
10. Performance Review: We will analyze and compare key hardware metrics—initiation interval, latency, Fmax, and resource utilization—from both the hls4ml and standalone RTL workflows to discuss their trade-offs.
11. Tuning Techniques: We will cover more advanced techniques, such as beta scheduling or targeting a specific resource budget in HGQ2 with PID control of beta, automatically logging models on the Pareto front to explore the accuracy-resource trade-off, and debugging common issues like divergent bit-widths during conversion.

Contributed talks / **54**

A Real-Time GNN-based Clustering Algorithm for the Level 1 Calorimeter Trigger at Belle II

Authors: Isabel Haide[1]; Marc Neu[None]; Frank Baptist[2]; Thomas Lobmaier[2]; Juergen Becker[2]; Torben Ferber[3]

[1] *Karlsruhe Institute for Technology*

[2] *Karlsruhe Institute of Technology*

[3] *KIT - Karlsruhe Institute of Technology (DE)*

With increasing beam background levels at Belle II, which have already been observed due to the world-record instantaneous luminosities achieved by SuperKEKB and which are expected to rise further, an upgrade of the current Level 1 (L1) trigger algorithms is necessary to handle the evolving conditions. In this work, we present an upgraded L1 electromagnetic calorimeter trigger, based on Graph Neural Networks (GNNs) using dynamic graph building, implemented on the AMD XCVU160 FPGA used in the Belle II Universal Trigger Board 4 (UT4). The algorithm was developed in a software-hardware codesign approach, including quantization-aware training, pruning, and post-training optimizations, having both performance optimization and hardware requirements in mind. The network performs cluster finding and reconstruction in a one-shot approach, without assuming a predefined maximum number of clusters. We demonstrate an implementation of a 15-layer deep GNN with multiple graph construction and message passing steps on the FPGA. This design achieves the required throughput of 8 MHz and an overall latency of 3 μs. The implementation on the UT4 was deployed in monitoring mode within the full Belle II L1 trigger system and was included in collision data-taking in December 2024. This is the first operation of a GNN-based hardware trigger. We report implementation results showing 75% logic block usage and full utilization of DSP resources, with validation on both collision and cosmic-ray data collected with the Belle II detector.

Posters and coffee / **55**

FloatQuant: Arbitrary-Precision Minifloats in QONNX

Authors: Ebby Samson[1]; Giuseppe Franco[2]; Ian Colbert[3]; Nicholas Fraser[2]; Nicolo Ghielmetti[4]; Shane Fleming[2]; Yaman Umuroglu[2]

[1] *AMD Research, Imperial College*

[2] *AMD Research*

[3] *AMD*

[4] *CERN*

The rising popularity of large language models (LLMs) has led to a growing demand for efficient model deployment. In this context, the combination of post-training quantization (PTQ) and low-precision floating-point formats such as FP4, FP6 and FP8 has emerged as an important technique, allowing for rapid and accurate quantization with the ability to capture outlier values in LLMs without requiring extensive Quantization-Aware Training (QAT) typically needed for fixed-point formats. Nevertheless, a notable challenge remains: the gap between quantized models created by research frameworks, their standardized representation, and their compatibility with downstream tools. In this work, we help bridge this gap by enhancing the Quantized Open Neural Network Exchange (QONNX) representation format to formally include support for arbitrary-precision minifloat quantization through the newly introduced FloatQuant operator. We also propose a novel cost function for minifloat quantization to help guide quantization decisions in a cost-conscious manner, informed by the architecture of floating-point multiply-accumulate (FPMAC) nodes on FPGAs. Finally, we expand the QONNX model zoo by providing a series of example models quantized with FloatQuant to facilitate practical implementation and testing. Our contributions are upstreamed into the QONNX GitHub repositories.

Posters and coffee / **57**

Evolution of the oneAPI backend for hls4ml

Authors: Enrico Lupi[1]; Harry Wang[2]; Jovan Mitrevski[3]; Lauri Antti Olavi Laatu[4]; Paul White[2]; Suleyman Demirsoy[2]; Vladimir Loncar[5]

[1] *CERN, INFN Padova (IT)*

[2] *Altera Corporation*

[3] *Fermi National Accelerator Lab. (US)*

[4] *Imperial College (GB)*

[5] *CERN*

Since version 1.0, hls4ml has provided a oneAPI backend for Altera FPGAs, as an evolution of the backend that targeted Intel HLS. Some design choices will be presented here, including the use of pipes and task sequences to develop a dataflow-style architecture. The oneAPI framework, unlike the Intel HLS framework, also naturally supports an accelerator-style deployment. Using always-run kernels and DMA transfers, we can efficiently deploy neural networks in hardware in an accelerator style, which we will describe.

Contributed talks / **58**

Experiences Deploying a Hybrid PVFinder Algorithm for Primary Vertex Reconstruction in LHCb's GPU-Resident HLT1

Author: Mohamed Elashri[1]

Co-authors: Conor Henderson [2]; Michael David Sokoloff [2]; Simon Akar [3]

[1] *University of Cincinnati*

[2] *University of Cincinnati (US)*

[3] *LPCA - Université Clermont Auvergne (FR)*

The PVFinder algorithm employs a hybrid deep neural network (DNN) approach to reconstruct primary vertices (PVs) in proton-proton collisions at the LHC, addressing the complexities of high pile-up environments in LHCb and ATLAS experiments. By integrating fully connected layers with a UNet architecture, PVFinder's end-to-end tracks-to-hist DNN processes charged track parameters to predict PV positions, achieving efficiencies above 97% and false positive rates as low as 0.03 per event in LHCb, surpassing conventional heuristic methods. We present the current status of embedding PVFinder into LHCb's Allen framework, a fully software-based, GPU-optimized first-level trigger system for Run 3, handling 30 MHz of beam crossing data. Key challenges include optimizing computational efficiency and model integration within Allen's real-time constraints. For ATLAS, PVFinder matches the Adaptive Multi-Vertex Finder's efficiency while improving vertex-vertex resolution (0.23–0.37 mm vs. 0.76 mm). Future efforts target ATLAS ACTS integration and graph neural network enhancements.

Posters and coffee / **61**

Electron/Positron - Proton Classification with AMS ECAL using Convolutional Vision Transformers

Author: Berk Turk[1]

Co-authors: Bilge Demirkoz [1]; Emre Akbas [1]; Zhili Weng [2]

[1] *Middle East Technical University (TR)*

[2] *Massachusetts Inst. of Technology (US)*

Alpha Magnetic Spectrometer (AMS-02) is a precision high-energy cosmic-ray experiment consisting of Transition Radiation Detector (TRD), Silicon Tracker, Magnet, Time of Flight (ToF), and Ring Imaging Cherenkov Detector (RICH), Anti-Coincidence Counter (ACC), and Electromagnetic Calorimeter (ECAL) on the ISS operating since 2011, and has collected more than 240 billion cosmic-ray events. Among them, positrons are important in understanding the particle nature of dark matter if it annihilates into positrons. Separating the positron signals from the abundant background protons is challenging because the protons can also produce showers with an electromagnetic component in the ECAL via neutral pion decays, mimicking electron/positron showers. Therefore, we use a state-of-the-art convolutional and transformer model, such as CoAtNet, that employs all energy depositions in the ECAL cells, represented in matrix format as images, to classify the electrons/positrons in the dominant cosmic proton background. We created training, validation, and test datasets for CoAtNet, in which purer samples of electrons and protons from ISS data and test beam data in the 50-200 GeV reconstructed energy range are selected by using information from Tracker, ToF, and TRD. To reject charge confused protons from the electron sample and create datasets in the 200-500 GeV range, we built an additional charge-confused proton and electron classifier with a neural network using Tracker, ToF, and RICH information. We trained and fine-tuned the ECAL CoAtNet using the datasets and obtained a proton rejection that outperforms the traditional methods by more than a factor of 2 on ISS Data in the 50-500 GeV range.

Posters and coffee / **64**

High-Throughput Ghost Track Rejection with Deep Learning at LHCb

Authors: Alvaro Fernandez Casani[1]; Arantza De Oyanguren Campos[1]; Jiahui Zhuo[1]; Valerii Kholoimov[2]

[1] *Univ. of Valencia and CSIC (ES)*

[2] *Instituto de Física Corpuscular (Univ. of Valencia)*

The LHCb experiment at CERN operates a fully software-based first-level trigger that processes 30 million collision events per second, with a data throughput of 4 TB/s. Real-time tracking—reconstructing particle trajectories from raw detector hits—is essential for selecting the most interesting events, but must be performed under tight latency and throughput constraints.
A key bottleneck in this process is the suppression of "ghost" tracks: false combinations of detector hits caused by noise or detector ambiguities. To meet these challenges, a dedicated neural network architecture has been developed and optimized for fast inference on GPUs. Its speed is critical—enabling high-quality tracking to run within the strict timing budgets of the trigger system. The network achieves very good physics performance with ghost rates below 20%. The techniques employed can be used for real-time ML in similarly demanding environments.

Contributed talks / 65

Real-Time Anomaly Detection in the CMS Level-1 Trigger with AXOL1TL

Authors: CMS Collaboration[None]; Sabrina Giorgetti[1]

[1] *Universita e INFN, Padova (IT)*

AXOL1TL is an anomaly detection (AD) trigger algorithm integrated into the Global Trigger (GT) of the CMS Level-1 Trigger (L1T) system since 2024. The GT reduces the event rate from proton–proton collisions at the LHC, lowering it from 40 MHz to 100 kHz within a fixed latency of 50 ns. The AD algorithm, implemented in the FPGA firmware of the GT board, uses an autoencoder to assign an anomaly score to each event, enabling the selection of more anomalous events for further analysis. We present the full deployment workflow to achieve ultra-low-latency anomaly detection: from hardware-aware model training to firmware synthesis and integration into the L1T system. We also report on the characterisation and performance of the AXOL1TL trigger, using the latest model, updated in 2025 with an increased rate budget and a novel feature extraction technique, based on a self-supervised method, which led to improved performance. This work demonstrates one of the first fully functional anomaly detection triggers within the CMS L1T system and showcases how novel trigger-level approaches can enhance sensitivity to new physics in real-time event selection.

Posters and coffee / 66

FPGA-Optimized ML for Fast electron identification and pT regression with the CMS Phase-2 L1 trigger

Author: Piero Viscone[1]

[1] *CERN & University of Zurich (CH)*

In preparation for the High Luminosity LHC (HL-LHC) run, the CMS experiment is developing a major upgrade of its Level-1 (L1) Trigger system, which will integrate high-granularity calorimeter data and real-time tracking using FPGA-based processors connected via a high-bandwidth optical network. A central challenge is the identification of electrons in a high pileup environment within strict latency and resource constraints.
The contribution will focus on the identification of electron signatures with high efficiency and energy resolution. In this context, a machine learning–based algorithm that combines calorimeter and tracking information to perform electron identification in a single inference step is presented. The model is implemented with fixed-point arithmetic and optimized for fast, resource-efficient execution on FPGAs. A lightweight regression model to estimate the transverse momentum of the identified electrons, enhancing energy resolution, is also discussed. The improved energy accuracy is particularly relevant in the context of L1 scouting, where enriched event information is streamed for real-time analysis. The impact on a benchmark on a low-mass dark photon to dielectron resonance analysis is demonstrated.

Posters and coffee / **67**

Highly Granular Quantization for CICADA

Authors: Abhishikth Mallampalli[1]; Isobel Ojalvo[2]; Jennifer Ngadiuba[3]; Lino Oscar Gerlach[2]

[1] *University of Wisconsin Madison (US)*

[2] *Princeton University (US)*

[3] *FNAL*

The CICADA (Calorimeter Image Convolutional Anomaly Detection Algorithm) project aims to detect anomalous physics signatures without bias from theoretical models in proton-proton collisions at the Compact Muon Solenoid (CMS) experiment at the Large Hadron Collider. CICADA identifies anomalies in low-level calorimeter trigger data using a convolutional autoencoder, whose behavior is transferred to a compact model via knowledge distillation. Careful quantization allows the model to meet the requirement of sub-200ns inference times on FPGAs.
While current implementations use Quantization Aware Training with per-layer quantization schemes, a gradient-based quantization optimization method introduced in 2024 enables fine-grained, parameter-level quantization. We present a novel CICADA implementation leveraging this highly granular quantization. Evaluated on CMS open data, it achieves comparable anomaly detection performance with significantly reduced resource usage in emulated FPGA conditions. We also demonstrate the feasibility of analyzing calorimeter energy deposits at finer spatial granularity while maintaining CMS's stringent latency and hardware constraints for real-time data selection.

Posters and coffee / **68**

EveNet: Towards a Generalist Event Transformer for Unified Understanding and Generation of Collider Data

Authors: Bai-Hong Zhou[1]; Ben Nachman[2]; Haoran Zhao[3]; Qibin Liu[4]; Shih-Chieh Hsu[5]; Shu Li[6]; Ting-Hsiang Hsu[7]; Vinicius Massami Mikuni[2]; Wei-Po Wang[3]; Yuan-Tang Chou[3]; Yue Xu[3]; Yulei Zhang[3]

[1] *Tsung-Dao Lee Institute (CN)*

[2] *Lawrence Berkeley National Lab. (US)*

[3] *University of Washington (US)*

[4] *SLAC National Accelerator Laboratory (US)*

[5] *University of Washington Seattle (US)*

[6] *Tsung-Dao Lee Institute, Shanghai Jiao Tong Univ. (CN)*

[7] *National Taiwan University (TW)*

With the increasing size of the machine learning (ML) model and vast datasets, the foundation model has transformed how we apply ML to solve real-world problems. Multimodal language models like chatGPT and Llama have expanded their capability to specialized tasks with common pre-train. Similarly, in high-energy physics (HEP), common tasks in the analysis face recurring challenges that demand scalable, data-driven solutions. In this talk, we present a foundation model for high-energy physics. Our model leverages extensive simulated datasets in pre-training to address common tasks across analyses, offering a unified starting point for specialized applications. We demonstrate the benefit of using such a pre-train model in improving search sensitivity, anomaly detection, event reconstruction, feature generation, and beyond. By harnessing the power of pre-trained models, we could push the boundaries of discovery with greater efficiency and insight.

Posters and coffee / **70**

Flexibly Equivariant Generative framework for Stochastic subgrid Turbulence modeling

Authors: Sharvaree Vadgama[1]; Julia Balla[2]; Ryley McConkey[2]; Abigail Bodner[2]; Erik Bekkers[1]; Tess Smidt[2]

[1] *University of Amsterdam*

[2] *MIT*

Introduction
Accurate climate prediction hinges on the ability to resolve multi-scale turbulent dynamics in the atmosphere and oceans 1. An important mechanism of energy exchange between the ocean and the atmosphere is mesoscale turbulence, which contains motions of length scale $\mathcal{O}$(100 km). Two-layer quasi-geostrophic (QG) simulations 2 are a popular technique for numerically simulating these motions due to their computational tractability. QG models strike a balance between computational cost and physical fidelity by resolving large-scale turbulent processes at the scale of the discretization grid and modeling smaller subgrid-scale processes. However, the accuracy and generalizability of QG simulations greatly depend on the subgrid-scale model. Recently, machine learning has been introduced as a tool to learn effective stochastic subgrid-scale turbulence models 2.

