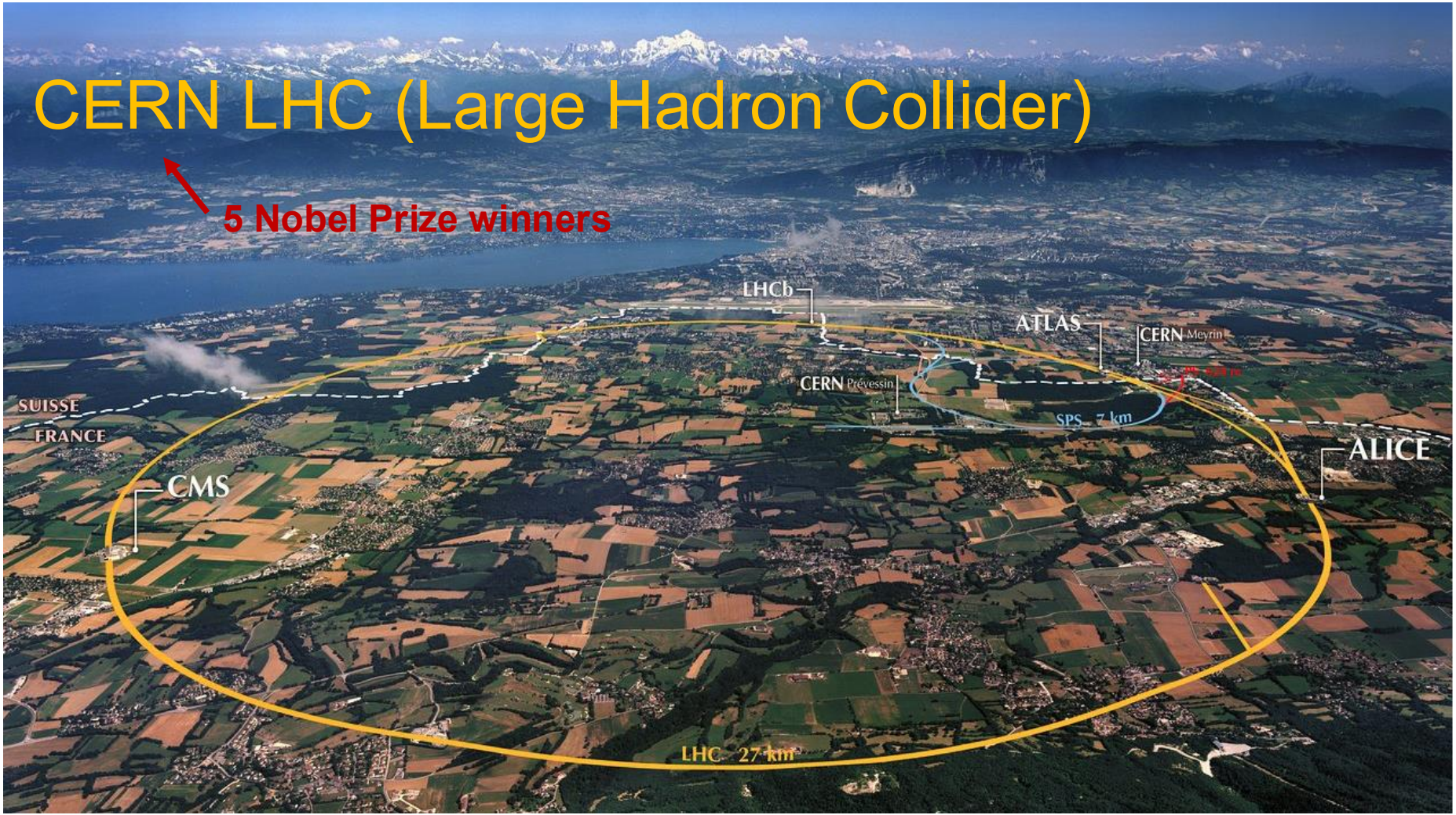
JEDI-linear: Fast and Efficient Graph Neural Networks for Jet Tagging on FPGAs

Zhiqiang (Walkie) Que, Chang Sun, Sudarshan Paramesvaran, Emyr Clement, Katerina Karakoulaki, Christopher Brown, Lauri Laatu, Arianna Cox, Alexander Tapper, Wayne Luk, Maria Spiropulu

Imperial College London, Caltech, University of Bristol, CERN
FPT 2025

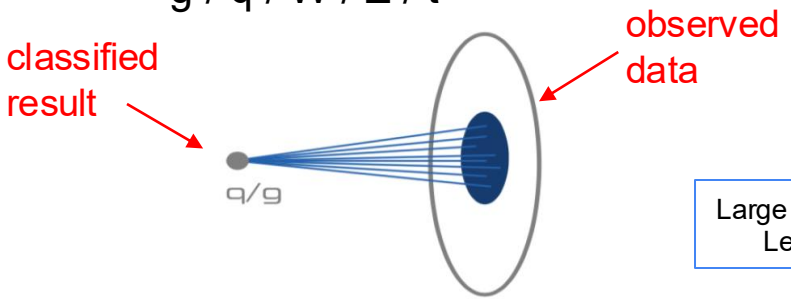
Artifacts Evaluation: Available, Evaluated Functional, Reusable, Results Replicated

