

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

Zhiqiang Que, Chang Sun

Sudarshan Paramesvaran, Emyr Clement, Katerina Karakoulaki, Christopher E. Brown

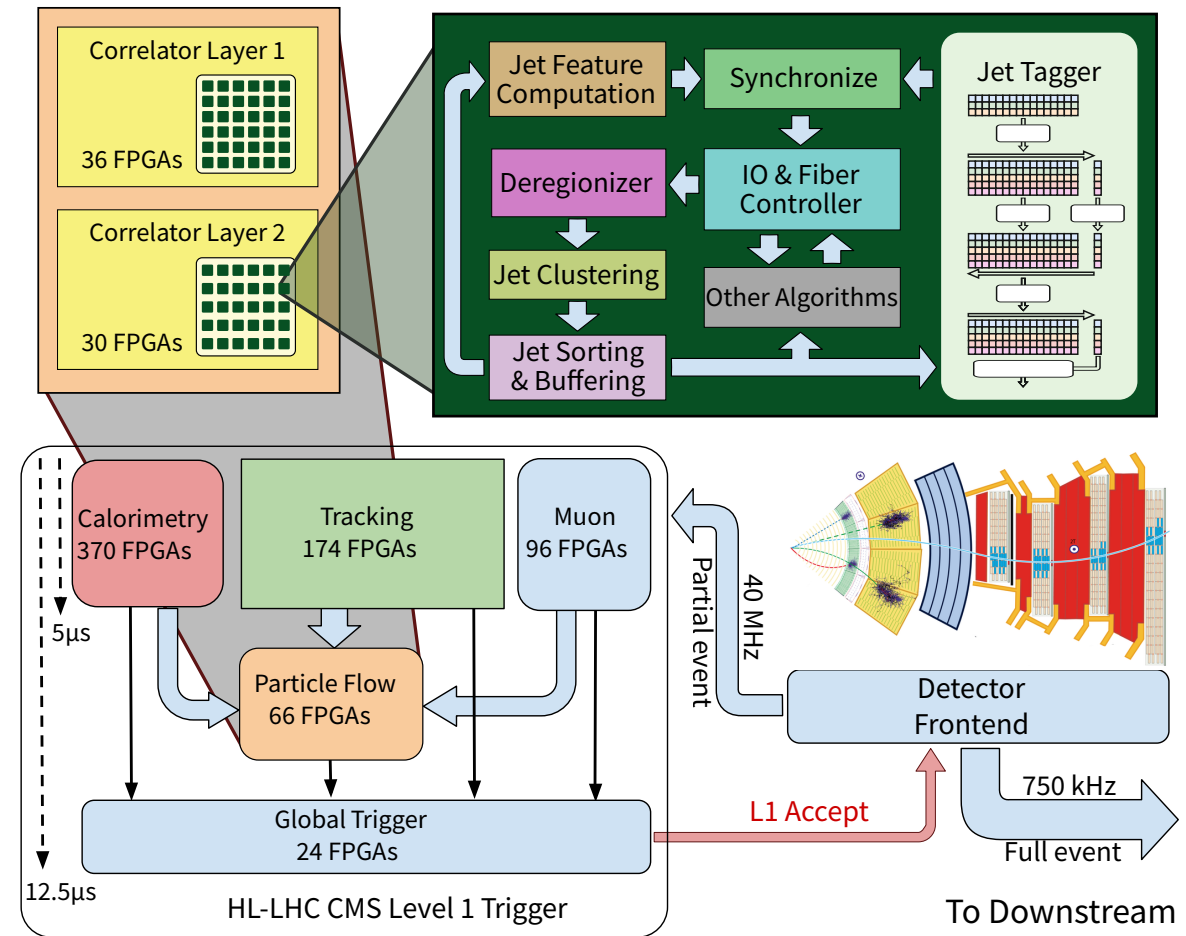
Lauri A O Laatu, Arianna Cox, Alexander Tapper, Wayne Luk, Maria Spiropulu

[arxiv:2508.15468](https://arxiv.org/abs/2508.15468)