In this work, we propose a novel approach for using geometric generative models to learn subgrid-scale closure schemes. By leveraging the geometric structure inherent in geophysical flows and the representational power of modern generative models, we aim to capture the essential physics of scale interactions more faithfully than conventional parameterizations, enabling more accurate and generalizable QG subgrid-scale models.

Problem Statement *Predict closure term within a two-layer quasi-geostrophic fluid simulation of an idealized ocean system through a generative approach.*

Given a potential vorticity field q with horizontal geostrophic velocity field u and a coarse-graining operator $\overline{(\cdot)}$, the task is to predict the unresolved subgrid forcing from the coarsely resolved (low-pass filtered) field:
$S_q = \overline{(u \cdot \nabla q)} - (\,\overline{u} \cdot \nabla \overline{q}\,)$
where $\overline{q}$ is the coarsely resolved field.

Method
Variational autoencoders (VAEs) 3 have been shown to be useful for encoding images in compressed representations as well as powerful generative models for images. Various works in literature have incorporated group symmetries to VAEs. In this work, we propose an equivariant encoder-decoder framework that maintains the training simplicity of standard VAEs while enabling the generation of samples that respect underlying group symmetries through 4. In our approach, we incorporate adjustable sensitivity to equivariant constraints, allowing for flexible control over the strictness of symmetry adherence during learning.

Dataset Ground truth training data is generated using a $256 \times 256 \times 2$ grid in pyqg *(256 horizontal resolution, 2 layers)*, considered 'high resolution' ground truth data. For more details, see 5.

Results
In this figure, we see a comparison of performance for standard VAE (without any group symmetries) compared with our model with different group symmetries (C4, SO(3)) with and without regularized equivariant loss.

References

1 Brunton, Steven L. et al, "Machine learning for fluid mechanics." Annual review of fluid mechanics, 2020

2 Perezhogin, Pavel et al "Generative data-driven approaches for stochastic subgrid parameterizations in an idealized ocean model." Journal of Advances in Modeling Earth Systems, 2023.

3 Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes" 2013.

4 Geiger, Mario et al "e3nn: Euclidean neural networks", 2022.

5: Ross, Andrew et al. "Benchmarking of Machine Learning Ocean Subgrid Parameterizations in an Idealized Model." Journal of Advances in Modeling Earth Systems, 2023.

Tutorials / **72**

NeuraLUT-Assemble: Hardware-aware Assembling of Sub-Neural Networks for Efficient LUT Inference

Authors: Marta Andronic[1]; Oliver Cassidy[1]; George A. Constantinides[1]

[1] *Imperial College London*

Neural networks (NNs) have gained significant interest in recent years due to their prevalence in AI applications. Lookup table (LUT) based NN architectures have emerged as a promising solution for ultra-low latency inference on reconfigurable hardware such as field programmable gate arrays (FPGAs). These techniques promise significant enhancements in both resource efficiency and inference speed, and have been shown to be effective across multiple latency-critical domains such as particle physics. However, existing LUT-based designs suffer from accuracy degradation due to the large fan-in required by neurons being limited by the exponential scaling of LUT resources with input width. In practice, in prior work this tension has resulted in the reliance on extremely sparse models.

In this tutorial, we will mainly demonstrate our latest work called NeuraLUT-Assemble, a state-of-the-art framework that addresses these limitations by combining mixed-precision techniques with the assembly of larger neurons from smaller units, thereby increasing connectivity while keeping the number of inputs of any given LUT manageable. NeuraLUT-Assemble closes the accuracy gap between LUT-based methods and (fully-connected) MLP-based models, achieving competitive accuracy on tasks such as network intrusion detection, digit classification, and jet classification. Additionally, we will illustrate NeuraLUT-Assemble's efficiency through a live demonstration of the network implemented on a physical FPGA.

At the beginning of our tutorial, we will walk through the prior work that led to NeuraLUT-Assemble. We will start by reviewing two broad families of ultra-low-latency FPGA inference: (1) methods that learn the LUTs directly and (2) methods that train traditional neural networks with constraints so they can later be fully mapped into LUTs. For the first category, we will briefly introduce LUTNet and DWNNs, which make LUT behavior differentiable by approximating gradients. In the second category, we will cover LogicNets, which maps sparse quantized networks to LUTs but suffers from limited fan-in; PolyLUT, which increases expressivity by embedding multivariate polynomials into neurons; and NeuraLUT, which hides small MLPs inside each LUT to combine flexibility with trainability. We will also touch on ReducedLUT, a complementary technique that performs post-training logic minimization by exploiting don't-care conditions to further compress LUTs.

After covering this background, we will introduce the methodology behind NeuraLUT-Assemble. We will explain the tree-based assembling strategy in detail, showing how it enables the construction of larger fan-in neurons by composing them from smaller, manageable L-LUTs. This section will include an intuitive breakdown of different assembly configurations, with visual diagrams and comparisons to traditional approaches. Next, we will walk through the training process, including the use of quantization-aware training, skip connections embedded within LUTs, and the structured pruning strategy used to determine connectivity. We will then present a hands-on demonstration of the full toolflow—from network specification in PyTorch to Verilog generation, FPGA synthesis, and deployment. The tutorial will conclude with an evaluation of NeuraLUT-Assemble across tasks such as MNIST and jet classification, highlighting empirical trade-offs in accuracy and area-delay product. Throughout, we aim to provide practical insight into designing, training, and deploying efficient hardware-aware neural networks using LUT-based techniques on FPGAs.

Contributed talks / **73**

FPGA-accelerated ML for real-time RHEED inference

Authors: Abdelrahman Asem Elabd[1]; Bo-Cheng Lai[None]; ChiJui Chen[2]; Geoff Jones[3]; Joshua Agar[4]; Matt Wilkinson[5]; Pujan Patel[3]; Scott Hauck[None]; Shih-Chieh Hsu[6]; Yu-Hao Hu[7]

[1] *University of Washington (US)*

[2] *National Yang Ming Chiao Tung University*

[3] *Department of Electrical and Computer Engineering, University of Washington*

[4] *M3 Learning*

[5] *University of Washington*

[6] *University of Washington Seattle (US)*

[7] *Graduate Degree Program of AI, National Yang Ming Chiao Tung University*

Reflection High-Energy Electron Diffraction (RHEED) is a common diffraction-based surface characterization technique for analyzing the properties of crystalline materials that are grown using a thin-film deposition technique like pulsed-laser deposition (PLD) or molecular-beam epitaxy (MBE). In this work, we design an FPGA-accelerated machine learning (ML) algorithm to perform real-time analysis of RHEED data, allowing us to track the growth process of a thin-film deposition sample in real-time. This enables future study of the dynamic, high-speed interactions between the applied material (e.g. plasma, gaseous atomic or molecular plume) and the sample being grown, and lays the groundwork for future development in optimal control of thin-film deposition techniques.

Our ML solution consists of 2 two standard CNNs in sequence: (a) an upstream object-detection CNN to locate the local maxima ("diffraction spots") in a RHEED diffraction pattern, followed by (b) a downstream regression CNN to parametrize the 'shape'of each diffraction spot. We use a combination of real, augmented, and synthetic training data to produce models which are accurate, generalizable and robust to out-of-distribution samples. Through a combination of quantization-aware training, pruning, and neural architecture search, we produce models which are both small (in terms of onboard FPGA resource usage) and fast (in terms of inference latency and throughput). We convert these CNNs into FPGA designs using the hls4ml translation tool, and integrate them into a higher-level 'inference block'design, which handles data I/O, image-cropping and normalization, and block-level control of each CNN. Finally this inference block is linked to a Phantom S200 high-speed camera using the Euresys FrameGrabber FPGA design.

We simulate the top-level FPGA design to show that this hardware-accelerated ML solution has sufficient accuracy, resource-usage, and latency to retrieve scientifically useful results from a real thin-film deposition growth process.

Posters and coffee / **74**

Fast Synthetic X-Ray Generation for AI Diagnostics: DALL·E vs. Stable Diffusion

Author: RUKSHAK KAPOOR[None]

Medical imaging is foundational to clinical diagnostics and biomedical research, enabling the identification and monitoring of a wide range of conditions—from pulmonary diseases to cancer. However, the development of high-performance AI diagnostic systems is often hampered by restricted access to large, diverse, and well-annotated imaging datasets. This limitation is particularly acute for rare diseases, where data scarcity is compounded by patient privacy regulations, high acquisition costs, and the need for expert annotation.

To address these challenges, synthetic medical image generation has emerged as a compelling approach. By producing diagnostically relevant artificial samples, generative AI models can supplement real-world datasets, thereby improving model robustness, class balance, and training efficiency —without compromising data privacy.

In this work, we propose a comparative evaluation of two state-of-the-art generative models—DALL·E, a proprietary transformer-based text-to-image model developed by OpenAI, and Stable Diffusion, a leading open-source latent diffusion model. Our focus is on the generation of high-fidelity synthetic chest X-ray images for respiratory disease categories including COVID-19, tuberculosis, and pneumonia. These disease domains are selected due to their clinical significance and the frequent lack of balanced, high-quality data in public repositories.

Our methodology involves generating condition-specific images using both models based on curated prompts or class labels and integrating these synthetic images into the training pipelines of deep learning classifiers. We will evaluate the diagnostic performance of models trained on real-only data, synthetic-only data, and hybrid datasets, using metrics such as accuracy, precision-recall, and F1-score. Additionally, image quality assessments will include perceptual metrics (e.g., SSIM, FID) and expert review, where feasible.

Beyond performance evaluation, we will examine trade-offs between model accessibility, customization capabilities, generation speed, and deployment efficiency, highlighting practical considerations for integrating such tools into medical ML pipelines. Special attention will be given to the feasibility of using these generative models in fast and resource-constrained environments.

By framing synthetic medical image generation through the lens of fast and scalable machine learning, this study contributes to the broader goal of developing privacy-conscious, reproducible, and robust diagnostic systems, especially in data-scarce settings. The resulting code, models, and analysis pipeline will be made publicly available to support future research and collaborative efforts in the medical AI community.

Posters and coffee / **77**

Enhancing Transit Detection to Uncover Long-Period Exoplanets

Author: David Degen[1]

[1] *ETH Zurich, Queloz Group*

Small ($R < 4\,\mathrm{R}_\oplus$), long-period ($30\,\mathrm{days} < P$) exoplanets with low equilibrium temperatures are an extremely interesting population, promising insights into planet formation, atmospheric chemistry and evolution, as well as habitability. However, for these planets, the current observing strategy of NASA's Transiting Exoplanet Survey Satellite (TESS) can only capture single transit events in different sectors, if any at all, as the baselines of observation are generally less than 30 days. Moreover, traditional detection methods, like median filtering followed by Box Least Squares, prove comparatively ineffective at finding planets with only a few shallow transits observed.

To address these challenges, we have developed an automated detection pipeline that integrates machine learning-based transit detection with robust vetting techniques to find long-period exoplanets more efficiently. When we identify two or more individual transit events at a star that are sufficiently similar, we assume they stem from the same planet, which limits the compatible periods to a set of aliases and enables cost-effective follow-up observations by the CHaracterising ExOPlanets Satellite (CHEOPS, Swiss Space Office & ESA). We have compared the performance of fully convolutional networks, U-Nets and other architectures in detecting transit signals. We have also investigated the advantages of incorporating features in addition to stellar flux, and the influence of the receptive field on the performance of the models.

When applied to the full TESS data set, we anticipate dozens of newly-detected candidate sub-Neptunes on long periods, providing critical data for investigating the characteristics and distribution of such planets.

Contributed talks / **79**

SparsePixels: Efficient Convolution for Sparse Data on FPGAs

Author: Ho-Fung Tsoi[1]

Co-authors: Dylan Sheldon Rankin [2]; Vladimir Loncar [3]; Philip Coleman Harris [4]

[1] *University of Pennsylvania*

[2] *University of Pennsylvania (US)*

[3] *CERN*

[4] *Massachusetts Inst. of Technology (US)*

Inference of standard convolutional neural networks (CNNs) on FPGAs often incurs high latency and long initiation intervals due to the nested loops required to slide filters across the full input, especially when the input dimensions are large. However, in some datasets, meaningful signals may occupy only a small fraction of the input, say sometimes just a few percent of the total pixels or even less. In such cases, most computations are wasted on regions containing no useful information. In this work, we introduce SparsePixels, a framework for efficient convolution for sparsely populated input data on FPGAs operating under tight resource and low-latency constraints. Our approach implements a special class of CNNs where only active pixels (non-zero or above a threshold) are retained and processed at runtime, while the inactive ones are discarded and ignored. We show that our framework can achieve performance comparable to standard CNNs in some target datasets while significantly reducing both latency and resource usage on FPGAs. Custom kernels for training and the HLS implementation are developed to support sparse convolution operations.

Tutorials / **82**

Super Neural Architecture Codesign Package (SNAC-Pack)

Authors: Dmitri Demler[None]; Jason Weitz[1]

Co-authors: Ben Hawks [2]; Javier Mauricio Duarte [1]; Nhan Tran [3]

[1] *Univ. of California San Diego (US)*

[2] *Fermi National Accelerator Lab*

[3] *Fermi National Accelerator Lab. (US)*

Machine learning has become a critical tool for analysis and decision-making across a wide range of scientific domains, from particle physics to materials science. However, the deployment of neural networks in resource-constrained environments, such as hardware accelerators and edge devices, remains a significant challenge. This often requires specialized expertise in both neural architecture design and hardware optimization.

To address this challenge, we introduce the Super Neural Architecture Codesign Package (SNAC-Pack), an integrated framework that automates the discovery and optimization of neural network architectures specifically tailored for hardware deployment. SNAC-Pack combines two powerful tools: Neural Architecture Codesign, which performs a two stage neural architecture search for optimal models, and the Resource Utilization and Latency Estimator, which predicts how an architecture will perform when implemented on FPGA software.

SNAC-Pack streamlines the neural architecture design process by enabling researchers to automatically explore diverse architectures optimized for both task performance and hardware efficiency. By providing quick estimates of resource utilization and latency without requiring time-consuming synthesis, SNAC-Pack accelerates the development cycle. State-of-the-art compression techniques, such as quantization-aware training and pruning, further optimize the models, resulting in architectures that can be deployed to FPGA hardware.

This tutorial provides a hands-on introduction to SNAC-Pack, guiding participants through the complete workflow from dataset preparation to hardware deployment. By the end of the tutorial, attendees will be able to run SNAC-Pack for their own applications, achieving improvements in accuracy, latency, and resource utilization compared to naive hand-crafted approaches.