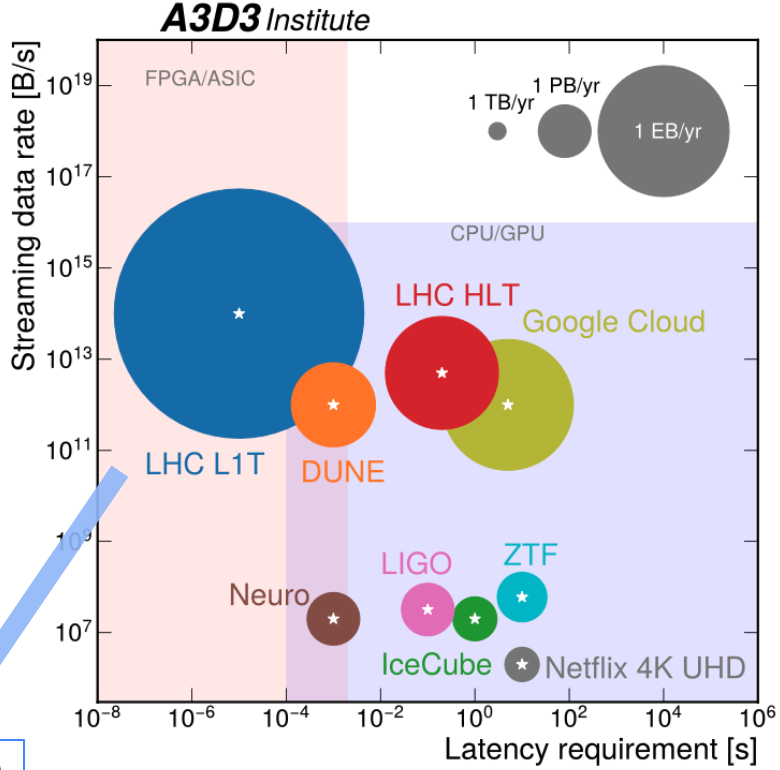

CERN LHC (Large Hadron Collider)
5 Nobel Prize winners
LHCb
ATLAS
CERN Meyrin
CERN Prévessin
SPS 7 km
SUISSE
FRANCE
CMS
ALICE
LHC 27 km

Enormous Data and Low Latency Challenge

- Next-gen key experiments
  - up to Petabyte per second
  - sub-100 ns latency
  - cannot store everything

- **Jet Tagging**
  - critical for selecting interesting events
  - classify into 5 particle types initiating a jet:
    g / q / W / Z / t





Large Hadron Collider (LHC)
Level-1 Trigger (L1T)

Source: A3D3 Institute, https://a3d3.ai/

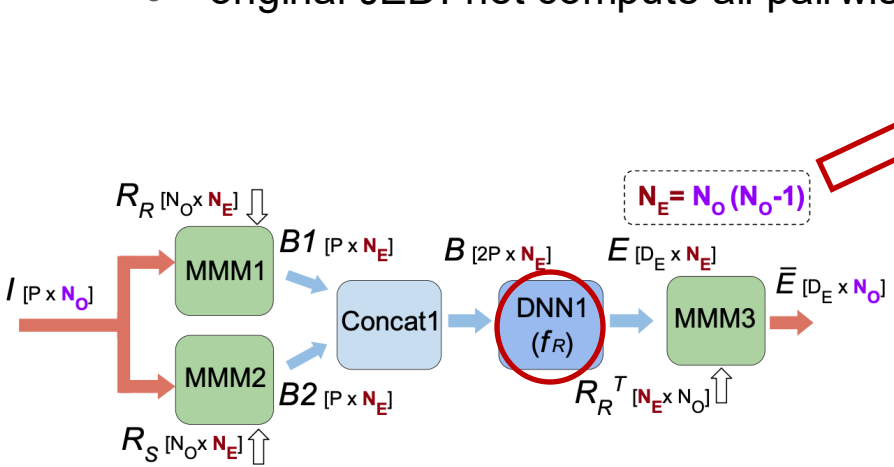
Overview

- Level 1 Trigger Challenge
  - jet tagging algorithm functions well
  - but too slow and too resource intensive
- Our innovations
  - linear-complexity interactions by Global Context Vectors
  - fine-grained mixed-precision quantization
  - distributed arithmetic
- Results
  - sub-100 ns latency
  - Digital Signal Processing (DSP) blocks of FPGA not required
  - best accuracy among state-of-the-art designs

Graph Neural Networks (GNNs)

- Why GNNs?
  - jets are naturally graph-structured
  - current approaches (like JEDI-net) adopt GNNs for particle-level relations
  - offer state-of-the-art accuracy
- Why challenging?
  - original JEDI-net compute all pairwise edges → **O(N²)** edge MLP $\mathbf{f_R}\,(\mathbf{I_i} \,||\, \mathbf{I_j})$

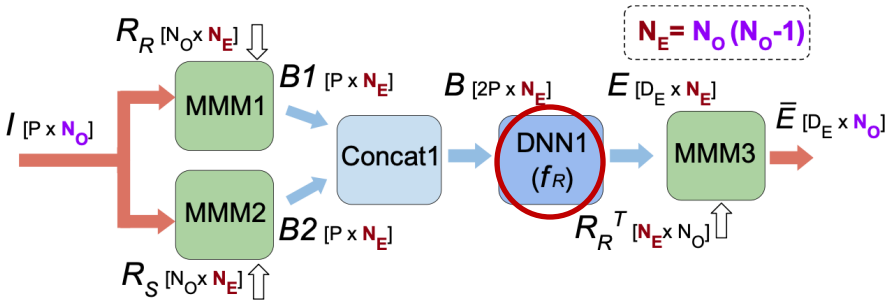
MLP: Multi Layer Perceptron



Graph Neural Networks (GNNs)