Task Description

Jet Tagging on L1T

- We target CMS L1T, Correlator L2
- Xilinx UltraScale+ 13P
- Maximum $<1\text{SLR}$, $II=1$
- Expect other logics on same chip

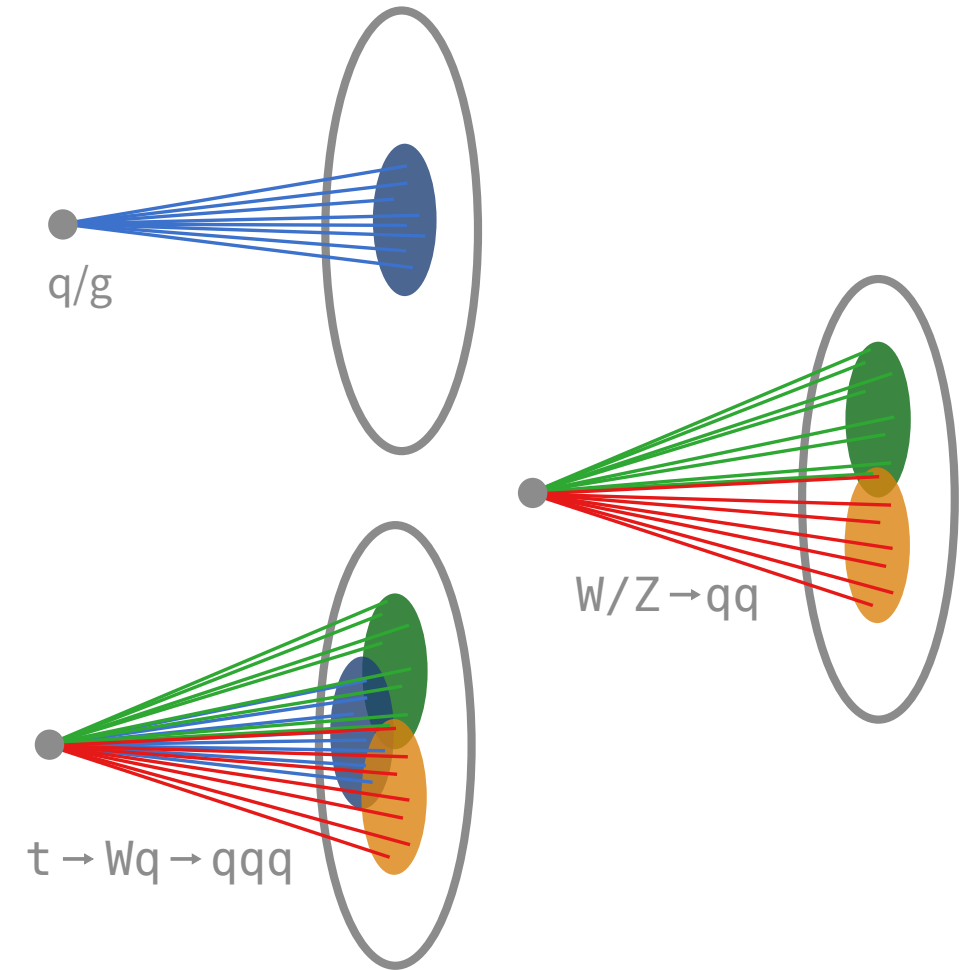


Task Description

Dataset Used

The [hls4ml jet tagging dataset](#) is used

- Objective: classify jets into 5 classes
 - g / q / W / Z / t - jets
- Input feature
- N (16/32/64/128) particles of the highest p_T
 - Each with 16 features $\rightarrow N \times 16$ array
 - Only p_T, η, ϕ , with $p_T \geq 2$ GeV cut
- p_T sorted
 - Descending order



Problem Statement

Problem:

GNNs great for jet tagging, but too slow/expensive for L1T.

Proposed method:

Linear-complexity interactions + hardware-aware co-design.

Result:

sub-100 ns latency, $ll=1$, 0 DSPs, SOTA accuracy among existing designs.