Contributed talks / **84**

Low-Latency On-Chip Tau Event Selection with Machine Learning for the Belle II Level-1 Trigger

Author: Deven Misra[1]

Co-authors: Taichiro Koga [2]; Yu Nakazawa [2]; Takeo Higuchi [3]

[1] *University of Tokyo*

[2] *KEK*

[3] *Kavli IPMU*

Belle II is a luminosity frontier experiment located at the SuperKEKB asymmetric e^+e^- collider, operating at the $\Upsilon(4S)$ resonance. The τ physics program at Belle II involves both probes of new physics and precision measurements of standard model parameters with large statistics. SuperKEKB is projected to reach a luminosity of 6×10^{35} cm^{-2}s^{-1} in the next decade. At these high luminosities, the hardware-based Level-1 Trigger system will require improved signal identification algorithms maintain high trigger efficiencies while keeping the total trigger rate below the data acquisition system limit of 30 kHz. Utilizing per-weight mixed-precision quantization aware training, we develop a fast machine-learning based logic for τ event selection with ~ 100 ns latency, implemented on an AMD XCVU080 FPGA. Our algorithm uses energy, timing, and position information provided by the electromagnetic calorimeter sub-trigger system as inputs to a feed-forward dense neural network to reconstruct low-multiplicity standard model τ decays. When compared with common trigger conditions currently used for τ selection, we achieve up to 50% reduction in total trigger rate while maintaining over 95% signal efficiency. The new firmware has been validated using cosmic ray data collected in early 2025, and is now implemented in the Belle II analysis software framework for further validation in simulation. Full implementation of the new logic is planned for the next Belle II physics run in fall 2025.

Contributed talks / 85

GELATO: A Generic Event-Level Anomaly detection Trigger for ATLAS

Author: Kenny Jia[1]

[1] *Stanford University/ SLAC*

The absence of BSM physics discoveries at the LHC suggests new physics could lie outside current trigger schemes. By applying unsupervised ML–based anomaly detection, we gain a model-agnostic way of spotting anomalous signatures that deviate from the current trigger's expectations. Here we introduce a Run-3 trigger chain that embeds fast anomaly detection algorithms in both hardware and software levels. We will describe its design, integration, commissioning strategy with emphasis on rate stability and robustness, and show first validation results from its data stream. This marks ATLAS's inaugural anomaly-detection trigger, laying the groundwork for future ML-driven triggers and novel sensitivity to a broad spectrum of new-physics signatures in Run-3 and beyond.

Contributed talks / 86

Hierarchical Dataflow Accelerator of Interaction Networks for Large-Scale Particle Tracking on FPGA

Author: Yuru Su[None]

Co-authors: Bo-Cheng Lai ; Hsuanwei Yu [1]; Mark Neubauer [2]; Santosh Parajuli [2]; Shih-Chieh Hsu [3]

[1] *NYCU*

[2] *Univ. Illinois at Urbana Champaign (US)*

[3] *University of Washington Seattle (US)*

The Interaction Network (IN) algorithm has shown great promise for particle tracking applications at the Large Hadron Collider (LHC), where identifying complex particle trajectories from raw detector data is a computationally intensive task. IN leverages graph-based representations of detector hits to learn relationships between particle interactions, making it well-suited for this domain. Given the extremely high data rates and stringent latency requirements in the LHC environment, Field-Programmable Gate Arrays (FPGAs) present an ideal platform for deploying IN-based inference systems, thanks to their low-latency, parallel-processing capabilities and energy efficiency.

However, existing FPGA implementations of the IN algorithm are constrained by the limited on-chip resources available to handle large-scale input graphs. As a result, they typically rely on graph reduction or subgraph sampling techniques that risk discarding important structural information and compromising the overall fidelity of the dataset. These limitations hinder the deployment of IN models in real-time applications that require both high throughput and high accuracy across the entire dataset.

To address these challenges, we propose a modular and hierarchical dataflow accelerator tailored for efficient large-scale graph processing on FPGA hardware. The proposed architecture introduces a novel parallel pipeline design, which enables fine-grained concurrency across computation stages, significantly improving processing efficiency. To further enhance performance, the architecture incorporates optimized utilization of on-chip memory, reducing access latency and mitigating data transfer bottlenecks during graph traversal and edge feature computation. The use of stream-based processing allows for continuous ingestion and processing of graph data without requiring full graph buffering, which is critical for real-time applications.

A key feature of our design is its hierarchical task-level structure, which facilitates scalability by decomposing large graphs into manageable subgraphs and distributing the computation across multiple processing units. This not only improves the overall throughput but also enables modular expansion of the architecture to accommodate larger and more complex datasets. Such a design makes the system adaptable to future upgrades in detector resolution and data complexity, ensuring long-term applicability.

We implement our design on a Xilinx Virtex UltraScale+ XCVU9P FPGA and evaluate its performance using representative large-scale graph workloads relevant to LHC tracking tasks. Experimental results show that our accelerator achieves a 129.1× speedup over a CPU baseline and a 7.16× improvement over a GPU implementation. Furthermore, the proposed design demonstrates significant energy efficiency advantages, with up to 972× improvement over CPU and 195× over GPU in terms of energy consumed per inference.

Contributed talks / 88

Toward a Ultra-Fast, Energy-Efficient Readout of Calorimeters with Neuromorphic Processing

Authors: Enrico Lupi[1]; Tommaso Dorigo[2]; Alessandro Breccia[3]; Riccardo Carroccio[3]

Co-authors: Abhishek [4]; Max Aehle [5]; Muhammad Awais [6]; Long Chen [7]; Abhijit Das [8]; Andrea De Vita [9]; Nicholas Ralph Gauger [7]; Ralf Keidel [10]; Jan Kieseler [11]; Anders Mikkelsen [8]; Federico Nardi [12]; Xuan Tung Nguyen [13]; Fredrik Sandin [14]; Kylian Schmidt [11]; Pietro Vischia [15]; Joseph Willmore [16]

[1] *CERN, INFN Padova (IT)*

[2] *Luleå University of Technology, INFN Padova, MODE Collaboration, Universal Scientific Education and Research Network*

[3] *University of Padova*

[4] *National Institute of Science Education and Research Jatni*

[5] *(University of Kaiserslautern-Landau (RPTU), MODE Collaboration*

[6] *INFN Padova, Luleå University of Technology, MODE Collaboration*

[7] *University of Kaiserslautern-Landau (RPTU), MODE Collaboration*

[8] *Lund University*

[9] *INFN Padova, University of Padova*

[10] *Karlsruhe Institute of Technology, MODE Collaboration*

[11] *Karlsruhe Institute of Technology*

[12] *University of Padova, Laboratoire de Physique Clermont Auvergne*

[13] *INFN Padova, University of Kaiserslautern-Landau (RPTU)*

[14] *Luleå University of Technology, MODE Collaboration*

[15] *University of Oviedo, MODE Collaboration, Universal Scientific Education and Research Network*

[16] *INFN Padova*

Hadronic calorimeters are a key part of high energy physics experiments. Traditionally, they rely on high granularity to improve performances, but this leads to various challenges in terms of cost, energy consumption and output data volume. Moreover, current detectors do not have the capability of exploiting temporal information of the shower development, as the time frame for pattern detection is condensed to sub-nanoseconds due to the particles' speed. Neuromorphic architectures might help with overcoming these limitations.

We explore a neuromorphic approach to calorimeter readout. Hadrons interacting with a homogeneous lead-tungstate ($PbWO_4$) calorimeter are simulated and the resulting time-dependent light signals, captured by a dense array of photodetectors, are encoded into spike trains to serve as input to a fully connected spiking neural network (SNN). This architecture is trained to reconstruct key physical quantities, including the total energy deposited and topological information about the events. The model performs effectively in both single- and multi-task regression settings, producing consistent results in both scenarios.

Finally, we discuss the feasibility of implementing such a readout system using nanophotonic hardware based on III-V semiconductor nanowires, highlighting a pathway toward ultra-fast, energy-efficient calorimeter readout.

Posters and coffee / 89

Scientific Machine Learning for Symbolic Recovery of Relativistic Effects in Black Hole Orbits

Author: Pothuraju Naveen Yadav[1]

Co-authors: Prathamesh Dinesh Joshi [2]; Raj Abhijit Dandekar [2]; Rajat Dandekar [2]; Sreedath Panat [2]; Dinesh Kumar Vishwakarma [1]

[1] *Delhi Technological University*

[2] *Vizuara AI Labs*

Simulating relativistic orbital dynamics around Schwarzschild black holes is essential for understanding general relativity and astrophysical phenomena like precession. Traditional numerical solvers face difficulty while dealing with noisy or sparse data, necessitating data-driven approaches. We develop a Scientific Machine Learning (SciML) framework to model orbital trajectories and symbolically recover the relativistic correction term. Neural Ordinary Differential Equations (Neural ODEs) accurately predict inverse radius u, radial velocity v, and precession δ, performing well under ideal conditions. However, performance degrades with limited data or poor data quality.

To address this, we employ Universal Differential Equations (UDEs), which embed a neural network to approximate the correction term $\alpha \frac{GM}{c^2} u^3$, achieving precise orbit predictions even under challenging conditions. Symbolic regression further recovers an analytical expression closely matching

the expected correction, with minimal error in fitted coefficients. We use adjoint-based training to quickly discover the best solution, and we achieve this in Julia with DiffEqFlux and Lux.

Using this method, we successfully demonstrate the ability to collect useful information for challenging assets by integrating machine learning with physical laws. This work can be further expanded for large-scale or detailed astrophysical projects.

Contributed talks / 90

Embedding domain knowledge: Inductive biases for algorithmic alignment in Machine Learning

Author: Serio Angelo Maria Agriesti[1]

Co-author: Francisco Camara Pereira [1]

[1] *Department of Technology, Management and Economics, Technical University of Denmark, Lyngby, Denmark*

Most of the current machine learning (ML) applications are purely data-driven solutions with little considerations for underlying problem dynamics, limited to in-distribution applications. To tackle this limitation a stream of literature is emerging to address out-of-distribution (OOD) performance: Algorithmic alignment, which focuses on embedding algorithmic structures into ML architectures to reflect the inherent logic of the problem at hand. The general idea is summarized in two steps: first, we formalize the dynamics, mathematical workings involved, as well as constraints and assumptions on data, outputs and parameters; second, we design the corresponding ML algorithm that maximally replicates such specification, i.e., we implement its inductive bias (IB).

The relatively recent literature of algorithmic alignment, however, shows a lack of proper characterization of existing algorithms and IBs.

We provide said characterization for our core research focus: acceleration of large-scale scientific simulators. We hypothesise that these have already mathematically embedded domain knowledge, as a result of widely detailed physical phenomena and decades of development of dedicated algorithms. Examples would be sequential, discrete-event simulators such as traffic ones or physically characterized ones such as the ones from hydrodynamics or climate science. The approach is detailed to be transferable to other scientific disciplines and facilitating the application of algorithmic reasoning in ML solutions. We analyze 3 main subjects: traditional inductive biases in ML and how they align with simulators; unconventional inductive biases inspired by the domain knowledge and generalizing power of simulators; algorithmic structures from the most common algorithms in large-scale simulators.

Our analysis of such multidisciplinary perspectives will result in a dictionary of IBs and their connections to specific tasks, which shall guide researchers and practitioners towards more robust ML solutions. We have already characterized simulators in the traffic domain and identified characteristic features of algorithms (such as Dijkstra, Frank Wolfe) and models (like the Cell Transmission Model or Elastic Traffic Assignment) in simulators. Some relevant trends are identified. We find, for example, that more simulators lean on addition and short term memory rather than multiplication or long term memory. The complexity of the simulation at hand is also a strong indicator of modularity, loop invariance and smoothness bias. Most simulators appear to deal with structural sparsity through computational power alone, but few more targeted models actually avoid it. Most simulators can also be separated in multiple components with different IBs and algorithmic structures, whose understanding is critical in designing ML meta-models able to perform OOD as well as current simulators.

The study is designed to be a toolbox for novel efficient architectures embedding specific generalization preferences that mirror the identified IBs. The analysis of common algorithmic structures in simulators will allow to more easily design ML models algorithmically aligned with the problem at hand, isolating non-nonlinearities and improving the expressiveness of the novel architectures. The findings shall be easily replicable to other scientific domains with a huge body of domain knowledge

historically already embedded in models, algorithms and simulations. A practical example may be presented.

Posters and coffee / **91**

TinyML-Based Early Detection of Pak Choi Leaf Diseases Using FPGA in Vertical Farming Environments

Author: PURABI MAZUMDAR.[1]

Co-authors: Gwo Rong Wong [1]; Jun Lun Chai [2]; Kausalyaa Kaliapan [1]; Muhammad Hakimi Bin Jaffar Jaffar [1]; Narayanan Ramakrishnan [2]

[1] *Centre for Research in Biotechnology for Agriculture, Universiti Malaya, Kuala Lumpur, Malaysia*

[2] *Electrical and Computer Systems Engineering, Monash University Malaysia*

Pak choi (*Brassica rapa subsp. chinensis*) is a leafy green vegetable widely cultivated in vertical urban farming systems due to its rapid growth and high yield under compact, hydroponic setups. However, even in these controlled environments, crops remain susceptible to various diseases. Among the most common threats are fungal infections such as Alternaria leaf spot and powdery mildew, and bacterial infection such as soft rot caused by *Pectobacterium carotovorum*. These diseases can spread rapidly and lead to significant crop loss if not identified and addressed early.
In our ongoing work, we are developing an autonomous ultraviolet (UV) robotic platform tailored for use in vertical farming environments. The robot is equipped with an integrated camera system designed for real-time disease monitoring using visual cues. Beyond detection, the robot is also intended to deliver targeted UV-C radiation to affected plant areas as a potential method for pathogen suppression or control. This integrated approach aims to reduce manual labour and targeted treatment as an alternative to chemical pesticide.
To support and validate the detection system, molecular analysis (ITS sequencing for fungus and 16sRNA sequencing for bacteria) were conducted to identify the pathogens responsible for observed symptoms. These confirmed samples are then used to build a curated dataset of annotated images for machine learning model development.
Machine learning models are being trained using this dataset and optimized for TinyML deployment, using TensorFlow Lite for Microcontrollers (TFLite Micro). The target hardware platform is an Intel Cyclone IV E FPGA, which integrates a Nios soft processor for on-device inference. The system also explores hardware-accelerated pre-processing using Sobel edge detection to highlight morphological features associated with disease progression, reducing model complexity while enhancing prediction accuracy.
The poster will present preliminary prototype results, machine learning training parameters, and hardware-software integration details. The conference provides a valuable platform for engaging with experts in machine learning and embedded systems to exchange ideas, identify optimization strategies, and foster collaborations for advancing ML applications in real-world agricultural settings.

