ÖAW Austrian Academy of Sciences

MARIETTA BLAU INSTITUTE FOR PARTICLE PHYSICS

SCIES 4 FREE

TU WIEN

Do BitNet Gains Survive Synthesis?

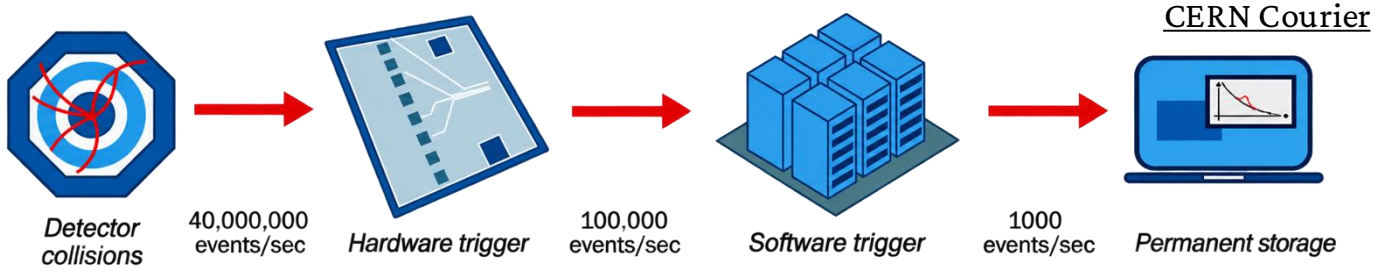
An Implementation-Aware Benchmark of Low-Precision Jet Classification

Dorian Sloot

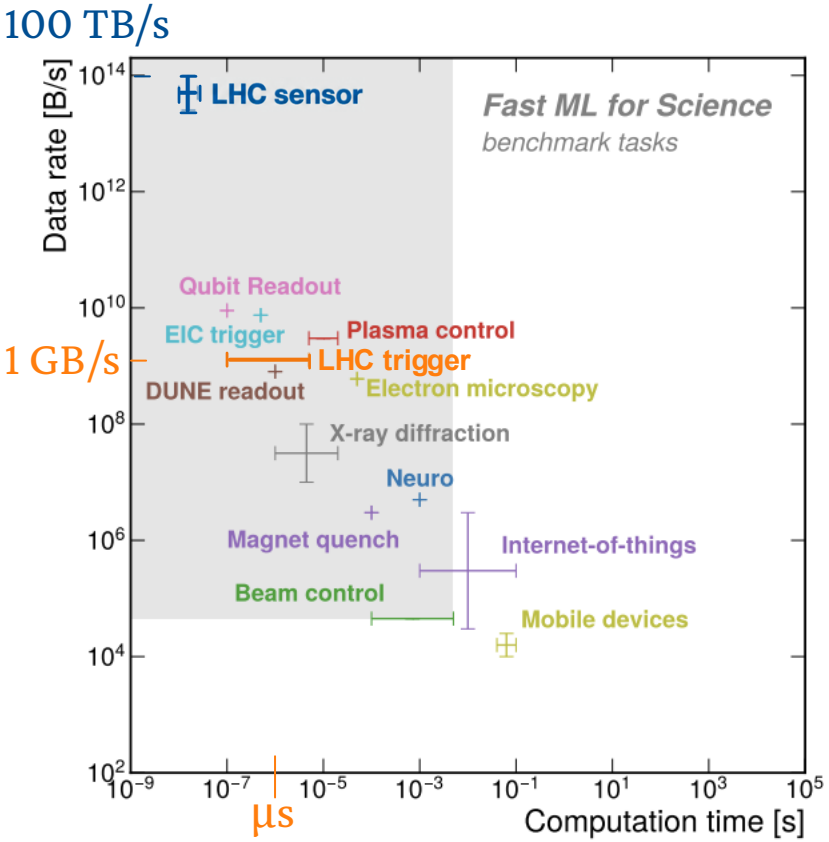
Vienna, 17.09.2026

Fast Machine Learning for HEP

- Machine learning is becoming an increasingly important tool across high-energy physics.
- This includes LHC detector readout and trigger systems, which operate in an extreme regime requiring high physics performance under strict constraints on latency, throughput and hardware resources.