- Why GNNs?
  - jets are naturally graph-structured
  - current approaches (like JEDI-net) adopt GNNs for particle-level relations
  - offer state-of-the-art accuracy
- Why challenging?
  - original JEDI-net compute all pairwise edges → **O(N^2)** edge MLP $\mathbf{f_R}$ ($\mathbf{I_i}$ || $\mathbf{I_j}$)
  - prior FPGA-based GNNs [**TECS'24**, **MLST'24**] hit latency / resource walls (coarse-grained DSPs > 8.7k, fine-grained Lookup Tables LUT > 1M)

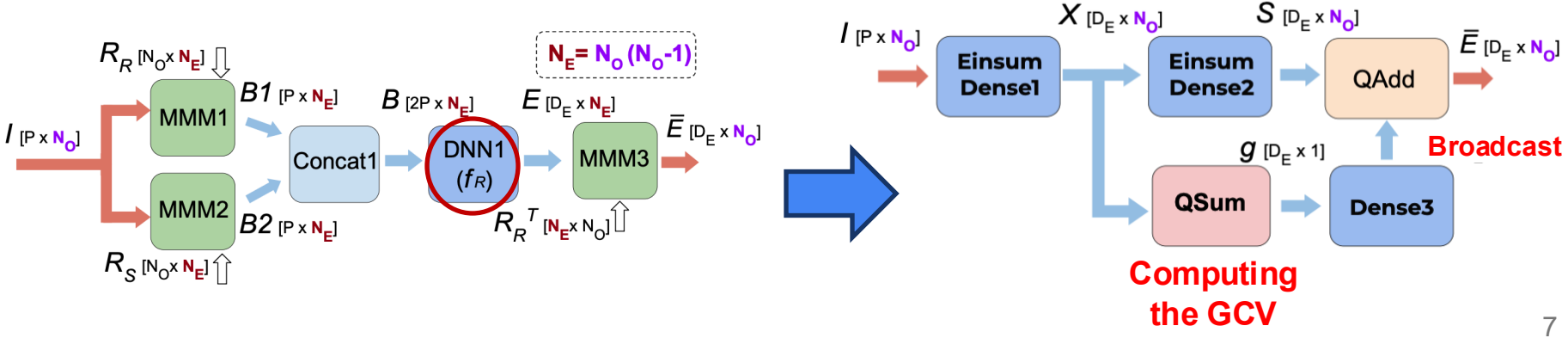
MLP: Multi Layer Perceptron



[**TECS'24**] Z. Que et al., "LL-GNN: Low latency graph neural networks on FPGAs for high energy physics." *ACM TECS 2024*
[**MLST'24**] P. Odagiu, et al., "Ultrafast jet classification at the HL-LHC." *Machine Learning: Science and Technology*, 2024.

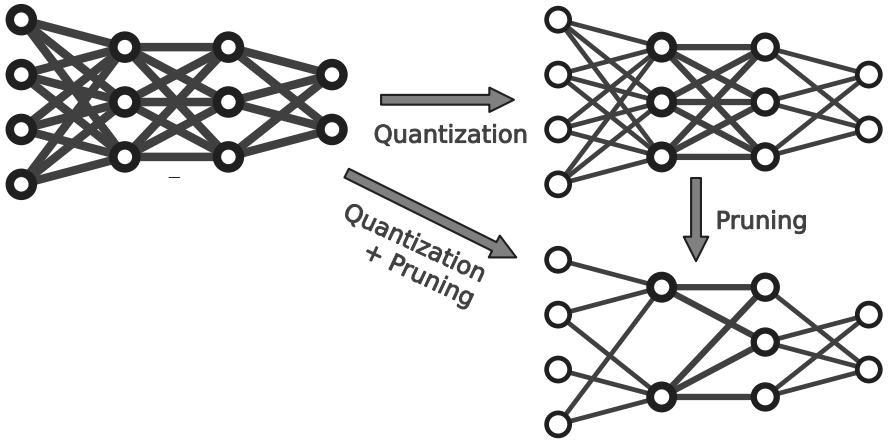
Key idea 1: Linearize interactions: Global Context Vector

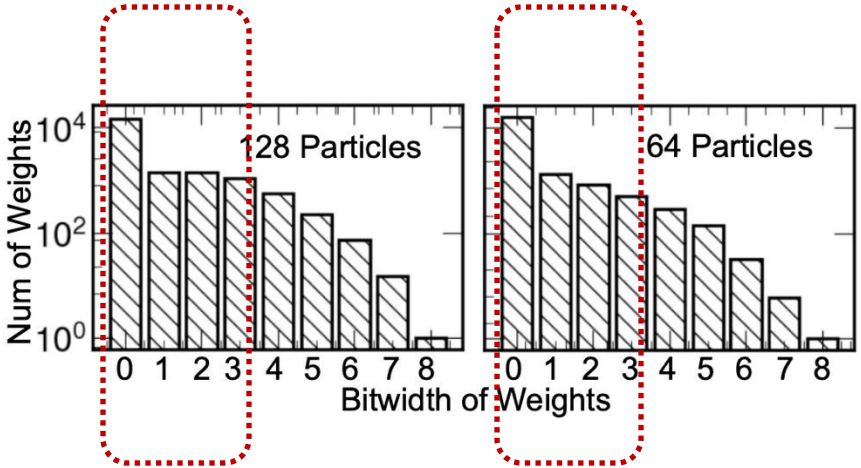
- Original JEDI-net: **O(N^2)** edge MLP $\mathbf{f_R\,(I_i \,||\, I_j)}$
- Explicit pairwise computation for global aggregation
  - compute a **Global Context Vector** (GCV): $\mathbf{W_2 \cdot \frac{1}{N} \sum_j I_j}$
  - **broadcast** this vector to update individual particles
- Each particle sees the jet's global average + its own local features
  - still interaction-aware, but linear
  - reduces complexity from **O(N^2) to O(N)**