Contributed talks / **92**

FPGA-Based Quantized Inception-Style Neural Networks for Enhancing Spatial Resolution in Electron Microscopy

Authors: Anna Bergamaschi[1]; Elisabeth Müller[1]; Emiliya Poghosyan[1]; Erik Fröjdh[1]; Kirsty A. Paton[1]; Lorenzo Asfour[2]; Luis Barba Flores[3]; Marius Köppel[2]; Thea Aarrestad[2]; Xiangyu Xie[1]

[1] *Paul Scherrer Institut*

[2] *ETH Zurich (CH)*

[3] *Swiss Data Science Center (SDSC Hub at PSI)*

High-resolution electron microscopy generates large volumes of pixel detector data due to beam rates reaching 10^7 to 10^{10} electrons per second directed at the sample. Of this data, only the electron entry point into the silicon detector prior to scattering is typically of interest for downstream analysis. Precise knowledge of these entry points is particularly important in electron microscopy with hybrid pixel detectors, which require sub-pixel resolution to compensate for their relatively large pixel size. While deep learning models have proven effective at reconstructing these entry points with sub-pixel accuracy, offline analysis requires storing full-frame raw data—introducing latency, bandwidth, and storage bottlenecks. Real-time, on-device inference addresses this by reducing each event to just two floating-point coordinates, enabling efficient data compression and immediate experimental feedback.

We present a quantized deep neural network designed for deployment on field-programmable gate arrays (FPGAs) for low-latency inference of electron entry points from 15×15 pixel energy maps. The model uses a compact, three-branch Inception-style architecture. Training is performed with quantization-aware techniques using QKeras, enabling deployment via the hls4ml toolchain and leveraging the high parallelism and low power consumption of FPGAs. Separate models trained on 80 keV and 200 keV single-electron events achieve root mean squared errors (RMSEs) of 0.289 and 0.621 pixels, respectively, matching full-precision baselines.

We extend the model with a lightweight reliability estimator trained to predict the accuracy of each position estimate via batchwise rank correlation. Discarding just the least confident 5% of events improves RMSE by over 11%, enabling tunable trade-offs between resolution and throughput in practical applications.

We report synthesis results demonstrating the model's feasibility for deployment in high-throughput experimental pipelines. This work demonstrates how real-time deep learning can be integrated directly into microscopy acquisition systems, eliminating the need for raw data storage while preserving spatial precision and enabling live feedback during imaging.

Contributed talks / **93**

Accelerated Graph Neural Network Inference on FPGAs for Real-Time Muon Triggering at the HL-LHC

Authors: Davide Fiacco[1]; Giuliano Gustavino[1]; Graziella Russo[1]; Martino Errico[1]; Stefano Giagu[2]; Valerio Ippolito[3]

[1] *Sapienza Universita e INFN, Roma I (IT)*

[2] *Sapienza Università di Roma and INFN Roma (IT)*

[3] *INFN Sezione di Roma (IT)*

The High Luminosity upgrade of the Large Hadron Collider (HL-LHC) presents a demanding environment for real-time data processing, with substantially increased event rates requiring faster and more efficient trigger systems. This study explores the deployment of graph neural networks (GNNs) on field-programmable gate arrays (FPGAs) for fast and accurate inference within future muon trigger pipelines. By leveraging the sparse and relational structure of detector data, GNNs enable robust pattern recognition while preserving spatial and topological correlations. We investigate hardware-friendly implementations of GNN architectures, focusing on model compression, parallelism, and low-latency execution; contributing to the broader goal of AI-driven event selection in high-energy physics experiments.

Posters and coffee / **94**

Loss Landscape Analysis for Reliable Quantized ML Models for Scientific Sensing

Author: Tommaso Baldi[1]

Co-authors: Alessandro Biondi [1]; Caleb Geniesse [2]; Javier Campos [3]; Olivia Weng [4]; Ryan Kastner [4]; Tran Nhan [3]

[1] *Scuola Superiore Sant'Anna*

[2] *Lawrence Berkeley National Laboratory, Berkeley, CA, USA*

[3] *Fermi National Accelerator Laboratory, Batavia, IL, USA*

[4] *University of California San Diego, San Diego, CA, USA*

In this paper, we propose a method to perform empirical analysis of the loss landscape of machine learning (ML) models. The method is applied to two ML models for scientific sensing, which necessitates quantization to be deployed and are subject to noise and perturbations due to experimental conditions.
Our method allows assessing the robustness of ML models to such effects as a function of quantization precision and under different regularization techniques—two crucial concerns that remained underexplored so far.
By investigating the interplay between performance, efficiency, and robustness by means of loss landscape analysis, we both established a strong correlation between gently-shaped landscapes and robustness to input and weight perturbations and observed other intriguing and non-obvious phenomena. Our method allows a systematic exploration of such trade-offs *a priori*, i.e., without training and testing multiple models, leading to more efficient development workflows. This work also highlights the importance of incorporating robustness into the Pareto optimization of ML models, enabling more reliable and adaptive scientific sensing systems.

Posters and coffee / **95**

Porting MADGRAPH to FPGA Using High-Level Synthesis (HLS)

Author: Hector Gutierrez Arance[1]

Co-authors: Alberto Valero Biot [1]; Arantza De Oyanguren Campos [1]; Francisco Hervas Alvarez [1]; Luca Fiorini [1]; Santiago Folgueras [2]

[1] *Univ. of Valencia and CSIC (ES)*

[2] *Universidad de Oviedo (ES)*

The escalating demand for data processing in particle physics research has spurred the exploration of novel technologies to enhance the efficiency and speed of calculations. This study presents the development of an implementation of MADGRAPH, a widely used tool in particle collision simulations, to FPGA using High-Level Synthesis. This research presents a proof of concept limited to a single, relatively simple process (e+ e- > mu+ mu-). The experimental evaluation methodology is described, focusing on performance comparison between traditional CPU implementations, GPU acceleration, and the new FPGA approach. This study describes the complex process of adapting MADGRAPH to FPGA using HLS, focusing on optimizing algorithms for parallel processing. These advancements could enable faster execution of complex simulations, highlighting FPGA's crucial role in advancing particle physics research. The encouraging results obtained in this proof of concept prove potential interest in testing the performance of the FPGA implementation of more complex processes.

Contributed talks / **96**

Quantum-Inspired Tensor Network Models for Ultrafast Jet Tagging on FPGAs

Authors: Alberto Coppi[None]; Ema Puljak[1]; Lorenzo Borella[2]

Co-authors: Andrea Triossi [2]; Daniel Jaschke ; Enrique Rico Ortega [3]; Jacopo Pazzini [4]; Maurizio Pierini [3]; Simone Montangero [5]

[1] *universitat Autònoma de Barcelona*

[2] *Universita e INFN, Padova (IT)*

[3] *CERN*

[4] *Università e INFN, Padova (IT)*

[5] *Padova University*

We conduct a systematic study of quantum-inspired Tensor Network (TN) models—Matrix Product States (MPS) and Tree Tensor Networks (TTN)—for real-time jet tagging in high-energy physics, with a focus on low-latency deployment on FPGAs. Motivated by the strict computational demands of the HL-LHC Level-1 Trigger system, we explore TN architectures as compact and interpretable alternatives to deep neural networks. Our models are trained on jet events represented by low-level features of jet constituents. Benchmarked against state-of-the-art deep learning classifiers, they demonstrated competitive performance in terms of classification accuracy and AUC. We implement quantization-aware training for TTNs and successfully deploy the best-performing models on FPGA hardware, evaluating DSP usage, latency and memory usage. We are currently working on extending the support for the quantization of MPS models and synthesizing their designs for full FPGA deployment, to be able to compare them with TTNs in terms of both performance and hardware cost. This work aims to highlight the potential of TN-based models for fast, resource-efficient inference in low-latency environments such as the LHC.

Contributed talks / 99

PrioriFI: Efficient Fault Injection for Edge Neural Networks

Author: Olivia Weng[None]

Co-authors: Andres Meza [1]; Nhan Tran [2]; Ryan Kastner [3]

[1] *UCSD*

[2] *Fermi National Accelerator Lab. (US)*

[3] *University of California San Diego, San Diego, CA, USA*

As neural networks (NNs) are increasingly used to provide
edge intelligence, there is a growing need to make the edge devices
that run them robust to faults. Edge devices must mitigate the resulting
hardware failures while maintaining strict constraints on power, energy,
latency, throughput, memory size, and computational resources. Edge
NNs require fundamental changes in model architecture, e.g., quantization and fewer, smaller layers. PrioriFI is an efficient fault injection (FI) algorithm that evaluates edge NN robustness by ranking NN bits based on their fault sensitivity. PrioriFI uses the Hessian for the initial parameter ranking. Then, during an FI campaign, PrioriFI uses the information gained from each FI to focus on the bits likely to be the next most sensitive. With PrioriFI, designers can quickly evaluate different NN architectures and co-design fault-tolerant edge NNs.

Birds-of-a-Feather / 101

QONNX Birds-of-a-Feather (BoF) Session

Author: Yaman Umuroglu[None]

QONNX (Quantized ONNX) serves as a shared input representation and frontend for several efficient inference projects, including FINN, chisel4ml and NN2FPGA. This birds-of-a-feather session would

serve as a gathering point for the community to discuss recent developments and future plans for QONNX.

Contributed talks / 102

Low-Latency Resource-Efficient GNNs for Jet Tagging on FPGAs

Authors: Zhiqiang (Walkie) Que[1]; Chang Sun[2]

Co-authors: Lauri Antti Olavi Laatu [3]; Arianna Cox [3]; Alex Tapper [1]; Wayne Luk

[1] *Imperial College London*

[2] *California Institute of Technology (US)*

[3] *Imperial College (GB)*

Graph Neural Networks (GNNs), particularly Interaction Networks (INs), have shown exceptional performance for jet tagging at the CERN High-Luminosity Large Hadron Collider. However, their computational complexity and irregular memory access patterns pose significant challenges for deployment on FPGAs in hardware trigger systems, where strict latency and resource constraints apply.

In this work, we present the first co-optimized framework that integrates High Granularity Quantization (HGQ) and Distributed Arithmetic for Machine Learning (da4ml) to enable efficient inference of IN-based GNNs on FPGAs. HGQ performs fine-grained, layer- and channel-level bitwidth allocation through hardware-aware optimization, minimizing precision where possible without sacrificing classification performance. Complementing this, da4ml replaces traditional multiply-accumulate units with highly parallel, LUT-based arithmetic units, allowing low-latency and resource-efficient implementation of quantized GNNs.

We demonstrate our approach on a public benchmark dataset for jet classification, implementing compressed and quantized IN models on FPGAs. Our results show that GNN designs with the combined HGQ+da4ml approach achieve significant reductions in DSP and LUT usage compared to state-of-the-art GNN designs, while maintaining model accuracy and satisfying strict latency constraints.

Contributed talks / 103

FINN+: Towards Hassle-Free Co-Design of FPGA DNN Inference Accelerators

Author: Felix Jentzsch[None]

Co-authors: Bjarne Wintermann [1]; Christoph Berganski [1]; Linus Jungemann [1]

[1] *Paderborn University*

Custom FPGA dataflow accelerators for DNN inference can enable unprecedented performance and efficiency for many applications. Dataflow accelerator compilers, such as the FINN framework, have improved in recent years and allow practitioners to explore this technology without requiring in-depth FPGA knowledge.

However, the overall design process remains quite tedious, time-consuming, and often requires significant manual intervention. This is primarily caused by limited flexibility and automation in the compiler, as well as the enormous size and complexity of the design space. In contrast to the typical

exploration process, where a quantized DNN is manually trained and then passed through the compiler, requiring many iterations to reach an acceptable solution, we envision an automated co-design of DNN and FPGA accelerator based on Automated Machine Learning (AutoML) techniques.

In an effort to realize this vision while also facilitating the exploration of FPGA dataflow accelerators for energy efficient inference in the datacenter, we introduce FINN+, our custom fork of the FINN framework. Our work so far includes empirical resource and power consumption modeling, support for Transformer topologies, efficient deployment on datacenter FPGAs, Multi-FPGA acceleration, and general usability improvements.
In this talk, we will share recent highlights as well as remaining challenges of the project.

Posters and coffee / 104

Dynamic Scheduling Support for Faster ML Inference in hls4ml

Authors: Andrei Girjoaba[1]; Andela Kostić[1]; Jiahui Xu[1]; Lana Josipović[1]

[1] *ETH Zurich*

FPGAs are performant and flexible microchips well-suited for experimental physics that efficiently run anomaly detection algorithms and identify potential new physical phenomena. However, FPGAs are not easy to program: A significant gap exists between the algorithms used to discover new physics and the low-level hardware description languages (HDLs) required to program FPGAs. To tackle the absence of hardware expertise, collaboration between physicists and hardware designers led to the creation of the hls4ml project 1: well-calibrated machine learning (ML) models—the core components in the algorithm used in experimental physics—are automatically translated into C code and embedded in the whole algorithm, which high-level synthesis (HLS) tools can then use to generate HDL.

The efficiency of the HDL code critically depends on how well the HLS tools can implement the underlying algorithm: hls4ml relies on HLS tools that use static scheduling, i.e., a predefined and fixed schedule dictates computations. These HLS tools are well-suited for problems with definite sizes and dimensions. Despite that, the computation involved in ML inference is usually highly regular; however, real-world events are often sparse and dynamic: certain operators may be conditionally bypassed, batch sizes can vary unpredictably, and data is frequently represented in a sparse format. Unfortunately, in the presence of variability, pipelining—the most prominent HLS optimization technique for fast and efficient hardware—is not achievable in static scheduling.

Dynamatic is an open-source HLS compiler that generates dynamically scheduled dataflow circuits 2. Dataflow circuits deliver the best performance in the presence of variability: the circuit does not need to follow a predefined computation schedule that is designed for the worst-case scenario; computations progress as soon as the data is available. Dynamatic has been successfully used to demonstrate a performance advantage over traditional, statically scheduled HLS tools on conditionally skipped operations, unpredictable workloads, or sparse linear algebra.

This poster presents our ongoing effort in adding the Dynamatic HLS compiler as a backend of hls4ml. The purpose is to enable scientists to develop more dynamic and irregular algorithms and, ultimately, broaden the applicability of hls4ml. Our poster will discuss the following elements:
- We give a general overview of Dynamatic, as well as its strengths and weaknesses.
- We show that Dynamatic-produced implementations of common machine learning operators are on par with those produced by other HLS tools.
- We discuss our adaptations to hls4ml to emit HLS C code that is compatible with Dynamatic.
- We explore the benefits of dynamic scheduling in an anomaly detection algorithm, and compare the implementation quality with those achieved by existing hls4ml backends.

We hope that our poster will be of interest to users and developers of hls4ml, and we look forward to valuable interactions and feedback from the Fast ML community.

References
- 1 Duarte et al., JINST ’18
- 2 Josipovic et al., FPGA ’18

Posters and coffee / **106**

TDSCAN : Trigger Distributed Spatial Convolution Area Network

Authors: Andres Upegui Posada[1]; Quentin Berthet[1]; Tanguy Dietrich[None]

[1] *HEPIA - Haute école du paysage, d’ingénierie et d’architecture (CH)*

Cherenkov Telescope cameras stream about 1 Billion frames per seconds and are dominated by night-sky background, yet the γ-ray air-shower patterns of interest appear only occasionally.
Filtering is thus paramount for guaranteeing that science-grade data are recorded without saturating the downstream read-out.
In this work we present TDSCAN (Trigger Distributed Spatial Convolution Area Network) meets this challenge by replacing full cluster reconstruction with a fixed 3-D neighbourhood vote implemented as a small convolution kernel on the camera’s native hexagonal grid.
The algorithm is deployed on three Kintex UltraScale FPGAs each analyse a third of the camera, and a fourth device merges their votes, collectively handling the 1 Billion frame per second stream in real time while keeping trigger latency safely below one microsecond regardless of event complexity.

