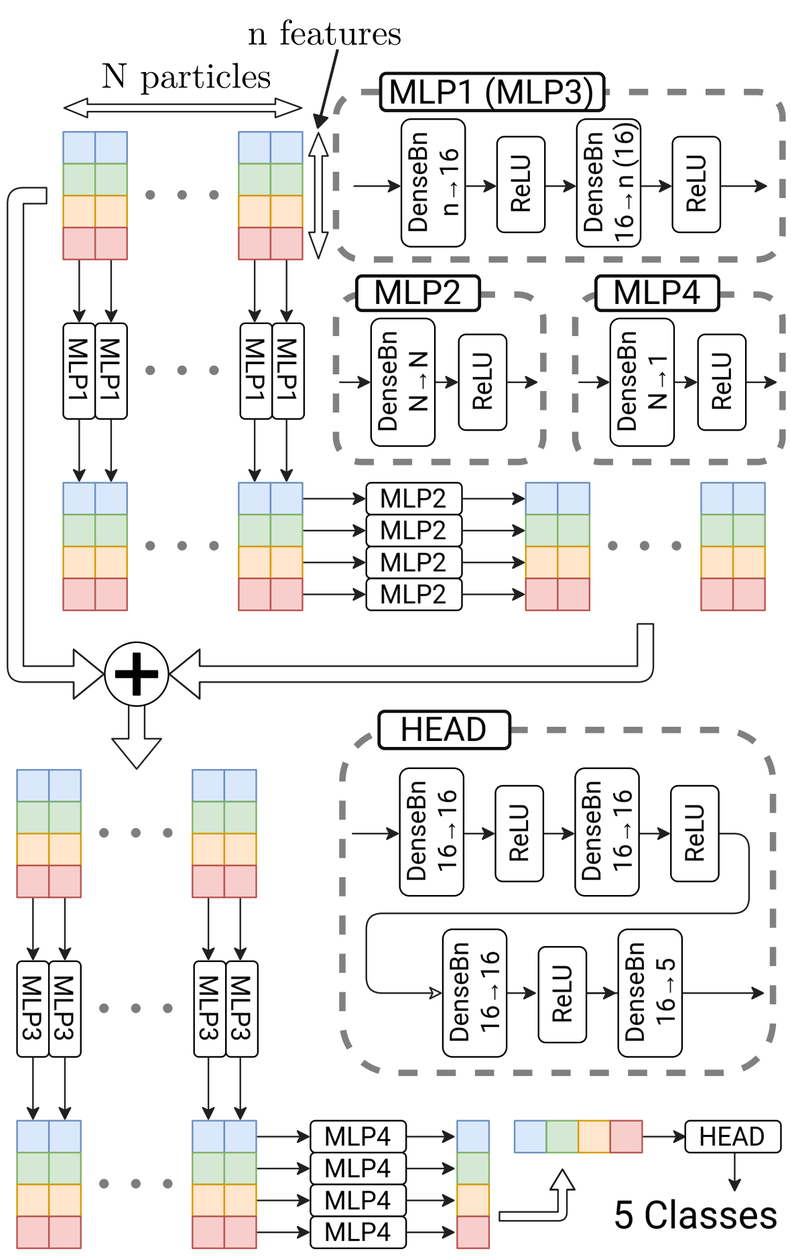
Fast Jet Tagging with MLP-Mixer

Chang Sun*,[1], Jennifer Ngadiuba[2], Maria Spiropulu[1]

*: chsun@cern.ch 1: California Institute of Technology 2: Fermi National Accelerator Laboratory

We demonstrate the use of the MLP-Mixer architecture for fast jet classification in high-energy physics.

The MLP-Mixer model is trained with the hls4ml LHC Jet dataset (150 partciles), which is meant to represent the type of L1 trigger objects that we may expect in future High-Luminosity LHC experiments. We show that our MLP-Mixer model could achieve JEDI-net 50 particle performance with ~1/8 of LUTs, no DSP, 100x throughput, and 1/10 latency simutaeously when trained with HGQ and synthesized with hls4ml + da4ml + Vitis 2023.2. This is the first time MLP Mixer is shown to be deployable for L1 Triggers at LHC.