Key idea 2: Fine-Grained Mixed-Precision Quantization

- per-weight bitwidth: a trainable variable
- **Quantization-Aware Training:** effective bitwidth in training loss
- Integrated pruning: weights with bitwidth as 0 are pruned automatically
- Majority of non-zero weights < 4 bits, high sparsity and high accuracy





Key idea 3: Distributed Arithmetic (DA)

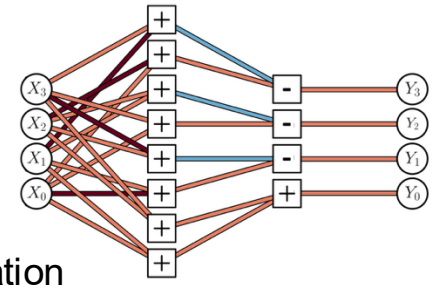
- Conventional designs
  - rely on fixed-point multipliers for Constant Matrix-Vector Multiplications (CMVMs)
- Replace CMVMs with DA adder trees: using da4ml (Journal Session 2)
- Decompose CMVM operations
  - into a static graph of shift-addition/subtraction operations: for fine-grained resources
- Common subexpression elimination
  - remove redundant adders/subtractors

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Naive implementation

6 adders

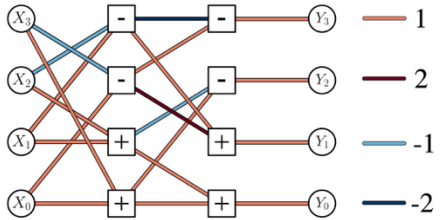
$Y_0 = X_0 + X_1 + X_2 + X_3$
$Y_1 = 2X_0 + X_1 - X_2 + X_3$



DA optimized

5 adders

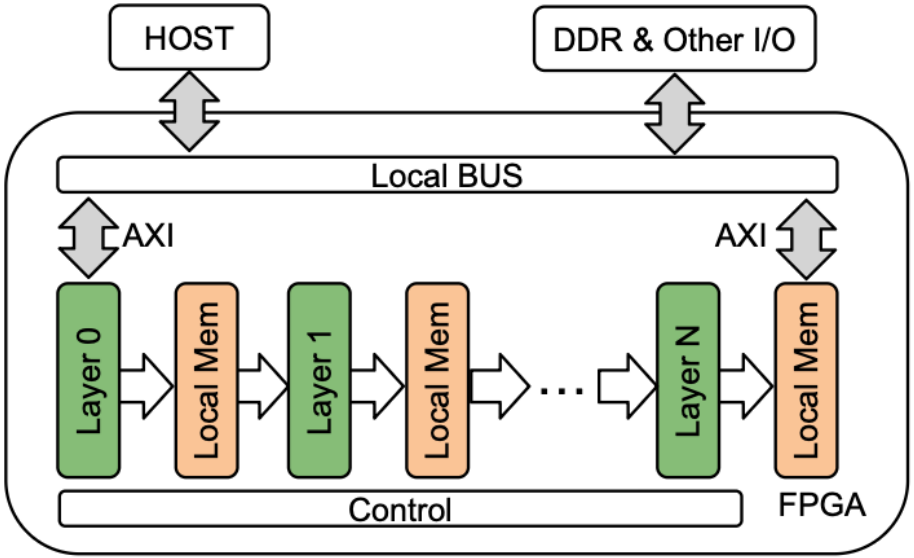
$\textbf{Temp} = X_1 + X_3$
$Y_0 = X_0 + X_2 + \textbf{Temp}$
$Y_1 = 2X_0 - X_2 + \textbf{Temp}$