Contributed talks / **107**

ENABOL: Enabling Neural Backpropagation On-chip Learning for Edge AI Systems

Authors: Manuel Valentin[1]; Houxuan Guo[1]; Ryan Forelli[1]; Ethan Gindlesperger[1]; Seda Ogrenci[1]

[1] *Northwestern University*

On-chip learning has the potential to unlock low-latency, low-power, and continuously adaptive AI directly on edge devices. However, research in this area remains limited by the lack of accessible hardware toolchains that support backpropagation. To address this gap, we propose *ENABOL*, a hardware-efficient extension of the *HLS4ML* toolchain that enables customizable backpropagation support for widely implemented neural network layers. *ENABOL* allows users to translate high-level Python models with training logic into synthesizable HLS-compatible C++ implementations, allowing seamless integration into FPGA/ASIC flows. It lowers the barrier for prototyping and exploring on-chip training strategies by automating key components of the training pipeline, such as gradient computation, weight updates, and control flow. While not intended as a fully optimized production framework, *ENABOL* provides a foundational platform for experimentation, hardware–algorithm co-design, and future optimization. Its compatibility with edge-oriented backend implementation platforms such as ESP facilitates scalable, real-time learning research in resource-constrained environments.

Contributed talks / **108**

SuperSONIC: Cloud-Native Infrastructure for ML Inferencing

Authors: Dmitry Kondratyev[1]; Benedikt Riedel[None]; Yuan-Tang Chou[2]; Miles Cochran-Branson[2]; Noah Paladino[3]; David Schultz[4]; Miaoyuan Liu[1]; Javier Mauricio Duarte[5]; Philip Coleman Harris[3]; Shih-Chieh Hsu[6]

[1] *Purdue University (US)*

[2] *University of Washington (US)*

[3] *Massachusetts Inst. of Technology (US)*

[4] *University of Wisconsin-Madison*

[5] *Univ. of California San Diego (US)*

[6] *University of Washington Seattle (US)*

The rising computational demands of increasing data rates and complex machine learning (ML) algorithms in large-scale scientific experiments have driven the adoption of the Services for Optimized Network Inference on Coprocessors (SONIC) framework. SONIC accelerates ML inference by offloading tasks to local or remote coprocessors, optimizing resource utilization. Its portability across diverse hardware platforms improves data processing and model deployment efficiency in advanced research domains such as high-energy physics (HEP) and multi-messenger astrophysics (MMA). We developed SuperSONIC, a scalable server infrastructure for SONIC that enables the deployment of computationally intensive inference tasks, such as charged particle reconstruction, on Kubernetes clusters equipped with graphics processing units (GPUs). Leveraging NVIDIA's Triton Inference Server, SuperSONIC decouples client workflows from server infrastructure, standardizing communication, improving throughput, and enabling robust load balancing and monitoring. SuperSonic has been successfully deployed in production environments, including the CMS and ATLAS experiments at CERN's Large Hadron Collider, the IceCube Neutrino Observatory, and the LIGO gravitational-wave observatory. It offers a reusable, configurable framework that addresses cloud-native challenges and enhances the efficiency of accelerator-based inference across diverse scientific and industrial applications.

Posters and coffee / **109**

Acceleration of a Quantized LeNet-based IDS on FPGA using FINN

Author: Ameth Thiam[None]

Co-authors: Cathrine Dezan ; Kevin DRUART

1. Introduction and Context

With the rise of cyberattacks and the growing volume of network traffic, intrusion detection systems (IDS) must provide fast, accurate, and resource-efficient analysis. Traditional CPU- or GPU-based solutions often struggle to meet low-latency and low-power requirements, especially in embedded environments.

Integrating artificial intelligence, particularly convolutional neural networks (CNNs), offers new perspectives for improving IDS. However, deploying such models on constrained hardware architectures remains challenging.

In this context, the FINN framework from Xilinx enables efficient execution of quantized neural networks on FPGA. This work evaluates the feasibility and benefits of implementing a quantized LeNet-based CNN IDS, focusing on latency and resource consumption after hardware synthesis.

1. FINN Framework Overview

FINN is an open-source framework developed by Xilinx, designed for hardware acceleration of quantized neural networks on FPGAs. Unlike traditional CPU/GPU inference frameworks, FINN targets ultra-low latency and high throughput, making it suitable for real-time embedded applications.

The process begins with model training in PyTorch, using the Brevitas library to integrate quantization during learning. The model is then exported to the QONNX format, allowing FINN to optimize it for hardware execution.

FINN applies a series of graph transformations to prepare the model for FPGA synthesis, including computation streamlining, conversion of layers into HLS blocks, and dataflow partitioning. The flow concludes with FPGA bitstream generation and the creation of hardware drivers.

1. Intrusion Detection System and Preprocessing

We used the DAPT dataset, containing 86,691 network traffic samples labeled across 14 activity classes. Each sample is represented by a 32-dimensional numerical feature vector. The dataset was split into 60,683 training samples and 26,008 test samples, preserving class distribution.

The classification task is multiclass, targeting activities such as web browsing, messaging, and various network attacks. We opted for a 1D-adapted, quantized LeNet CNN suitable for vector-based data.

Preprocessing included feature normalization and label encoding. Training was performed using Brevitas with quantized layers (4-bit weights and activations) to ensure compatibility with FINN. The final model was exported in QONNX format for hardware deployment.

1. Results and Evaluation

Hardware generation was carried out by configuring the FINN pipeline with a 200 MHz target frequency, a 100 FPS objective, and a ZCU102 FPGA board. After QONNX export and compilation, the estimated pre-synthesis latency was 137.16 µs with a throughput of 19,841 images per second.

After Vivado synthesis, the RTL simulation showed a latency of 2.18 ms and a throughput of 572.6 images per second, with a resource usage of 11,474 LUTs and 33 DSPs. Real-world execution achieved an effective throughput of 9,882 images per second.

In terms of classification performance, the model achieved 80% accuracy before quantization and 78% after Quantization-Aware Training (QAT). Final post-synthesis accuracy on the deployed FPGA is currently under evaluation.

Posters and coffee / **110**

Locality-Sensitive Hashing-Based Efficient Point Transformer for Charged Particles Reconstruction

Authors: Advaith Anand[1]; Amit Saha[2]; Jack Patrick Rodgers[3]; Miaoyuan Liu[3]; Pan Li[None]; Shih-Chieh Hsu[4]; Shitij Govil[2]; Siqi Miao[5]; Yuan-Tang Chou[1]

[1] *University of Washington (US)*

[2] *Georgia Institute of Technology*

[3] *Purdue University (US)*

[4] *University of Washington Seattle (US)*

[5] *Purdue University*

Charge particle track reconstruction is the foundation of the collider experiments. Yet, it's also the most computationally expensive part of the particle reconstruction. The innovation in tracking reconstruction using graph neural networks (GNNs) has demonstrated a promising capability to address the computing challenges posed by the High-Luminosity LHC (HL-LHC) with Machine learning. However, GNNs face limitations involving irregular computations and random memory access, slowing down their speed. In this talk, we introduce a Locality-Sensitive Hashing-Based Efficient Point Transformer (HEPT) with advanced attention methods, offering a superior alternative with near-linear complexity, achieving milliseconds of latency and memory consumption. We present a comprehensive evaluation of HEPT's computational efficiency and physics performance

compared to other algorithms, such as GNN-based pipelines, highlighting its potential to accelerate full track reconstruction with novel object condensation and track building approaches.

Contributed talks / 113

Arbolta: A Fault Tolerance Study using Minimal Hardware Simulation

Authors: Alexander Redding[1]; Ian Colbert[2]; Olivia Weng[3]; Ryan Kastner[3]

[1] *UC San Diego*

[2] *AMD*

[3] *University of California San Diego, San Diego, CA, USA*

The widespread deployment of embedded ML systems has created a need for resilient, fault-tolerant hardware and software capable of operating in inherently noisy conditions. While the standardization of low-precision (≤ 8-bit) datatypes has allowed for reduced training and inference costs and increased interoperability across commercial accelerators, clear guidelines for robust implementation under faulty conditions remains underdeveloped. Prior work has improved the efficiency of accelerator resilience studies through targeted fault injection campaigns at the software level (e.g., weights and inputs) and hybrid approaches which model high-level architectural state. However, these methods rely on assumptions which do not hold across the diverse hardware implementations of arithmetic units for emerging datatypes. This work extends the open-source Arbolta framework to enable fault injection in simulated accelerators through lightweight hardware-level simulation, capturing critical microarchitectural effects within an accessible Python environment. We compare Arbolta to existing tools which focus solely on fault injection in model weights/inputs and high-level architectural registers and propose a novel workflow for the design space exploration of resilient accelerators and models. We demonstrate the value of lightweight hardware simulation by presenting a series of case studies culminating in a fault-injection campaign on a minimal accelerator. Finally, we discuss the insights of our case studies and their applicability to more realistic hardware designs.

Posters and coffee / 114

Real-Time Andean Cryo-Hydrology Intelligence: A Heterogeneous Computing Framework for Glacial Hazard Early Warning

Author: Siwar Jose Basualdo Garcia[None]

The accelerated retreat of tropical glaciers in the Peruvian Andes poses an imminent and catastrophic threat of Glacial Lake Outburst Floods (GLOF). These events can devastate downstream communities like Huaraz with warning times of less than 15 minutes 1. Existing monitoring systems are inadequate for this challenge; optical satellite observations (e.g. Landsat, Sentinel-2) are frequently obscured by persistent cloud cover, while in-situ sensor networks provide only sparse point-based data, failing to capture the holistic dynamics of glacial instability [2, 3]. This creates critical gaps in situational awareness, precluding effective real-time early warning.

To address this, we propose a novel, end-to-end framework for GLOF early warning that leverages heterogeneous computing and multi-modal data fusion. Our system is designed to integrate continuous all-weather Synthetic Aperture Radar (SAR) data from Sentinel-1 with intermittent optical/thermal imagery (Sentinel-2, ASTER) and high-frequency data from ground-based GPS and meteorological sensor networks [1, 4]. The analytical core of the system is a spatio-temporal Graph Neural Network (GNN). This architecture is uniquely suited to model the complex cryo-hydrological system—encompassing the glacier, moraine dam, lake, and downstream valley—as a dynamic graph.

The GNN is trained on historical data to learn the complex, nonlinear correlations between disparate data modalities and identify subtle, system-wide patterns that are precursors to instability 5.

The central "Fast Machine Learning"innovation is the deployment of this model within a two-stage, ultra-low-latency pipeline that adapts the trigger system paradigm from high-energy physics to environmental science [6, 7]. The first stage consists of a lightweight, compressed version of the GNN, created via quantization and pruning, implemented on a power-efficient edge Field-Programmable Gate Array (FPGA) [6, 8]. This FPGA performs continuous inference directly on the raw data stream, acting as an intelligent trigger. Upon detecting a potential precursor event, it initiates the second stage: the relevant high-resolution data subset is transmitted to a cloud backend where a full-precision GNN, running on high-performance GPUs, confirms and characterizes the anomaly. The confirmed event parameters are then used to automatically initialize a GPU-accelerated, physically-based hydrological model (e.g., HEC-RAS), simulating the GLOF's inundation path and providing actionable, minutes-matter warnings to disaster response agencies [9].

This edge-to-cloud architecture represents a paradigm shift from retrospective analysis to proactive, predictive hazard assessment. By successfully transferring and adapting a core "Fast Machine Learning"concept from experimental physics to a critical geoscience application, our system provides a scalable, robust, and life-saving blueprint for mitigating GLOF risks in the Peruvian Andes and other vulnerable mountain regions worldwide. The economic impact of such a real-time system is also fundamental for regional development, as it provides the necessary hazard mitigation to secure high-value infrastructure projects, such as the proposed railway connecting the new port of Chancay to Brazil [10]. Furthermore, this approach is designed to serve as a precedent for future applications tackling other complex emergent phenomena in similarly challenging geographical environments.

Birds-of-a-Feather / 116

MLOps Pipeline for Continuous Deployment of Machine Learning Algorithms for HEP

Authors: Christopher Edward Brown[1]; Ioannis Xiotidis[1]; Maciej Mikolaj Glowacki[1]; Marius Köppel[2]

[1] *CERN*

[2] *ETH Zurich (CH)*

We present an MLOps-based approach for managing the end-to-end lifecycle of machine learning (ML) algorithms deployed on FPGAs in real-time trigger systems, as used in experiments such as CMS and ATLAS. The primary objective of this pipeline is to enable agile and robust responses to evolving detector and beam conditions by automating the collection of new training data, retraining and optimizing models, validating performance, synthesizing firmware, and deploying updated versions to both online and offline environments.
To monitor model stability over time, we incorporate dedicated data streams that bypass trigger selections (e.g., scouting or express streams). These streams allow for continuous monitoring of model outputs and the detection of distributional drifts, enabling us to assess model operational lifetimes and support strategies like continual learning, periodic retraining, or threshold adjustment to ensure consistent performance.
Our pipeline uses existing computing infrastructure, which includes distributed computing resources, container orchestration frameworks like Kubeflow, and CI/CD tools such as GitLab, to provide a scalable and maintainable foundation for real-time ML integration. This architecture supports rapid iteration cycles while promoting long-term sustainability, both of which are essential as ML becomes more central to trigger design and real-time data processing in modern collider experiments.
We invite discussion on shared challenges, solutions, and future directions for managing the ML lifecycle in low-latency HEP environments. This includes topics like model validation, deployment strategies, firmware synthesis workflows, and the role of community-developed tooling across experiments.

Contributed talks / 117

Smartpixels: Intelligent pixel detectors: Towards a radiation hard ASIC with on-chip machine learning in 28nm CMOS

Author: Jannicke Pearkes[1]

Co-authors: Abhijith Gandrakota [2]; Alice Bean [3]; Anthony Badea [4]; Benjamin Parpillon [2]; Benjamin Weiss [5]; Corrinne Elaine Mills [6]; Douglas Ryan Berry [2]; Farah Fahim [7]; Giuseppe Di Guglielmo [7]; Jennet Elizabeth Dickinson [8]; Jim Hirschauer [2]; Karri Folan Di Petrillo [9]; Keith Ulmer [10]; Lindsey Gray [2]; Mark Neubauer [11]; Miaoyuan Liu [12]; Morris Swartz [13]; Nhan Tran [2]; Nick Manganelli [14]; Petar Maksimovic [15]; Ronald Lipton [2]; Shruti R Kulkarni [16]

[1] *University of Colorado Boulder (US)*
[2] *Fermi National Accelerator Lab. (US)*
[3] *The University of Kansas (US)*
[4] *University of Chicago (US)*
[5] *Cornell University*
[6] *University of Illinois Chicago (US)*
[7] *Fermilab*
[8] *Cornell University (US)*
[9] *University of Chicago*
[10] *University of Colorado, Boulder (US)*
[11] *Univ. Illinois at Urbana Champaign (US)*
[12] *Purdue University (US)*
[13] *Johns Hopkins University (JHU)*
[14] *Northeastern University (US)*
[15] *Johns Hopkins University (US)*
[16] *Oak Ridge National Laboratory*

The Smartpixels project is a coordinated effort to co-design pixel ASICs, design tools, ML algorithms, and sensors for on-detector data reduction, motivated by the technical challenges of current and future colliders. The drive to greater precision requires smaller pixel pitch, which together with higher event rates arising from pileup and/or beam-induced background generates petabytes of data per second. Readout chips must be power-efficient, radiation-hard, and capable of real-time data processing.