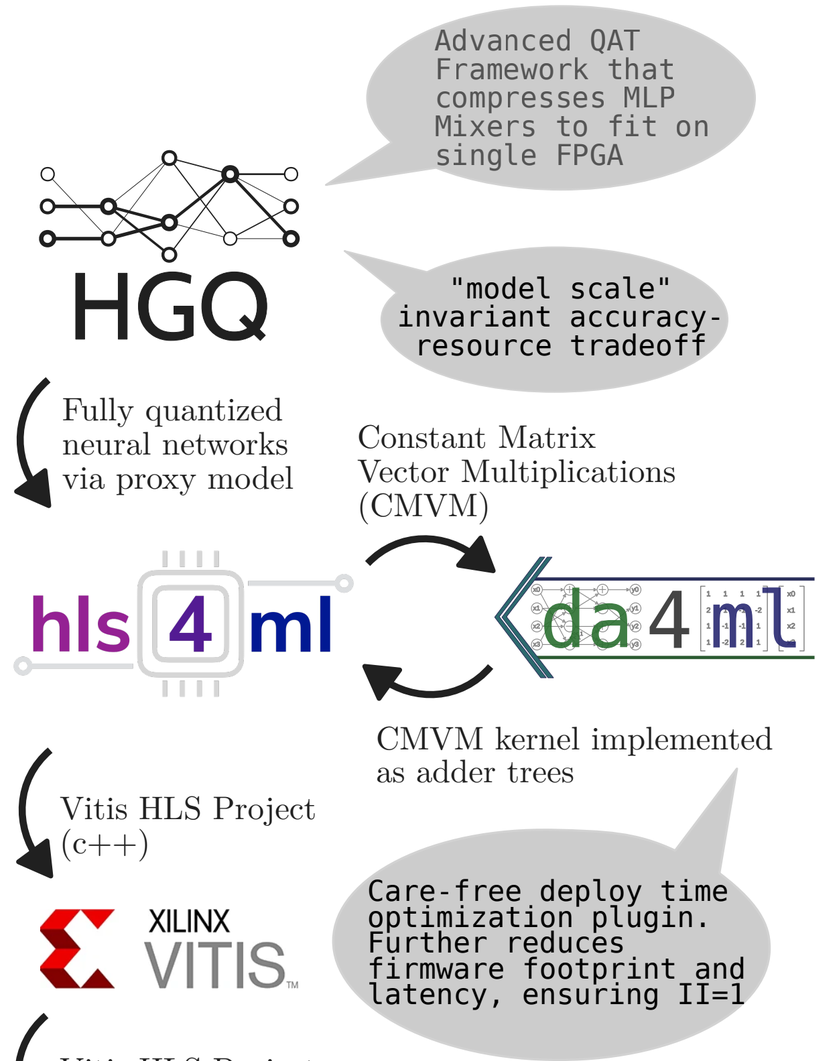


HGQ

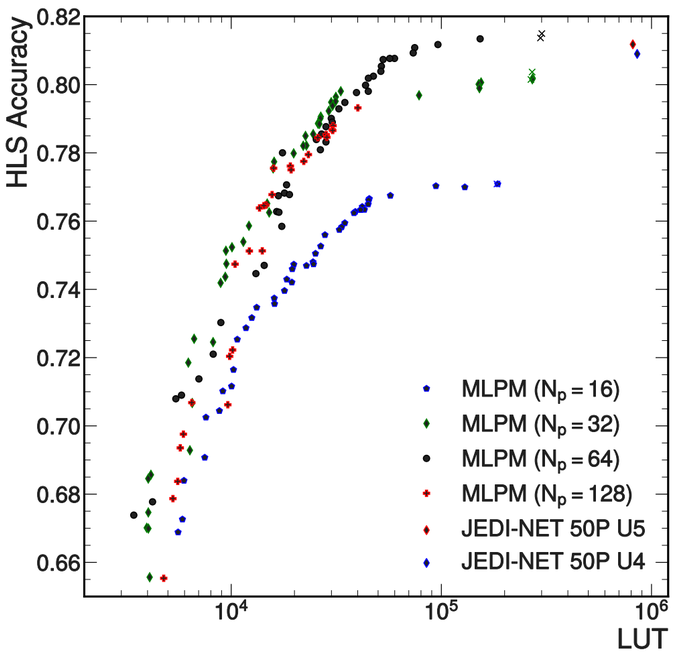
High Granularity Quantization is an advanced QAT method & framework that makes these MLP Mixers fit on a single FPGA.