Automated Design Flow with Optimized Architecture

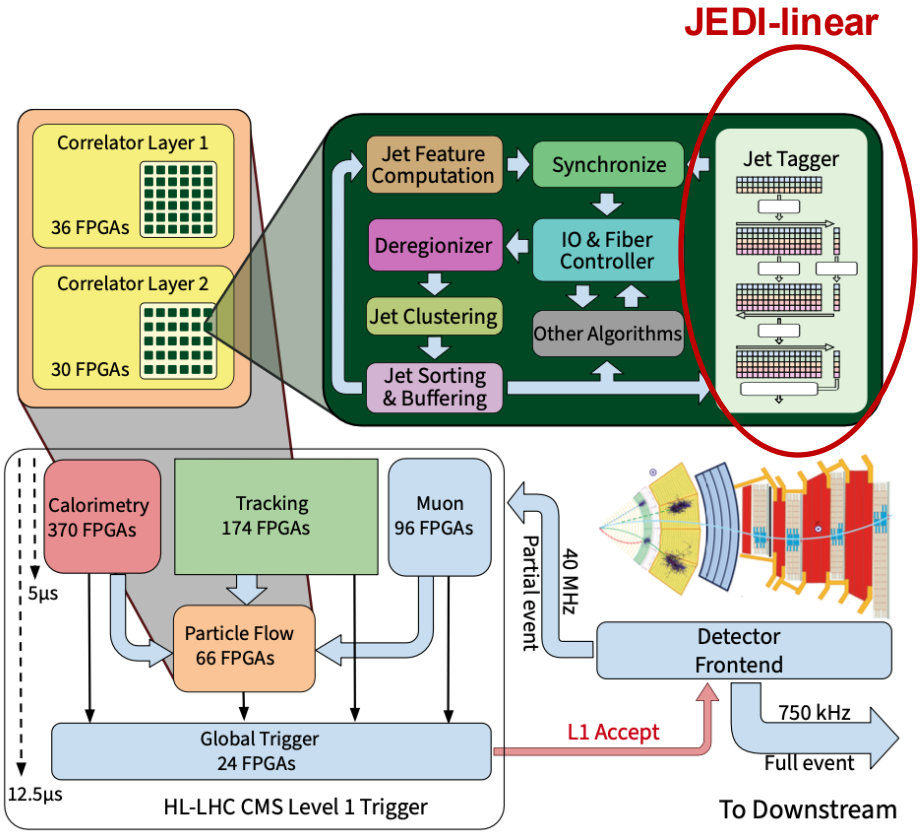
- Fully unrolled, static pipelined dataflow architecture
  - no resource sharing; data kept on-chip
  - deterministic latency: crucial for trigger synchronization

- Extend symbolic tracing capabilities of da4ml tool
  - Python-based model
    → Enhanced *da4ml*
    → Synthesizable RTL
  - generated RTL models: functionally validated by Verilator



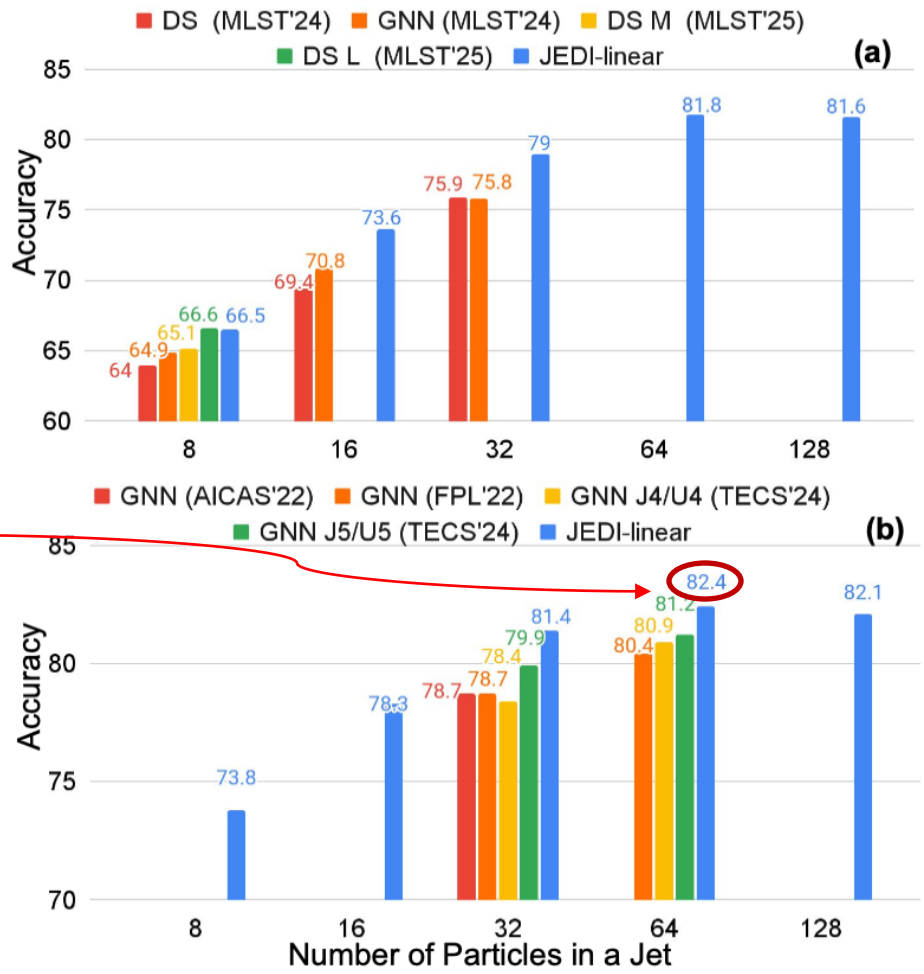
Integration in CMS Correlator Trigger Layer 2 (CTL2)

- CTL2
  - 30 VU13P FPGAs, 5 x 6 FPGA nodes
  - 6 nodes available per event
- Jet Tagger sits after jet clustering / sorting / buffering
- Latency budget
  - O(100 ns) per algorithm
  - deterministic for synchronization
- Our design
  - latency < 60~110 ns