The smartpixels team has developed algorithms for selecting the signatures of high-momentum tracks and coarse particle-trajectory reconstruction, and explored how the performance changes with pixel sensor geometry, orientation, and irradiation.

We have leveraged and extended hls4ml to support neural network architectures meeting the strict latency and area constraints. To target our TSMC 28nm ASIC implementations, we have integrated the flow with Catapult HLS, allowing seamless synthesis of these designs into RTL for backend integration, and our first custom pixel ASICs have been produced and are undergoing testing.

We will present the status of ongoing work, including efforts in testing ASICs, producing a new ASIC with trajectory-reconstruction algorithm, and improving the realism of the detector simulation though including noise, charge thresholds, and other effects.

Contributed talks / 121

Low-latency Jet Tagging for HL-LHC Using Transformer Architectures

Authors: Abhijith Gandrakota[1]; Alex Tapper[2]; Arianna Cox[3]; Benedikt Maier[3]; Chang Sun[4]; Jennifer Ngadiuba[5]; Lauri Antti Olavi Laatu[3]; Zhiqiang (Walkie) Que[2]

[1] *Fermi National Accelerator Lab. (US)*

[2] *Imperial College London*

[3] *Imperial College (GB)*

[4] *California Institute of Technology (US)*

[5] *FNAL*

Transformers are the state-of-the-art model architectures and widely used in application areas of machine learning. However the performance of such architectures is less well explored in the ultra-low latency domains where deployment on FPGAs or ASICs is required. Such domains include the trigger and data acquisition systems of the LHC experiments.

We present a transformer-based algorithm for jet tagging built with the HGQ2 framework, which is able to produce a model with heterogeneous bitwidths for fast inference on FPGAs, as required in the trigger systems at the LHC experiments. The bitwidths are acquired during training by minimizing the total bit operations as an additional parameter. By allowing a bitwidth of zero, the model is pruned in-situ during training. Using this quantization-aware approach, our algorithm achieves state-of-the-art performance while also retaining permutation invariance which is a key property for particle physics applications

Due to the strength of transformers in representation learning, our work serves also as a stepping stone for the development of a larger foundation model for trigger applications.

Posters and coffee / **122**

NomAD: Low-Latency Unsupervised Anomaly Detection for the ATLAS Trigger

Authors: ATLAS Collaboration[None]; ATLAS Collaboration[None]; Rajat Gupta[1]

[1] *University of Pittsburgh (US)*

We present NomAD (Nanosecond Anomaly Detection), a real-time anomaly detection algorithm designed for the ATLAS Level-1 Topological (L1Topo) trigger using unsupervised machine learning. The algorithm combines a Variational Autoencoder (VAE) with Boosted Decision Tree (BDT) regression to compress and distill deep learning inference into a firmware-compatible format for FPGAs. Trained on 2024 ATLAS data using low-level L1 muon features, NomAD identifies anomalous dimuon events beyond standard trigger logic. The resulting anomaly score enables mass-agnostic, model-independent selection of low-momentum dimuons with clear signal-background separation. Firmware tests confirm sub-25 ns latency and successful synthesis using standard FPGA design tools. This talk highlights the full ML-to-firmware pipeline, performance on B-physics signals, and its implications for fast, inclusive triggers in high-throughput environments.

Tutorials / **123**

Part 1: Agile Hardware Design for AI: A Hands-On Tutorial with SODA and Bambu

Authors: Nicolas Bohm Agostini[1]; Giovanni Gozzi[2]; Michele Fiorito[2]; Serena Curzel[2]; Vito Giovanni Castellana[3]; Antonino Tumeo[3]; Fabrizio Ferrandi[2]

[1] *Northeastern University and Pacific Northwest National Laboratory*

[2] *Politecnico di Milano*

[3] *Pacific Northwest National Laboratory*

This tutorial explores the growing demand for domain-specific hardware accelerators driven by the rapid evolution of AI and data analytics. Traditional hardware design cycles are too slow to keep up with the pace of algorithmic innovation. To address this, new agile hardware design methodologies are emerging, leveraging compiler technologies and High-Level Synthesis (HLS) to automate and accelerate the design process.

The tutorial introduces the SODA Synthesizer, an open-source, compiler-based toolchain that enables the generation of efficient hardware accelerators from high-level algorithm descriptions.

It consists of:

- SODA-OPT: A front-end and optimizer built on the MLIR (Multi-Level Intermediate Representation) framework. It interfaces with popular Python-based data science and machine learning frameworks, performing hardware/software partitioning and domain-specific optimizations.
- Bambu: A state-of-the-art open-source HLS tool developed at Politecnico di Milano. It translates optimized high-level code into hardware designs, supporting both FPGA and ASIC targets, and integrates with RTL simulation and logic synthesis tools.

The tutorial highlights the limitations of traditional HLS tools, which were primarily designed for digital signal processing and required expertise in hardware description languages (HDLs). Modern HLS tools now support parallel programming models and integrate with high-level frameworks, making them more accessible to software developers and domain scientists.

Key topics include:

1. Current trends and methodologies in agile hardware design.
2. Advantages and limitations of conventional HLS approaches.
3. The role of MLIR in supporting diverse domains and frameworks.
4. Hands-on demonstrations of both SODA-OPT and Bambu, showing how to go from Python code to optimized hardware accelerators.

The tutorial also demonstrates how the toolchain integrates with both commercial and open-source Electronic Design Automation (EDA) tools, such as OpenROAD, enabling a complete end-to-end flow from Python to silicon. The design flow is flexible and supports both FPGA prototyping and ASIC implementation, making it suitable for a wide range of applications in AI and data-intensive computing.

This tutorial is intended for researchers, engineers, and developers interested in accelerating AI and data-driven applications through agile hardware design, even without deep expertise in traditional hardware design languages.

References

 [1] G. Gozzi, M. Fiorito, S. Curzel, C. Barone, V. G. Castellana, M. Minutoli, A. Tumeo, and F. Ferrandi: SPARTA: High-Level Synthesis of Parallel Multi-Threaded Accelerators. ACM Trans. Reconfigurable Technol. Syst. 18, 1, Article 9 (March 2025).

 [2] N. Bohm Agostini, S. Curzel, J. Zhang, A. Limaye, C. Tan, V. Amatya, M. Minutoli, V. G. Castellana, J. B. Manzano, D. Brooks, G-Y. Wei, A. Tumeo: Bridging Python to Silicon: The SODA Toolchain. IEEE Micro 42(5): 78-88 (2022) (Best paper for 2022)

 [3] N. Bohm Agostini, S. Curzel, V. Amatya, C. Tan, M. Minutoli, V. G. Castellana, J. B. Manzano, D. R. Kaeli, A. Tumeo: An MLIR-based Compiler Flow for System-Level Design and Hardware Acceleration. ICCAD 2022: 6:1-6:9

 [4] F. Ferrandi, V. G. Castellana, S. Curzel, P. Fezzardi, M. Fiorito, M. Lattuada, M. Minutoli, C. Pilato, A. Tumeo: Invited: Bambu: an Open-Source Research Framework for the High-Level Synthesis of Complex Applications. DAC 2021: 1327-1330

Posters and coffee / 124

State Space Models for Scientific Time Series Applications

Authors: Christina Reissel[1]; Maira Khan[2]

Co-authors: Abhijith Gandrakota [3]; Ahmed Enis Cetin [4]; Emadeldeen Hamdan [5]; Jennifer Ngadiuba [6]; Liam Sheldon ; Mengke Zhang ; Nhan Tran [3]; Philip Coleman Harris [1]; Tao Wei [7]

[1] *Massachusetts Inst. of Technology (US)*

[2] *Fermi National Accelerator Laboratory*

[3] *Fermi National Accelerator Lab. (US)*

[4] *UIUC*

[5] *University of Illinois Chicago*

[6] *FNAL*

[7] *Clemson University*

We investigate the application of state space models (SSMs) to a diverse set of scientific time series tasks. In particular, we benchmark the performance of SSMs against a set of baseline neural networks across three domains: magnet quench prediction, gravitational wave signal classification (LIGO), and neural phase estimation. Our analysis evaluates both computational efficiency—quantified by the number of mathematical operations—and task-specific performance metrics. Results suggest that SSMs offer a favorable trade-off between accuracy and computational efficiency, making them a possible alternative to conventional deep learning models in scientific settings.

Posters and coffee / **125**

Fast Adaptive Neural Control of Resonant Extraction at Fermilab

Authors: Andrew Whitbeck[1]; Jingtian Ji[2]; Maira Khan[3]

Co-authors: Aakash Naryan [3]; Jason St. John [3]; Jose Berlioz [3]; Kit Danielson-Fieldhouse [3]; Kyle Hazelwood [3]; Matthew Walter [2]; Nhan Tran [1]

[1] *Fermi National Accelerator Lab. (US)*

[2] *Toyota Technical Institute of Technology*

[3] *Fermi National Accelerator Laboratory*

We present the development of a machine learning (ML) based regulation system for third-order resonant beam extraction in the Mu2e experiment at Fermilab. Classical and ML-based controllers have been optimized using semi-analytic simulations and evaluated in terms of regulation performance and training efficiency. We compare several controller architectures and discuss the integration of neural control into an adaptive framework. We also present progress on surrogate models that predict the controller response given a spill intensity and controller action history. To enable real-time deployment, we report progress on implementing low-latency, edge-based inference suitable for hardware-constrained environments. Our results demonstrate the feasibility and advantages of ML-based control in managing complex, time-varying physical systems, with broader implications for accelerator operations and other domains requiring fast, adaptive regulation.

Posters and coffee / **126**

Towards Online Machine Learning in DUNE Data Acquisition

Author: Olivia Dalager[1]

Co-authors: Amanda Weinstein ; Andrew Mogan ; Avik Ghosh [2]; Bonnie King [3]; Georgia Karagiorgi ; Jack Henry Cleeve [4]; Janina Dorin Hakenmueller [5]; Jennifer Ngadiuba [6]; Joshua Queen ; Jovan Mitrevski [3]; Kate Scholberg ; Maira Khan [7]; Matthew King ; Michael H L Wang ; Nhan Tran [3]

[1] *Fermilab*

[2] *Iowa State University*

[3] *Fermi National Accelerator Lab. (US)*

[4] *Columbia University*

[5] *Duke University*

[6] *FNAL*

[7] *Fermi National Accelerator Laboratory*

Processing the large volumes of data produced by liquid argon time projection chamber (LArTPC) experiments presents a significant challenge, especially those at the scale of the Deep Underground Neutrino Experiment (DUNE). This is a particular challenge when aiming to trigger on low-energy neutrinos from core-collapse supernovae, which are typically buried in a high-rate radiological background. To enable real-time event selection suitable for such rare signals, we are developing machine learning–based data filtering methods. In order to demonstrate the feasibility of this approach, we implemented such pipeline using the ICEBERG detector at Fermilab as a small-scale LArTPC, with a focus on identifying Michel electrons as a proxy for low-energy neutrino interactions. This talk will present the current status of integrating these machine learning models into the data acquisition (DAQ) system of this detector via the SONIC (Services for Optimized Network Inference on Coprocessors) framework, leveraging the NVIDIA Triton Inference Server. This demonstrator presents the first online approach to ML based data filtering within DUNE-DAQ, and generates a valuable dataset that can be used for evaluation of directional reconstruction algorithms for pointing.

Contributed talks / 128

Accelerating Efficient Transformer Architectures for Point Cloud Data using hls4ml (REMOTE)

Authors: Bo-Cheng Lai[None]; Chia-En Chang[1]; Divij Agarwal[2]; Jan-Frederik Schulte[3]; Miaoyuan Liu[3]; Rian Flynn[2]; Siqi Miao[4]; Vladimir Loncar[5]

[1] *National Yang Ming Chia Tung University*

[2] *Purdue University*

[3] *Purdue University (US)*

[4] *Georgia Institute of Technology*

[5] *CERN*

The analysis of point cloud data, for example signals from charged particles recorded by detectors in high energy physics (HEP) experiments, can be significantly enhanced and accelerated by the application of machine learning models. In recent years, transformer architectures have come into focus as offering excellent model performance. However, for traditional transformers,the need to compute attention between all elements of the input data set results in high computational requirements and poor scaling of the inference performance with increasing data set size. To address this, the Locality-Sensitive Hashing-Based Efficient Point Transformer (HEPT) has been proposed, which segments the input dataset into smaller samples based on their adjacency which is evaluated using a hashing function. This approach has been shown to greatly improve computational efficiency when deployed on traditional GPU architectures.
For deployment with stricter latency requirements, for example in the trigger systems of HEP experiments, further accelerating the inference of the HEPT architecture is required. We present an implementation of HEPT for AMD/Xilinx FPGAs using hls4ml, which includes the hashing and segmentation of the data set, the attention computation, and recombination of the data. Using a charged particle track reconstruction model as the benchmark, latencies on the microsecond scale are achieved within the computing resources available on an Alveo u250 FPGA. Model compression using pruning and quantization with PQuant is explored.

Contributed talks / **129**

da4ml: Distributed Arithmetic for Real-time Neural Networks on FPGAs

Author: Chang Sun[1]

Co-authors: Zhiqiang (Walkie) Que [2]; Vladimir Loncar [3]; Wayne Luk ; Maria Spiropulu [1]

[1] *California Institute of Technology (US)*

[2] *Imperial College London*

[3] *CERN*

Neural networks with a latency requirement on the order of microseconds, like the ones used at the CERN Large Hadron Collider, are typically deployed on FPGAs pipelined with II=1. A bottleneck for the deployment of such neural networks is area utilization, which is directly related to the required constant matrix-vector multiplication (CMVM) operations. In this work, we propose an efficient algorithm for implementing CMVM operations with distributed arithmetic (DA) on FPGAs that simultaneously optimizes for area consumption and latency. The algorithm achieves resource reduction similar to state-of-the-art algorithms while being significantly faster to compute.

We release da4ml, a free and open source package that enables end-to-end, bit-exact neural network to Verilog or HLS design conversion, optimized with the proposed algorithm. For easy adoption into existing workflows, we also integrate da4ml into the hls4ml library. The results show that da4ml can reduce on-chip resources by up to a third for realistic, highly quantized neural networks while simultaneously reducing latency compared to the native implementation hls4ml, enabling the implementation of previously infeasible networks.