Jet tagging need to be fast, deterministic, and resource-lean.

We use the [JEDI-net](#) [EPJC'20] as the starting point.

However, JEDI-net is **dense** (i.e., compute all pairwise edges) $\rightarrow \mathcal{O}(N^2)$ cost.

- Prior FPGA GNNs [[TECS'24](#), [MLST'24](#)] hit latency/II/resource walls (e.g., DSPs > 8.7k)

JEDI-net

JEDI-net is a fully-connected interaction network for jet tagging.

The core node-edge-node interaction part:

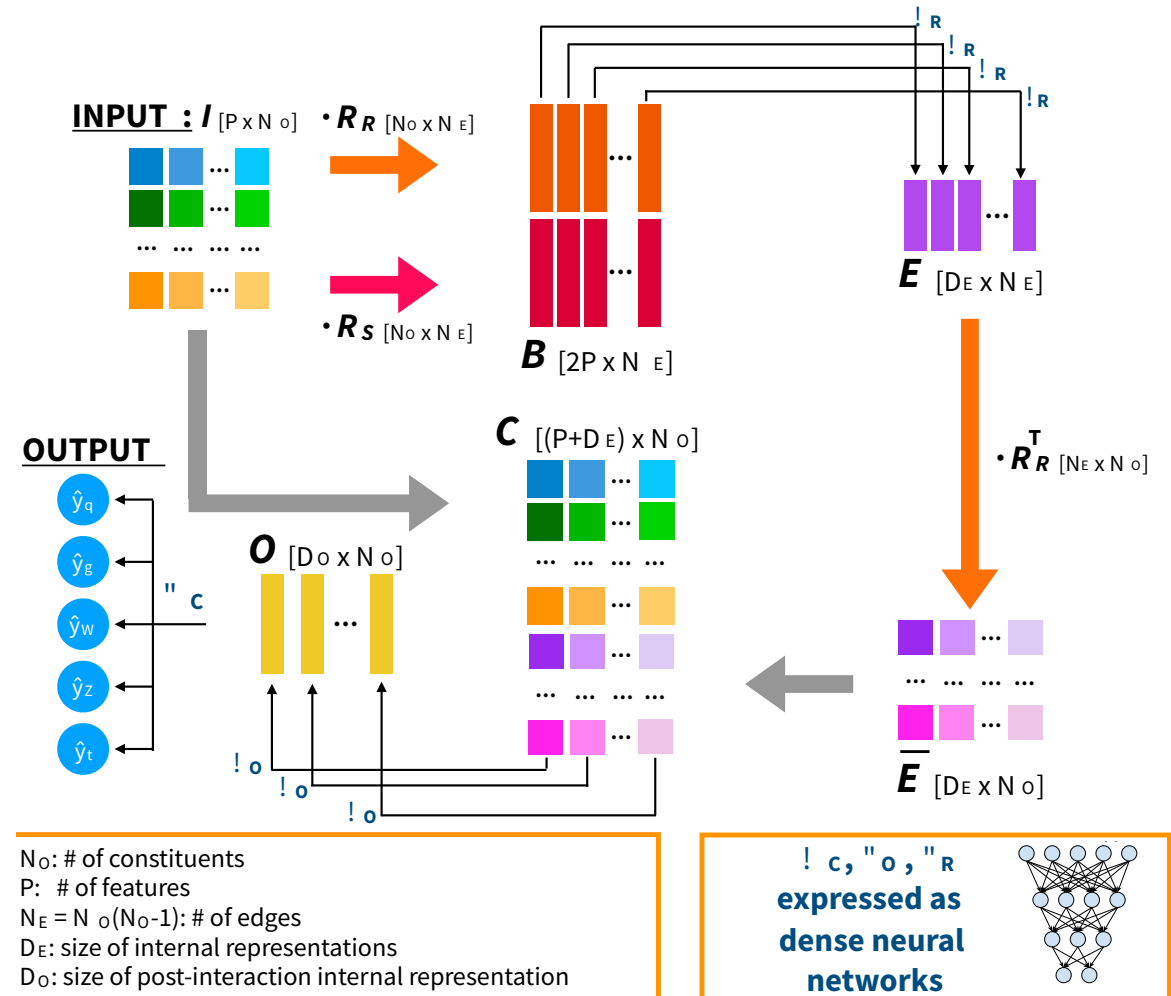
$$B_{ij} = I_i || I_j \in \mathbb{R}^{2P}$$