Evaluation: Accuracy

- JEDI-linear consistently outperforms GNN baselines
- 16 features, 64 particles, accuracy for
  - JEDI-linear: up to **82.4%**
  - others: 80.4 … 81.2%
- improved accuracy vs prior FPGA-based architectures



Evaluation: Latency & Resources

Model	Particles	Features	Acc. (%)	Latn. (ns)	DSP	LUT (k)	FF (k)	BRAM	II (clk)	F_{max} (MHz)
DS (MLST'24) [7]	8	3	< 64.0	95	626	386	121	4	3	N/A
DS (MLST'24) [7]	16	3	< 69.4	115	555	747	239	4	3	N/A
DS (MLST'24) [7]	32	3	< 75.9	130	434	903	359	4	2	N/A
GNN (MLST'24) [7]	8	3	< 64.9	160	2,120	472	192	132	3	N/A
GNN (MLST'24) [7]	16	3	< 70.8	180	5,362	1,388	594	52	3	N/A
GNN (MLST'24) [7]	32	3	< 75.8	205	2,120	1,162	761	12	3	N/A
DS M (MLST'25) [30]	8	3	65.1	110	548	130	49	4	3	N/A
DS L (MLST'25) [30]	8	3	66.6	135	2,458	337	140	4	3	N/A
JEDI-linear	8	3	66.5	79	0	136	73	0	1	302.8
JEDI-linear	16	3	73.6	75	0	136	71	0	1	305.7
JEDI-linear	32	3	79.0	80	0	136	79	0	1	299.4
JEDI-linear	**64**	3	81.8	78	0	164	93	0	1	307.0
JEDI-linear	**128**	3	81.6	138	0	296	163	0	1	203.1
GNN (AICAS'22) [16]	30	16	78.7	3000	7417	810	205	924	600	N/A
GNN (FPL'22) [17]	30	16	78.7	1910	11504	1158	246	1392	400	N/A
GNN (FPL'22) [17]	50	16	80.4	10660	12,284	1515	533	1607	650	N/A
GNN J4 (TECS'24) [4]	30	16	78.4	290	8,776	865	138	37	30	N/A
GNN J5 (TECS'24) [4]	30	16	79.[illegible]	905	9,833	911	158	37	150	N/A
GNN U4 (TECS'24) [4]	50	16	80.9	650	8,945	855	201	25	100	N/A
GNN U5 (TECS'24) [4]	50	16	81.2	905	8,986	815	189	37	150	N/A
JEDI-linear	8	16	73.8	67	0	72	40	0	1	311.3
JEDI-linear	16	16	78.3	72	0	99	50	0	1	307.0
JEDI-linear	32	16	81.4	79	0	147	71	0	1	304.7
JEDI-linear	64	16	82.4	93	0	192	92	0	1	268.1
JEDI-linear	**128**	16	82.1	110	0	243	111	0	1	237.4

3.7~ 11.5x lower latency

Best Latency

Evaluation: Latency & Resources

Model	Particles	Features	Acc. (%)	Latn. (ns)	DSP	LUT (k)	FF (k)	BRAM	II (clk)	F_{max} (MHz)
DS (MLST'24) [7]	8	3	< 64.0	95	626	386	121	4	3	N/A
DS (MLST'24) [7]	16	3	< 69.4	115	555	747	239	4	3	N/A
DS (MLST'24) [7]	32	3	< 75.9	130	434	903	359	4	2	N/A
GNN (MLST'24) [7]	8	3	< 64.9	160	2,120	472	192	132	3	N/A
GNN (MLST'24) [7]	16	3	< 70.8	18[illegible]	5,362	1,388	594	52	3	N/A
GNN (MLST'24) [7]	32	3	< 75.8	[illegible]05	2,120	1,162	761	12	3	N/A
DS M (MLST'25) [30]	8	3	65.1	110	548	130	49	4	3	N/A
DS L (MLST'25) [30]	8	3	66.6	135	2,458	337	140	4	3	N/A
JEDI-linear	8	3	66.5	[illegible]	0	136	73	0	1	302.8
JEDI-linear	16	3	73.6	7[illegible]	0	136	71	0	1	305.7
JEDI-linear	32	3	79.0	80	0	136	79	0	1	299.4
JEDI-linear	**64**	3	81.8	78	0	164	93	0	1	307.0
JEDI-linear	**128**	3	81.6	138	0	296	163	0	1	203.1
GNN (AICAS'22) [16]	30	16	78.7	3000	7417	810	205	924	600	N/A
GNN (FPL'22) [17]	30	16	78.7	1910	11504	1158	246	1392	400	N/A
GNN (FPL'22) [17]	50	16	80.4	10660	12,284	1515	533	1607	650	N/A
GNN J4 (TECS'24) [4]	30	16	78.4	290	8,776	865	138	37	30	N/A
GNN J5 (TECS'24) [4]	30	16	79.9	90[illegible]	9,833	911	158	37	150	N/A
GNN U4 (TECS'24) [4]	50	16	80.9	[illegible]0	8,945	855	201	25	100	N/A
GNN U5 (TECS'24) [4]	50	16	81.2	905	8,986	815	189	37	150	N/A
JEDI-linear	8	16	73.8	67	0	72	40	0	1	311.3
JEDI-linear	16	16	78.3	72	0	99	50	0	1	307.0
JEDI-linear	32	16	81.4	[illegible]	0	147	71	0	1	304.7
JEDI-linear	64	16	82.4	93	0	192	92	0	1	268.1
JEDI-linear	**128**	16	82.1	110	0	243	111	0	1	237.4