Posters and coffee / **130**

wa-hls4ml: A Benchmark and Surrogate Models for hls4ml Resource and Latency Estimation

Authors: Audrey Corbeil Therrien[1]; Ben Hawks[2]; Dennis Plotnikov[3]; Dmitri Demler[None]; Donovan Sproule[4]; Elham Khoda[5]; Giuseppe Di Guglielmo[6]; Hamza Ezzaoui Rahali[7]; Jason Weitz[8]; Javier Mauricio Duarte[8]; Karla Tame-Narvaez[9]; Keegan Smith[10]; Mohammad Medhi Rahimifar[7]; Nhan Tran[11]; Russell Marroquin[12]; Vladimir Loncar[13]

[1] *Université de Sherbrooke*

[2] *Fermi National Accelerator Lab*

[3] *Johns Hopkins University (US)*

[4] *Columbia University*

[5] *University of Washington (US)*

[6] *Fermilab*

[7] *University of Sherbrooke*

[8] *Univ. of California San Diego (US)*

[9] *Fermilab National Accelerator Laboratory*

[10] *Texas A&M University*

[11] *Fermi National Accelerator Lab. (US)*

[12] *University of California San Diego*

[13] *CERN*

As machine learning (ML) is increasingly implemented in hardware to address real-time challenges in scientific applications, the
development of advanced toolchains has significantly reduced the time required to iterate on various designs. These advancements have
solved major obstacles, but also exposed new challenges. For example, processes that were not previously considered bottlenecks, such
as hardware synthesis, are becoming limiting factors in the rapid iteration of designs. To mitigate these emerging constraints, multipleefforts have been undertaken to develop an ML-based surrogate model that estimates resource usage of synthesized ML accelerator
architectures. We introduce wa-hls4ml, a benchmark for ML accelerator resource and latency estimation, and its corresponding
initial dataset of over 680 000 fully connected and convolutional neural networks, all synthesized using hls4ml and targeting Xilinx
FPGAs. The benchmark evaluates the performance of resource and latency predictors against several common ML model architectures,
primarily originating from scientific domains, as exemplar models, and the average performance across a subset of the dataset.
Additionally, we introduce GNN- and transformer-based surrogate models that predict latency and resources for ML accelerators. We
present the architecture and performance of the models and find that the models generally predict latency and resources for the 75%
percentile within several percent of the synthesized resources on the synthetic test dataset.

Posters and coffee / **131**

FPGA Toolchain Integration for CI/CD workflows

Authors: Andrew Whitbeck[1]; Ben Hawks[2]; Mariana Gonzalez[None]

[1] *Fermi National Accelerator Lab. (US)*

[2] *Fermi National Accelerator Lab*

Modern development flows that use tooling for automated building, testing, and deployment of software are becoming the norm for large scale software and hardware projects. These flows offer quite a few advantages that make them desirable, but when attempting to implement them for projects that use FPGAs, some complications can arise when attempting to integrate them with traditional FPGA toolchains and deployment workflows. In this poster, we present early our early efforts integrating the Vitis FPGA toolchain into a modern, Kuberentes and Github actions based CI/CD pipeline.

Posters and coffee / **132**

MLCommons Science Benchmarks

Authors: Ben Hawks[1]; Gregor von Laszewski[2]; Marco Colombo[None]; Nhan Tran[3]

[1] *Fermi National Accelerator Lab*

[2] *University of Virginia*

[3] *Fermi National Accelerator Lab. (US)*

Benchmarks are a cornerstone of modern
machine learning practice, providing standardized evaluations that enable reproducibility, comparison, and
scientific progress. Yet, as AI systems —particularly
deep learning models —become increasingly dynamic,
traditional static benchmarking approaches are losing

their relevance. Models rapidly evolve in architecture, scale, and capability; datasets shift; and deployment contexts continuously change, creating a moving target for evaluation. Without adaptive benchmarking frameworks, both scientific assessment and real-world deployment risk becoming misaligned with actual system behavior.
Drawing on our experience from MLCommons, educational initiatives, and government programs such as the DOE's Million Parameter Consortium, we identify key barriers that hinder the broader adoption and utility of benchmarking in AI. These include substantial resource demands, limited access to specialized hardware, lack of expertise in benchmark design, and uncertainty among practitioners about how to relate benchmark results to their own application domains. Moreover, current benchmarks often emphasize peak performance on leadership-class hardware, offering limited guidance for more diverse, real-world deployment scenarios.
We argue that benchmarking itself must become dynamic in order to incorporate evolving models, updated data, and heterogeneous computational platforms while maintaining transparency, reproducibility, and interpretability. Democratizing this process requires not only technical innovation, but also systematic educational efforts spanning undergraduate to professional levels to develop sustained expertise in benchmark design and use. Finally, benchmarks should be framed and communicated to support application-relevant comparisons, enabling both developers and users to make informed, context-sensitive decisions. Advancing dynamic and inclusive benchmarking practices will be essential to ensure that evaluation keeps pace with the evolving AI landscape and supports responsible, reproducible, and accessible AI deployment.

Contributed talks / 133

COLLIDE-2V - 750 Million Dual-View LHC Event Dataset for Low-Latency ML

Authors: Eric Anton Moreno[1]; Philip Oliver Ploner[2]; Ranit Das[3]

Co-authors: Abhijith Gandrakota [4]; Alex Tapper [5]; Benedikt Maier [6]; David Shih ; Javier Mauricio Duarte [7]; Jennifer Ngadiuba [8]; Maciej Mikolaj Glowacki [9]; Mia Liu [10]; Philip Coleman Harris [11]; Shiqi Kuang [12]; Thea Aarrestad [2]

[1] *Massachusetts Institute of Technology (US)*

[2] *ETH Zurich (CH)*

[3] *Rutgers University*

[4] *Fermi National Accelerator Lab. (US)*

[5] *Imperial College London*

[6] *Imperial College (GB)*

[7] *Univ. of California San Diego (US)*

[8] *FNAL*

[9] *CERN*

[10] *Purdue University*

[11] *Massachusetts Inst. of Technology (US)*

[12] *Purdue University (US)*

Modern foundation models (FMs) have pushed the frontiers of language, vision, and multi-model tasks by training ever-larger neural networks (NN) on unprecedented volumes of data. The use of FM models has yet to be established in collider physics, which both lack a comparably sized, general-purpose dataset on which to pre-train universal event representations, and a clear demonstrable need. Real-time event identification presents a possible need due to a requirement for fast event classification and selection of all possible collisions at the LHC. As a result, we construct a dual-view LHC collision dataset (COLLIDE-2V), a 50TB public dataset comprising ˜750 million proton-proton events generated with MadGraph + Pythia + Delphes under High-Luminosity LHC conditions (<μ> = 200). Spanning everything from minimum-bias and γ+jets to top, Higgs, di-boson, multi-boson, exotic long-lived signatures and dark showers, the sample covers 50+ distinct processes and >99% of the CMS Run-3 trigger menu in a single coherent format. To allow for effective real-time event interpretation each event is provided twice, as Parquet files which retain physics-critical content:

- Offline - a full CMS-like reconstruction emulated by a tuned Delphes card
- L1T - a low-latency, lower-resolution view obtained via a custom Level-1 Trigger (L1T) parameterisation (degraded vertex, track and calorimeter performance, altered puppi, $|\eta| \leq 2.5$ tracking, pT thresholds, etc.)

As a proof-of-concept, COLLIDE-2V supports a wide spectrum of research applications ranging from few-shot transfer learning, fine-tuning, pileup mitigation, detector-level generative modelling, cross-experiment benchmarking, to fast simulation surrogates and real-time trigger inference, and entirely novel anomaly-detection - thereby accelerating the shift from handcrafted topology cuts to data-driven decision making throughout the HL-LHC program.

Contributed talks / 134

KAN-LUT: Efficient LUT-Based Acceleration of Kolomogorov-Arnold Networks (KANs) on FPGAs

Authors: Aarush Gupta[1]; Duc Hoang[1]; Philip Coleman Harris[1]; Vladimir Loncar[2]

[1] *Massachusetts Inst. of Technology (US)*

[2] *CERN*

Optimized FPGA implementations of tiny neural networks are crucial for low-latency and hardware-efficient inference for a variety of applications. Neural networks based on lookup tables (LUTs) are a standard technique for such problems due to their hardware efficiency and strong expressivity. However, such networks are often difficult to scale up as their resource usage scales exponentially with LUT fan-in. To address this issue, we propose a LUT-based implementation of the recently proposed Kolmogorov-Arnold Network (KAN). KANs consist of spline-based, trainable activations as edges between neurons in adjacent layers, with each node performing a sum operation on incoming activations. Because of the strong expressivity of spline-based activations, KANs can often achieve similar accuracies as compared to multi-layer perceptrons (MLPs) using significantly fewer layers. Since each node-to-node spline computation is performed with a LUT lookup, the fan-in of each LUT is only unity which avoids scaling issues associated with other LUT-based networks. Along with quantization-aware training (QAT), this architecture is well-suited for edge-pruning to decrease hardware resources after sparsification in training. Empirically, we demonstrate on various benchmarks that our design achieves task performance similar to other state-of-the-art techniques while also using comparable or fewer hardware resources.

Posters and coffee / 136

Integrating Support for Google XLS in hls4ml

Author: Andrei Girjoaba[None]

Co-authors: Benjamin Ramhorst [1]; Dimitrios Danopoulos [2]; Vladimir Loncar [2]

[1] *ETH Zurich*

[2] *CERN*

As Moore's Law comes to an end, domain-specific architectures (DSA) are considered the next direction for performance improvements in compute. Unfortunately, the development environment of DSAs falls short in comparison to that of general-purpose architectures (e.g., CPUs). The transition from general-purpose to DSA is hindered by the fact that software engineers lack the knowledge to transition to hardware design. Even if they do, the tool landscape from which they must choose is highly partitioned with multiple competing technologies for tasks such as simulation, verification, synthesis, etc.

At CERN, not only software engineers are interested in hardware designs, but also physicists that must develop latency-critical triggers that filter interesting particle interactions or collisions. Such triggers can be implemented as machine learning (ML) models performing classification. To aid in this endeavour, hls4ml is an open-source project that unifies multiple high-level synthesis (HLS) toolkits and ML libraries. Internally, a ML library is chosen as the frontend, then it is translated into the hls4ml intermediate representation (IR), and finally is written as input for an HLS backend that synthesizes the result with the generated hardware description language.

Externally, it allows physicists or software engineers to directly translate their ML models to field-programmable gate array (FPGA) designs. Having multiple HLS toolkits acting as backends extends vendor compatibility while reducing the complexity of experimenting and finding the best suitable backend. By using hls4ml, most of the complexities of hardware design are abstracted away.

With this poster, we present the implementation of the XLS backend in the hls4ml project. XLS is an open source HLS toolkit from Google that is suitable for the hls4ml flow with no licensing requirement. XLS distinguishes itself by providing its domain specific language (DSL) called DSLX. Most HLS tools provide an extension of C/C++ as their frontend, which is criticised to be ill-suited for circuit design due to its Von Neumann architecture nature. DSLX claims to be a dataflow DSL based on Rust. In combination with the fact that it was built for hardware design, it is well-suited for both hardware and ML designs. This is seen by enforcing many good practices —already integrated in the hls4ml IR, such as per-layer bit precision inference—in the language itself. Another benefit is demonstrated by the ease of implementing other advanced features. Using "procs", streaming convolutional layers and stateful LSTM cells can be implemented with reduced development time.

In our preliminary results, in the fully parallel setting with an initiation interval of one, we obtain two times fewer Look-Up Tables and three times latency improvement compared to the Vitis HLS backend. On top of the development benefits, we note the potential hardware efficiency benefits that could be achieved with this XLS backend.

We wish to highlight the main design decisions, the advantages and the difficulties when writing the XLS backend, while also comparing the synthesized results with already existent backends.

Posters and coffee / 137

AutoDeploy-HEP: An Intelligent Toolkit for ML Deployment on Heterogeneous Hardware in High-Energy Physics

Author: MUSTOFA ABDULHAFIZ AHMED mustofa[None]

Deploying ML models today requires deep expertise in both hardware and software optimization. It often involves laborious trial-and-error to determine the right combination of tools, techniques, and configurations. While industry and academia benefit from a wide array of deployment frameworks and automation tools, the High-Energy Physics (HEP) community still faces major challenges in adopting and adapting these techniques. The fragmentation and volume of available methods make it nearly impossible for HEP engineers to explore and implement optimal solutions for each specific use case, leading to slow, non-scalable, and often suboptimal deployment workflows.

We propose **AutoDeploy-HEP**, an adaptive toolkit that bridges this gap by automating the end-to-end deployment of ML models across heterogeneous hardware platforms such as GPUs, FPGAs, NPUs, and DCUs. Given a model and deployment objectives via configuration, CLI, or API, the system uses a continually evolving **knowledge graph** to recommend efficient deployment pipelines composed of architecture optimization, model compression, and hardware acceleration steps. These pipelines are executed in a feedback-driven manner, adapting dynamically to intermediate results.

Over time, the toolkit evolves from rule-based logic to learning-driven intelligence using knowledge graph embeddings, ML-based recommenders, and inference cost predictors to optimize decisions before deployment. AutoDeploy-HEP is designed to support a wide range of HEP applications, across experiments and hardware setups, empowering researchers to deploy ML models efficiently without needing deep deployment expertise, and accelerating scientific innovation in the field.

Posters and coffee / **138**

Inverse Design for Femtosecond-Laser Photonic Surfaces with Direct Gradient Optimization

Author: Amine Haboub[1]

Co-authors: Abdelkrim Khelif [2]; Brahim Aissa [1]

[1] *Qatar Environment and Energy Research Institute*

[2] *Hamad Bin Khalifa University*

The inverse design of photonic surfaces produced by high-throughput femtosecond laser processing is limited by a strongly non-linear, many-to-one mapping from laser parameters (power, speed, hatch spacing) to the resulting optical spectrum. Tandem Neural Networks (TNNs) mitigate this ill-posedness by pairing a forward surrogate with a separately trained inverse network, but they still rely on artificial noise injection to uncover diverse solutions and still explore the design space only sparsely.

We propose Direct Gradient Optimization (DGO), a single-network alternative that treats the pre-trained forward surrogate as a differentiable proxy for the laser–material interaction and back-propagates errors all the way to the process parameters. This strategy (i) eliminates inverse-model training, (ii) allows the embedding of practical constraints such as laser-power penalties to be encoded directly in the loss function, and (iii) yields transparent sensitivity information through native gradients and SHapley Additive exPlanations (SHAP).

Two optimization modes are assessed: a single-start DGO and a parallel multi-start "Tournament" DGO that launches multiple random seeds, runs a brief qualification phase, and refines only the five most promising candidates. Across 10000+ inverse-design tasks, Tournament-DGO cuts the average spectral root-mean-squared error (RMSE) from 1.29 % (best TNN) to 0.70 %, and boosts design novelty (quantified by the Normalized Euclidean Parameter Distance, NEPD) from 0.26 to 0.38. A SHAP-based meta-analysis shows that convergence is dominated by the initial hatch spacing, explaining why broad sampling followed by pruning addresses this sensitivity.