$$E_{ij} = f_R(B_{ij}) \in \mathbb{R}^{D_E}$$

$$\bar{E}_i = \sum_{j \neq i} E_{ij} \in \mathbb{R}^{D_E}$$

Hence,

$$\bar{E}_i = \sum_{j \neq i} f_R(I_i || I_j)$$



Linearized Interaction

Require each edge network to be an affine transformation:

$$f_R(B_{ij})_l = W (I_i \| I_j) + C = W_1 I_i + W_2 I_j + C$$

Then, the interaction becomes:

$$\begin{aligned}\bar{E}_i &= \sum_j f_R(I_i \| I_j) \\ &= \sum_{j \neq i} (W_1 I_i + W_2 I_j + C) \\ &= W_2 \sum_j (I_j) - I_i + (N - 1) (W_1 I_i + C)\end{aligned}$$

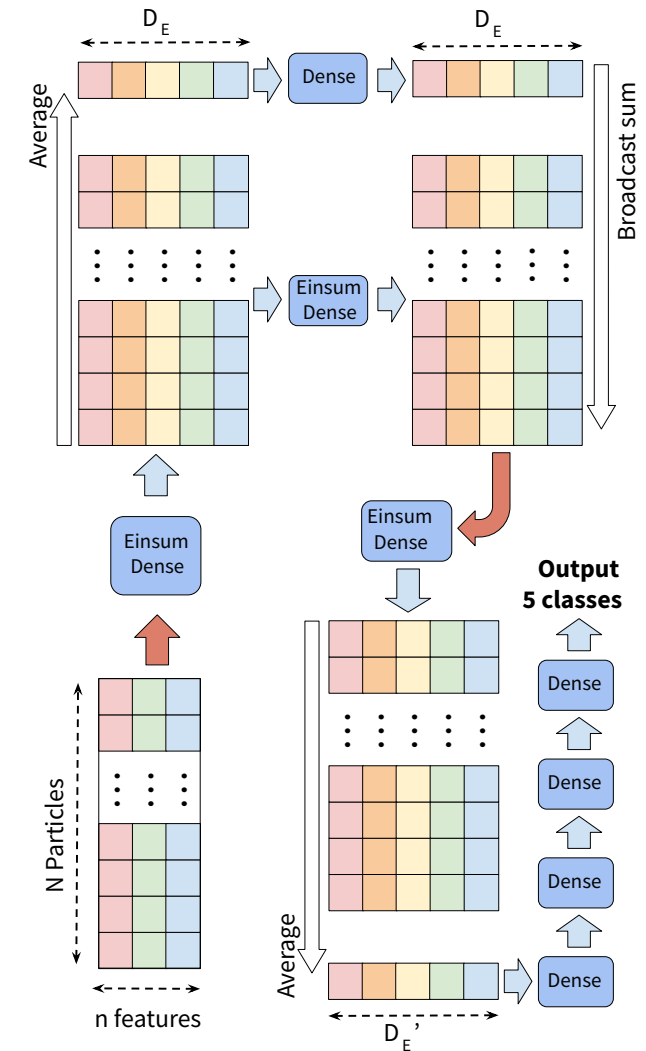
Linearized Interaction

Further, we normalize everything by N , and discard $\frac{1}{N}$ terms for large N :