0 Coarse-grained DSPs

Evaluation: Latency & Resources

Model	Particles	Features	Acc. (%)	Latn. (ns)	DSP	LUT (k)	FF (k)	BRAM	II (clk)	F_{max} (MHz)
DS (MLST'24) [7]	8	3	< 64.0	95	626	386	121	4	3	N/A
DS (MLST'24) [7]	16	3	< 69.4	115	555	747	239	4	3	N/A
DS (MLST'24) [7]	32	3	< 75.9	130	434	903	359	4	2	N/A
GNN (MLST'24) [7]	8	3	< 64.9	160	2,120	472	192	132	3	N/A
GNN (MLST'24) [7]	16	3	< 70.8	180	5,362	1,388	594	52	3	N/A
GNN (MLST'24) [7]	32	3	< 75.8	205	2,120	1,162	761	12	3	N/A
DS M (MLST'25) [30]	8	3	65.1	110	548	130	49	4	3	N/A
DS L (MLST'25) [30]	8	3	66.6	135	2,458	337	140	4	3	N/A
JEDI-linear	8	3	66.5	79	0	136	73	0	1	302.8
JEDI-linear	16	3	73.6	75	0	136	71	0	1	305.7
JEDI-linear	32	3	79.0	80	0	136	79	0	1	299.4
JEDI-linear	**64**	3	81.8	78	0	164	93	0	1	307.0
JEDI-linear	**128**	3	81.6	138	0	296	163	0	1	203.1
GNN (AICAS'22) [16]	30	16	78.7	3000	7417	810	205	924	600	N/A
GNN (FPL'22) [17]	30	16	78.7	1910	11504	1158	246	1392	400	N/A
GNN (FPL'22) [17]	50	16	80.4	10660	12,284	1515	533	1607	650	N/A
GNN J4 (TECS'24) [4]	30	16	78.4	290	8,776	865	138	37	30	N/A
GNN J5 (TECS'24) [4]	30	16	79.9	905	9,8[illegible]	911	158	37	150	N/A
GNN U4 (TECS'24) [4]	50	16	80.9	[illegible]	[illegible]945	855	201	25	100	N/A
GNN U5 (TECS'24) [4]	50	16	81.2	[illegible]	8,986	815	189	37	150	N/A
JEDI-linear	8	16	73.8	[illegible]	0	72	40	0	1	311.3
JEDI-linear	16	16	78.3	[illegible]	0	99	50	0	1	307.0
JEDI-linear	32	16	81.4	79	[illegible]	147	71	0	1	304.7
JEDI-linear	64	16	82.4	93	0	192	92	0	1	268.1
JEDI-linear	**128**	16	82.1	110	0	243	111	0	1	237.4