HGQ quantizes weights and activations individually and optimizes their bitwidths with gradients automatically (pruning is included as bw→0). This reduces the firmware footprint by more than 10x during training with minimal impact on accuracy.

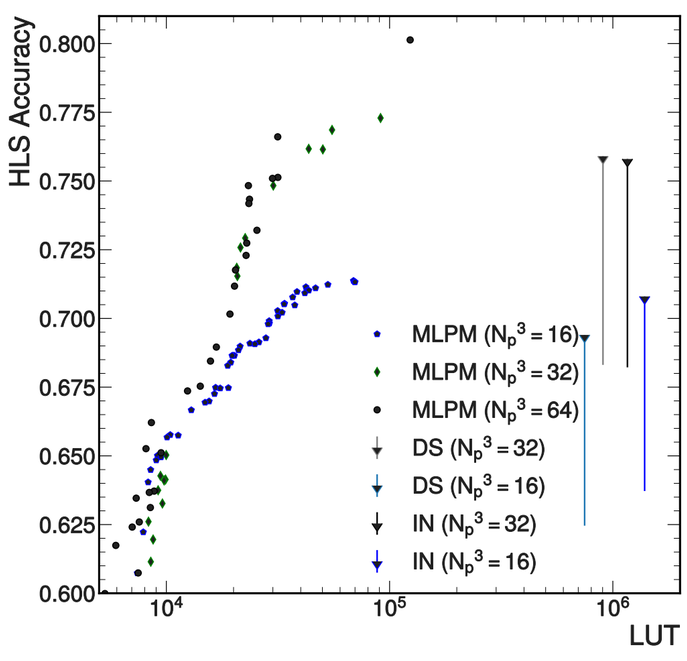
All models with the same size are trained with a single run, covering the entire Pareto Front in ~3 hours.



16 feaures



3 feaures



×: models with timing violation

16 feaures

Model	Accuracy (%)	Latency (cc)	DSP	LUT (k)	FF (k)	II (cc)
JEDI-NET U4 (N_p=50) [7]	80.90	130 (650 ns)	8,945	855	201	110
JEDI-NET U5 (N_p=50) [7]	81.18	181 (905 ns)	8,986	815	189	150
JEDI-NET J4 (N_p=30) [7]	78.41	58 (290 ns)	8,776	865	138	110
JEDI-NET J5 (N_p=30) [7]	79.85	181 (905 ns)	9,833	911	158	150
MLPM-1 (N_p=64)	81.17	14 (70 ns)	0	104	36	1
MLPM-2 (N_p=64)	81.34	15 (75 ns)	0	162	46	1
MLPM-3 (N_p=32)	78.50	9 (45 ns)	0	23	6	1
MLPM-4 (N_p=32)	79.81	13 (65 ns)	0	33	10	1

3 feaures

Model	Accuracy (%)	Latency (cc)	DSP	LUT (k)	FF (k)	II (cc)
DS (N_p^3=32) [8]	68.3 ~ 75.9	26 (130 ns)	434	903	359	2
DS (N_p^3=16) [8]	62.5 ~ 69.4	23 (115 ns)	555	747	239	3
IN (N_p^3=32) [8]	68.2 ~ 75.8	41 (205 ns)	2,120	1,162	761	3
IN (N_p^3=16) [8]	63.7 ~ 70.8	36 (180 ns)	5,362	1,388	594	3
MLPM'-1 (N_p^3=64)	80.13	15 (75 ns)	0	124	30	1
MLPM'-2 (N_p^3=64)	76.60	12 (60 ns)	0	32	12	1
MLPM'-3 (N_p^3=32)	76.17	13 (65 ns)	0	43	10	1
MLPM'-4 (N_p^3=16)	70.92	13 (65 ns)	0	12	10	1

[7]: https://arxiv.org/abs/2209.14065
[8]: https://arxiv.org/abs/2402.01876

da4ml

Distributed Arithmetic for ML (da4ml) is a framework that transforms Constant Matrix Vector Multiplication (CMVM) with distributed arithmetic, which produces an adder tree that is equivalent to the original CMVM. It currently serves as plugin for hls4ml.

By performing aggressive common subexpression eliminations, da4ml may reduce the LUT usage by 1/3 ~ 1/2 while removing all DSP usage without affect the output bit accuracy.

da4ml is required for running the MLP Mixer with II=1. Without explicitly performing DA, II remains unstable with Vitis HLS and some models fail to synth.

#77