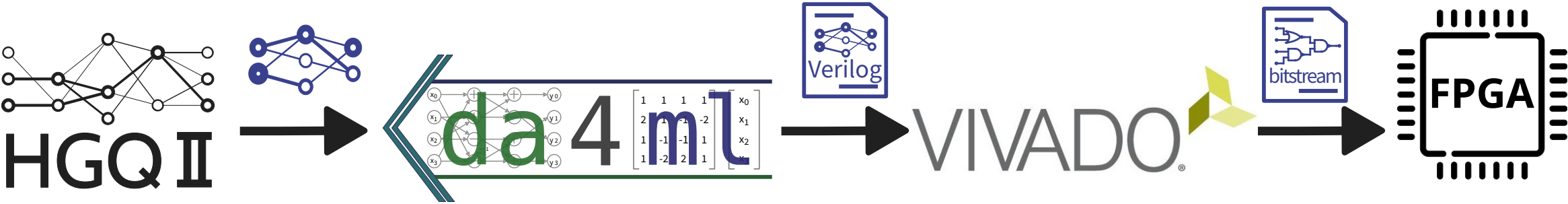
- The $\frac{I_i}{N}$ terms can also be absorbed into W_1 by slight rescaling and adding diagonal matrix

$$\begin{aligned}\bar{E}'_i &= \frac{1}{N} \bar{E}_i \\ &= W_2 \frac{1}{N} \sum_j (I_j) - \frac{I_i}{N} + \frac{N-1}{N} (W_1 I_i + C) \\ &\approx W_2 \frac{1}{N} \sum_j (I_j) + W_1 I_i + C\end{aligned}$$

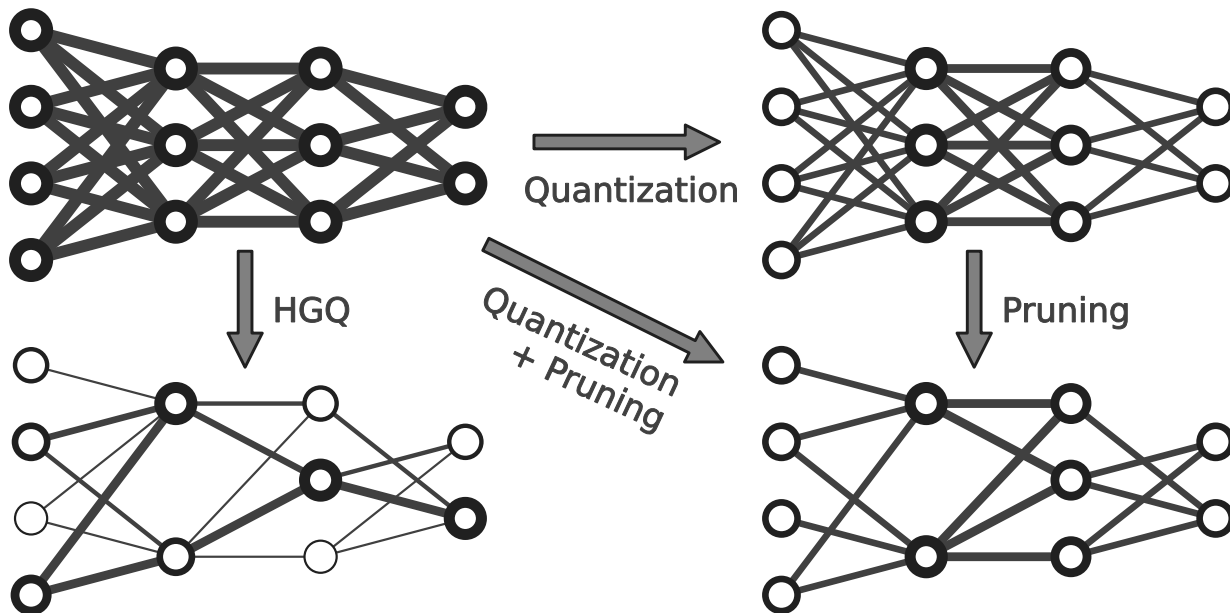
- Can be viewed as a **restricted MLP-Mixer** in the particle dimension