6.2x fewer LUTs

Fewest fine-grained LUTs

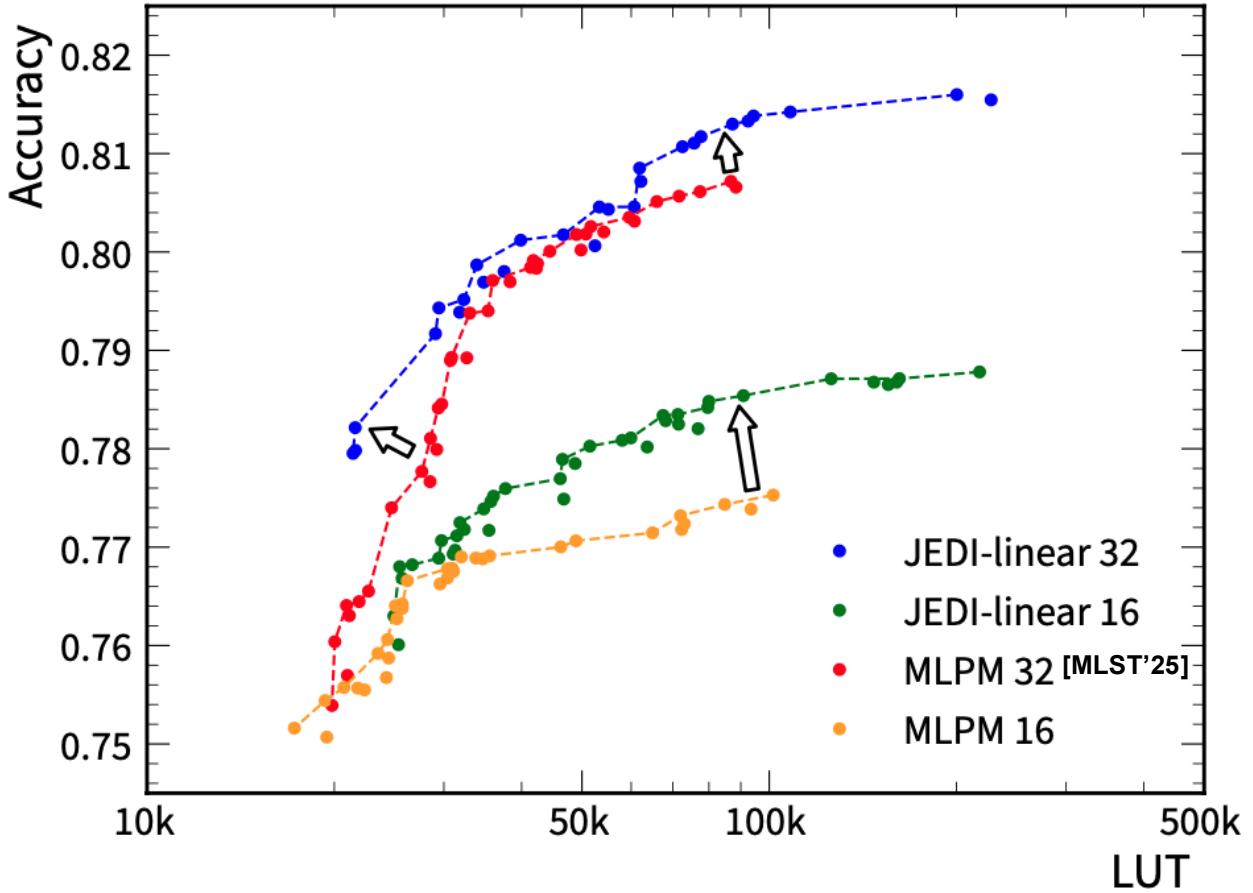
Evaluation: Latency & Resources

Model	Particles	Features	Acc. (%)	Latn. (ns)	DSP	LUT (k)	FF (k)	BRAM	II (clk)	F_{max} (MHz)
DS (MLST'24) [7]	8	3	< 64.0	95	626	386	121	4	3	N/A
DS (MLST'24) [7]	16	3	< 69.4	115	555	747	239	4	3	N/A
DS (MLST'24) [7]	32	3	< 75.9	130	434	903	359	4	2	N/A
GNN (MLST'24) [7]	8	3	< 64.9	160	2,120	472	192	132	3	N/A
GNN (MLST'24) [7]	16	3	< 70.8	180	5,362	1,388	594	52	3	N/A
GNN (MLST'24) [7]	32	3	< 75.8	205	2,120	1,162	761	[illegible]	3	N/A
DS M (MLST'25) [30]	8	3	65.1	110	548	130	49	4	3	N/A
DS L (MLST'25) [30]	8	3	66.6	135	2,458	337	140	4	3	N/A
JEDI-linear	8	3	66.5	79	0	136	73	[illegible]	1	302.8
JEDI-linear	16	3	73.6	75	0	136	71	[illegible]	1	305.7
JEDI-linear	32	3	79.0	80	0	136	79	0	1	299.4
JEDI-linear	**64**	3	81.8	78	0	164	93	0	1	307.0
JEDI-linear	**128**	3	81.6	138	0	296	163	0	1	203.1
GNN (AICAS'22) [16]	30	16	78.7	3000	7417	810	205	924	600	N/A
GNN (FPL'22) [17]	30	16	78.7	1910	11504	1158	246	1392	400	N/A
GNN (FPL'22) [17]	50	16	80.4	10660	12,284	1515	533	1607	650	N/A
GNN J4 (TECS'24) [4]	30	16	78.4	290	8,776	865	138	37	30	N/A
GNN J5 (TECS'24) [4]	30	16	79.9	905	9,833	911	158	37	150	N/A
GNN U4 (TECS'24) [4]	50	16	80.9	650	8,945	855	201	[illegible]	100	N/A
GNN U5 (TECS'24) [4]	50	16	81.2	905	8,986	815	189	37	150	N/A
JEDI-linear	8	16	73.8	67	0	72	40	0	1	311.3
JEDI-linear	16	16	78.3	72	0	99	50	[illegible]	1	307.0
JEDI-linear	32	16	81.4	79	0	147	71	0	1	304.7
JEDI-linear	64	16	82.4	93	0	192	92	0	1	268.1
JEDI-linear	**128**	16	82.1	110	0	243	111	0	1	237.4

Lowest iteration interval (II): time between processing successive jets

Comparison with Non-GNN Models (Non-Permutation-Invariant)

- New Pareto Frontier
- Better accuracy with lower LUTs
- Not just a single point
- A full design space better than prior work



[**MLST'25**] C. Sun, et al., "Fast Jet Tagging with MLP-Mixers on FPGAs." *Machine Learning: Science and Technology*, 2025.

Future Work

- "linearization" algorithm-hardware codesign strategy
  - efficient and low latency transformer on FPGAs
  - demanding applications beyond FPGAs
- High performance trustworthy computing for jet tagging
  - JEDI-linear + Bayesian neural network
- Other scientific domains requiring low latency
  - aerospace
  - healthcare
- Design automation with metaprogramming
  - use Python for static and dynamic optimization of C++

Summary and Impact

- A1 Algorithmic Optimization:
  - linearization with Global Context Vector strategy
  - reduces complexity from **$O(N^2)$ to $O(N)$**
- A2 Hardware Optimization
  - fine-Grained Mixed-Precision Quantization
  - Distributed Arithmetic (DA)
  - end-to-end automation with RTL generation
- A3 Evaluation: compared to state-of-the-art
  - open source: https://github.com/calad0i/JEDI-linear
  - 3.7x to 11.5x lower latency, up to 6.3x lower LUT
  - less than 60 ns latency

Artifacts Evaluation Results:
Available,
Evaluated Functional,
Reusable,
Results Replicated

My webpage

- next-gen design for CERN: enabling new scientific discoveries
- public code base: novel optimizations for applications beyond jet tagging