DGO delivers state-of-the-art accuracy, enhanced robustness, and diverse inverse solution without noise heuristics or extra model training. Its ability to embed manufacturability constraints directly in the optimization loop establishes a powerful, energy-aware platform for scalable photonic devices, thermal emitters, and meta surfaces.

Keywords: inverse design; femtosecond laser processing; photonic surfaces; machine learning

References
1 Park, M., Grbčić, L., Motameni, P., Song, S., Singh, A., Malagrino, D., ⋯Lubner, S. D. (2024). Inverse design of photonic surfaces via high-throughput femtosecond laser processing and tandem neural networks. Advanced Science, 11(26), 2401951.
2 Grbčić, L., Park, M., Elzouka, M., Prasher, R., Müller, J., Grigoropoulos, C. P., ⋯de Jong, W. A. (2025). Inverse design of photonic surfaces via a multi-fidelity ensemble framework and femtosecond laser processing. npj Computational Materials, 11(1), 35.

Contributed talks / **139**

Balancing Prediction Performance, Transparency and Energy Consumption in Machine Learning Models for Data Streams

Authors: Kirsten Köbschall[None]; Lisa Hartung[None]; Raphael Fischer[None]; Sebastian Buschjäger[None]; Stefan Walter Theodor Kramer[None]

In the era of continuous data generation, real-time processing of data streams has become crucial for timely, adaptive, and context-aware decision-making. However, maintaining effective learning models in such dynamic environments requires carefully balancing prediction performance, transparency and energy consumption.

In the talk, we will present two new state-of-the-art methods for classification on data streams in such settings: (i) SoHoTs (soft Hoeffding trees) for balancing prediction performance and transparency, and (ii) HEROS (Heterogeneous Online Ensembles) for balancing prediction performance and energy consumption. SoHoTs are transparent, differentiable decision tree models for data streams. They employ a novel routing mechanism based on the Hoeffding inequality and adapt to changing data distributions through gradient-based weight updates, like soft decision trees. They process data in real-time, one sample at a time, without the need for storage, and enhance interpretability via decision-rule-based feature importance, sparse activation, and visualized decision paths. To study the trade-off between prediction performance and energy consumption, we introduce HEROS, which avoid expensive hyperparameter optimization by maintaining a diverse pool of preconfigured models. At each time step, HEROS select a resource-aware subset of models for training. A novel zeta-policy is introduced to guide this selection process, prioritizing models that deliver near-optimal performance under strict resource constraints. Empirical evaluations across 20 data streams (SoHoTs) and 11 benchmark datasets (HEROS) demonstrate that both methods achieve strong predictive performance while ensuring transparency or reduced resource consumption.

Posters and coffee / **140**

A Hybrid Architecture for Real-Time Experiment Feedback in X-ray Microscopy Enabled by Edge and Cloud Computing

Authors: Adam Thompson[None]; Darren Battey[1]; Denis Leshchev[None]; Felix Bachmair[2]; Michael Rissi[None]; Sascha Grimm[None]

[1] *Diamond Light Source*

[2] *Dectris Ltd.*

Ptychographic imaging generates high-resolution datasets at the cost of heavy computational complexity, limiting its use in real-time experimental decision-making. In this cross-institutional effort, we introduce a hybrid edge-to-cloud workflow that delivers fast feedback for ptychography experiments by combining a modern synchrotron beamline at Diamond Light Source I13-1, featuring an upgraded motion control system for rapid probe scanning, with a high-speed detector (DECTRIS SELUN), a GPU-accelerated edge processing pipeline, and DECTRIS CLOUD services.

Our approach integrates on-prem edge computing for immediate reconstruction previews or anomaly detection with cloud-based processing pipelines for full-resolution analysis. This will enable scientists to monitor sample quality and adapt acquisition parameters within seconds to minutes, aiming to improve experimental efficiency and data quality.

This framework was prototyped during a joint hackathon and demonstrated during initial measurements at Diamond Light Source (Beamline I13-1). It pivots how scientific experiments can benefit from near real-time feedback and enables the integration of modern ML workflows, edge computing, and scalable cloud infrastructure. We discuss our architectural design and outline future steps toward deployable real-time feedback systems that pave the way towards autonomous exploration in materials science and X-ray imaging.

Invited talks / 141

Wearable TinyML Platforms for Biosignal Intelligence Across the Body

Most commercial wearables still capture only basic metrics such as step counts or heart rate, and remain closed systems without access to raw data. In this talk, I will present our holistic approach to full-body biosignal intelligence, where ultra-low-power embedded platforms and machine learning algorithms are co-designed to capture and process signals from the brain, eyes, muscles, and cardiovascular system in real time, while sustaining day-long battery lifetimes. I will show how open, modular platforms can be adapted into diverse wearable form factors, and how tailored ML algorithms make these signals usable for applications such as seizure detection and eye-movement classification. Finally, I will discuss how this vision extends to emerging modalities such as wearable ultrasound, representing the next leap in multimodal, ML-enabled wearables.

Invited talks / 142

Real-time ML and neuromorphic computing for smart robots

Invited talks / 143

The Instrumental Edge: Real-Time and AI-Ready Scientific Discovery

From radio telescopes to particle accelerators and electron microscopes, scientific instruments produce tremendous amounts of data at equally high rates; previous architectures that have relied on offline storage and large data transfers are unable to keep up. The future of scientific discovery is interactive, streaming, and AI driven, placing the autonomous and intelligent instrument at the center of a given science workflow. By making our instruments smarter, scientists can run higher impact experiments.

In this tools-focused talk, we will highlight NVIDIA's work with real-time data processing including high speed data movement, FPGA co-design, and laying the foundations for moving AI inferencing as close to the data converter as possible.

Invited talks / 144

AI for next-gen cellular networks (6G)

Invited talks / **145**

Scaling up Advanced, Near-Sensor AI: An Open Platform Approach

AI is accelerating into the generative era, and it is poised to disrupt multiple businesses and applications. With the increasing focus on edge and extreme-edge, near sensor applications, inference is becoming the key workload and computational challenge. Computing system need to scale out and scale up to meet the challenge. In this talk I will discuss how to scale up chip(lets) for efficient inference at the edge targeting advanced AI models, optimizing the whole hardware stack, from processing elements to the global interconnect. I will emphasize the strategic importance of an end-to-end (models, software, instruction set architecture, digital design) open-platform approach to ensure a healthy innovation ecosystem, with long term sustainability.

Invited talks / **146**

Co-Design for Efficient & Adaptive ML

Beyond the well-known highlights in computer vision and natural language, AI is steadily expanding into new application domains. This Pervasive AI trend requires supporting diverse and fast-moving application requirements, ranging from specialized I/O to fault tolerance and limited resources, all the while retaining high performance and low latency. Adaptive compute architectures such as AMD FPGAs are an excellent fit for such requirements but require co-design of hardware and ML algorithms to reap the full benefits. In this talk, we will cover a breadth of co-design techniques, including their merits and challenges, from streaming dataflow architectures to quantization, from sparsity to full circuit co-design. By combining such techniques, we can enable nanosecond-latency and performance in the hundreds of millions of inferences per second. The proliferation of this technology is enabled via open-source AMD tools such as FINN, Brevitas and LogicNets, as well as the AMD-FastML collaborative project QONNX.

Invited talks / **147**

Efficient Graph neural networks at Google

Graph Neural Networks (GNNs) are a powerful paradigm for Neural Net ML models to operate on relational data or data with structural information. This talk explores the practical use and ongoing research on GNN done at Google for industrial applications. We provide a brief overview of GNNs modeling, including GCNs, Graph Transformers, and geometric-aware models. Then we discuss a variety of real-world applications. Finally, we talk about scaling challenges on very large graphs, dynamic graphs, and fast inference on specialized hardware acceleration.

Invited talks / **148**

Fast inference with Decision Forests

Decision Forests such as Random Forests and Gradient Boosted Trees are an effective and widely used class of models for machine learning, particularly for tabular data and forecasting. This talk covers the practical use and ongoing research on Decision Forests at Google. We provide a brief overview of decision forest modeling with a focus on novel split conditions. We will analyze their impact on model quality as well as on performance characteristics during training and inference. Then, we discuss a variety of real-world applications for these models. Finally, we will explore how algorithmic approaches to structuring tree traversal can be optimized for diverse hardware architectures, including CPUs, GPUs, and FPGAs.

Invited talks / 149

Next Generation GPU Signal Processing Pipeline for Radio Astronomy

As digitizer technologies scale, efficient processing of massive amounts of sensor data is essential for the next generation of science projects. This talk focuses on the next-generation electromagnetic signal processing pipeline developed at the Allen Telescope Array. Backed by the NVIDIA Holoscan SDK, this pipeline utilizes cutting-edge technologies to address the three key pillars of high-speed scientific sensor data acquisition in real-time. In this model, the GPU device acquires, processes, and stores the data without the direct involvement of the CPU memory. This topology, apart from performance gains, enables the deployment of cutting-edge detection technology based on machine learning models that directly work with spectrogram data in an online fashion. This work equips observatories with the capability to meet the growing demands of next-generation astronomy.

Contributed talks / 150

State space models for Project 8 event reconstruction

Authors: Christina Reissel[1]; Hannah Binney[None]; Joseph Formaggio[None]; Kyungseop Yoon[2]; Philip Coleman Harris[1]

[1] *Massachusetts Inst. of Technology (US)*

[2] *Massachusetts Institute of Technology*

The Project 8 experiment aims to directly probe the neutrino mass by precisely measuring the energy spectrum of beta electrons emitted in the decay of tritium. The collaboration has pioneered the cyclotron radiation emission spectroscopy technique (CRES), which measures the energy of single electrons by detecting the cyclotron radiation they emit in a magnetic field. Traditional methods for event reconstruction rely on detecting tracks in a spectrogram after transforming the voltage output into the frequency domain. With the goal of achieving 0.3 eV root mean square (rms) energy resolution in the next prototype, these frequency-based methods face challenges, such as how to determine the electron's location in the detector volume. State space models (SSMs) have shown promise for performing well on long time series data with good computational efficiency. In this work, we will demonstrate that the diagonal structured state space architecture (S4D) shows potential for reconstructing event parameters directly from the voltage time series in high-fidelity Project 8 simulations. The architecture's minimal operations and good efficiency also opens the possibility for real-time reconstruction at the 400 MHz sampling frequency.

151

Registration

152

Commute to ETH Main Building (by ETH link 17:34 or 17:54). Meet Polyterrasse

Tutorials / 153

Part 2: Agile Hardware Design for AI: A Hands-On Tutorial with SODA and Bambu

Authors: Nicolas Bohm Agostini[1]; Giovanni Gozzi[2]; Michele Fiorito[2]; Serena Curzel[2]; Vito Giovanni Castellana[3]; Antonino Tumeo[3]; Fabrizio Ferrandi[2]

[1] *Northeastern University and Pacific Northwest National Laboratory*

[2] *Politecnico di Milano*

[3] *Pacific Northwest National Laboratory*

This tutorial explores the growing demand for domain-specific hardware accelerators driven by the rapid evolution of AI and data analytics. Traditional hardware design cycles are too slow to keep up with the pace of algorithmic innovation. To address this, new agile hardware design methodologies are emerging, leveraging compiler technologies and High-Level Synthesis (HLS) to automate and accelerate the design process.

The tutorial introduces the SODA Synthesizer, an open-source, compiler-based toolchain that enables the generation of efficient hardware accelerators from high-level algorithm descriptions.

It consists of:

- SODA-OPT: A front-end and optimizer built on the MLIR (Multi-Level Intermediate Representation) framework. It interfaces with popular Python-based data science and machine learning frameworks, performing hardware/software partitioning and domain-specific optimizations.
- Bambu: A state-of-the-art open-source HLS tool developed at Politecnico di Milano. It translates optimized high-level code into hardware designs, supporting both FPGA and ASIC targets, and integrates with RTL simulation and logic synthesis tools.

The tutorial highlights the limitations of traditional HLS tools, which were primarily designed for digital signal processing and required expertise in hardware description languages (HDLs). Modern HLS tools now support parallel programming models and integrate with high-level frameworks, making them more accessible to software developers and domain scientists.

Key topics include:

1. Current trends and methodologies in agile hardware design.
2. Advantages and limitations of conventional HLS approaches.
3. The role of MLIR in supporting diverse domains and frameworks.
4. Hands-on demonstrations of both SODA-OPT and Bambu, showing how to go from Python code to optimized hardware accelerators.

The tutorial also demonstrates how the toolchain integrates with both commercial and open-source Electronic Design Automation (EDA) tools, such as OpenROAD, enabling a complete end-to-end flow from Python to silicon. The design flow is flexible and supports both FPGA prototyping and ASIC implementation, making it suitable for a wide range of applications in AI and data-intensive computing.

This tutorial is intended for researchers, engineers, and developers interested in accelerating AI and data-driven applications through agile hardware design, even without deep expertise in traditional hardware design languages.

References

 [1] G. Gozzi, M. Fiorito, S. Curzel, C. Barone, V. G. Castellana, M. Minutoli, A. Tumeo, and F. Ferrandi: SPARTA: High-Level Synthesis of Parallel Multi-Threaded Accelerators. ACM Trans. Reconfigurable Technol. Syst. 18, 1, Article 9 (March 2025).

 [2] N. Bohm Agostini, S. Curzel, J. Zhang, A. Limaye, C. Tan, V. Amatya, M. Minutoli, V. G. Castellana, J. B. Manzano, D. Brooks, G-Y. Wei, A. Tumeo: Bridging Python to Silicon: The SODA Toolchain. IEEE Micro 42(5): 78-88 (2022) (Best paper for 2022)

 [3] N. Bohm Agostini, S. Curzel, V. Amatya, C. Tan, M. Minutoli, V. G. Castellana, J. B. Manzano, D. R. Kaeli, A. Tumeo: An MLIR-based Compiler Flow for System-Level Design and Hardware Acceleration. ICCAD 2022: 6:1-6:9

 [4] F. Ferrandi, V. G. Castellana, S. Curzel, P. Fezzardi, M. Fiorito, M. Lattuada, M. Minutoli, C. Pilato, A. Tumeo: Invited: Bambu: an Open-Source Research Framework for the High-Level Synthesis of Complex Applications. DAC 2021: 1327-1330

Invited talks / **155**

Conference summary

Invited talks / **156**

Poster prizes and closing

Posters and coffee / **157**

Poster & Coffee

158

Satellite event: hls4ml dev meeting

For minutes of the discussion, see https://indico.cern.ch/event/1586270/

Contributed talks / **159**

An ML Pipeline for Real-time Gravitational Wave Alerts

Contributed talks / 160

Edge Deep Learning for Particle Physics (EPIGRAPHY)

161

Conference photo outside of auditorium!

162

Map to Tessin Grotto

Contributed talks / 163

Get your poster!