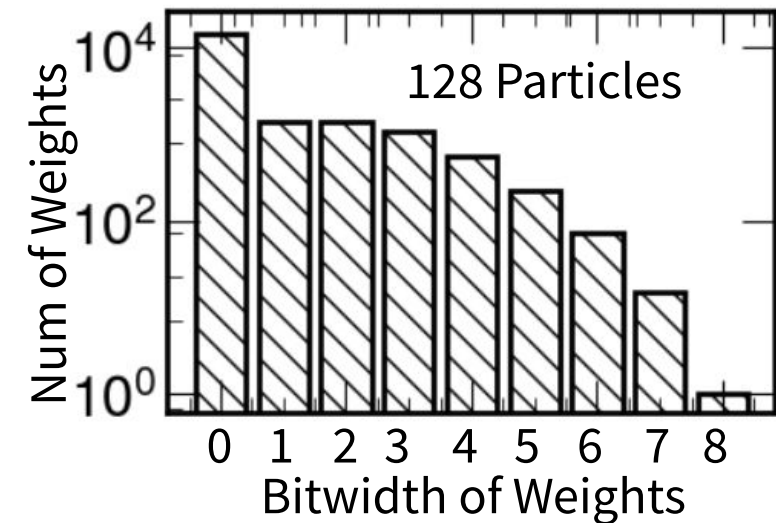
da4ml standalone workflow



High Granularity Quantization



- Fully Heterogeneous precision everywhere



More details in [[code](#), [doc](#), [paper](#)].

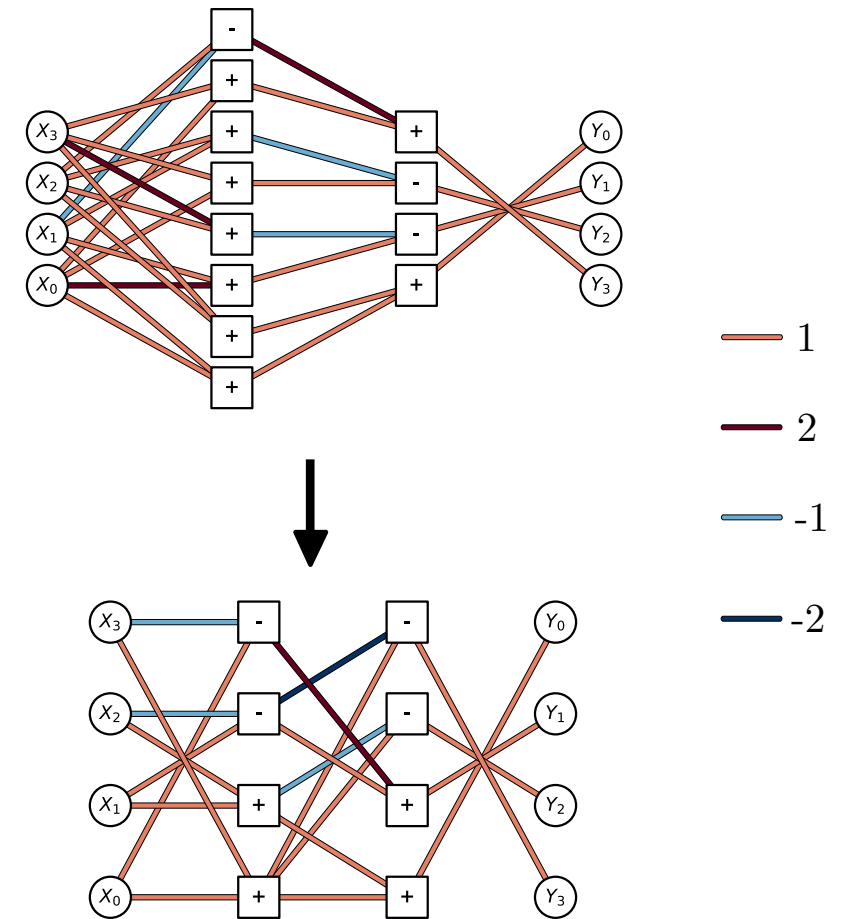
Distributed Arithmetic

Talk on Tuesday

Implementing NN and adder Graph

- No explicit multiplication
- Everything as LUTs and fast carriers
- **Less LUTs and 0 DSPs for free**

More details in [[code](#), [doc](#), [paper](#)].



Experimental Condition

- Synthesis with Vivado2025.1
- `xcvu13p-f1ga2577-2-e` (UltraScale+ 13P)
- Accuracy reported from **verilog simulation**
- All other results reported are after **OOB routing**
- $P(\text{keras-verilator-mismatch}) \sim \mathcal{O}(0.0001)$
 - With small differences, likely floating point rounding error in keras
 - Negligible accuracy impact
- Fully pipelined

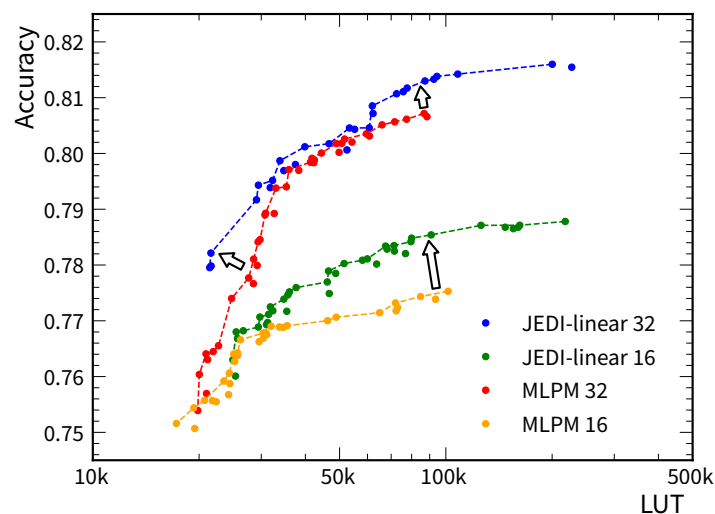
Results - Permutation Invariant Models

- Order or magnitude lower latency and resource
- All together with higher accuracy
- Supports up to 128 particles

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) [27]	8	3	65.1	110	548	130	49	4	3	N/A
DS L (MLST'25) [27]	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) [13]	30	16	78.7	3000	7417	810	205	924	600	N/A
GNN (FPL'22) [14]	30	16	78.7	1910	11504	1158	246	1392	400	N/A
GNN (FPL'22) [14]	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	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

Results - Non-Permutation Invariant Models

- Slightly better Pareto frontier



Model	Particles	Features	Acc. (%)	Latn. (ns)	DSP	LUT (k)	FF (k)	BRAM	II (clk)	F _{max} (MHz)
MLPM (MLST'25) [26]	16	3	71.7	68	0	75	17	0	1	205.5
MLPM (MLST'25) [26]	32	3	78.0	62	0	63	15	0	1	211.0
MLPM (MLST'25) [26]	64	3	79.7	72	0	159	36	0	1	209.4
MLPM (MLST'25) [26]	128	3	79.8	72	0	83	21	0	1	208.7
JEDI-linear	16	3	71.9	54	0	44	22	0	1	354.4
JEDI-linear	32	3	78.0	63	0	45	26	0	1	300.8
JEDI-linear	64	3	80.9	61	0	71	38	0	1	328.4
JEDI-linear	128	3	80.9	82	0	98	48	0	1	257.6
MLPM (MLST'25) [26]	16	16	77.5	71	0	102	24	0	1	210.9
MLPM (MLST'25) [26]	32	16	80.7	65	0	87	22	0	1	215.8
MLPM (MLST'25) [26]	64	16	81.6	65	0	126	32	0	1	213.9
MLPM (MLST'25) [26]	128	16	81.3	77	0	151	42	0	1	207.8
JEDI-linear	16	16	77.6	52	0	38	20	0	1	381.7
JEDI-linear	32	16	80.9	60	0	62	33	0	1	347.7
JEDI-linear	64	16	81.8	67	0	84	47	0	1	327.8
JEDI-linear	128	16	81.7	77	0	93	46	0	1	285.7