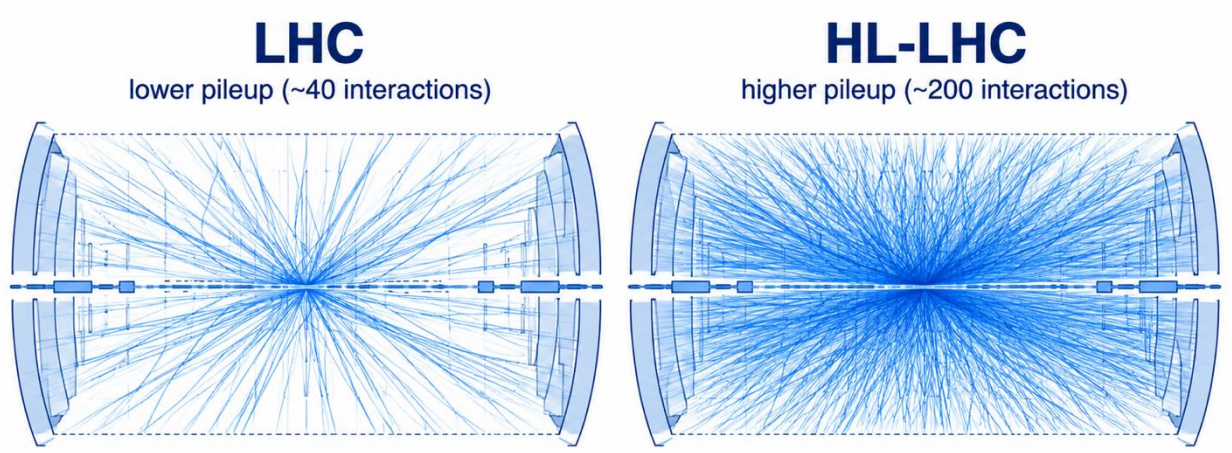
CERN Courier

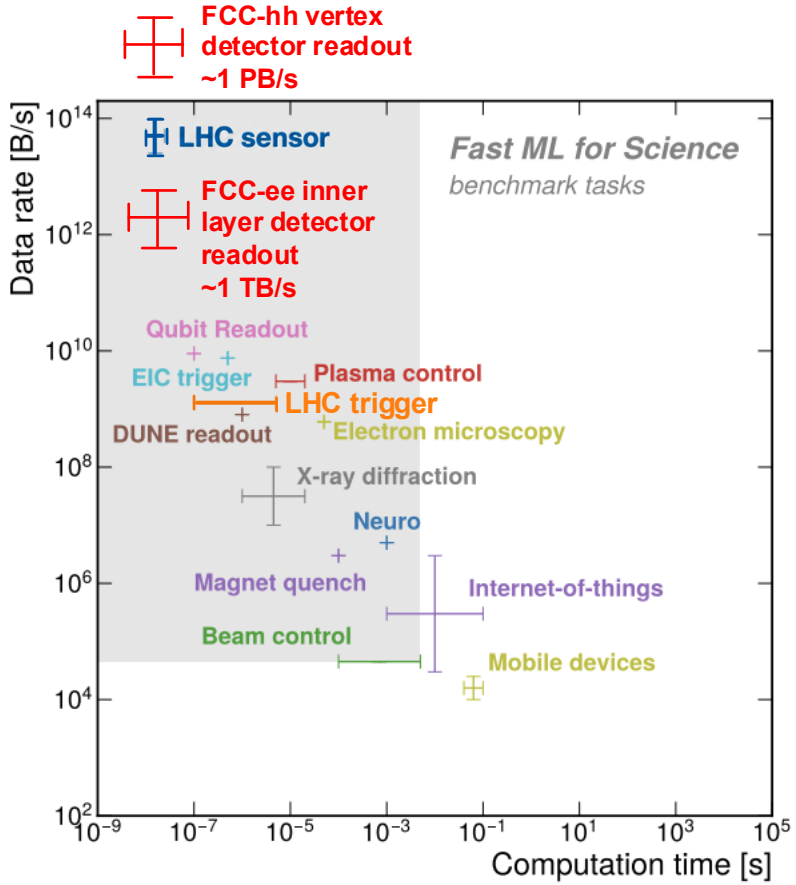


arXiv:2406.19522

Fast ML for the Future of HEP

- Low-precision ML enables more sophisticated models within strict real-time hardware constraints.
- The aim is to reduce computational cost without sacrificing, and ideally improving, physics performance.
- This becomes increasingly important as detector upgrades and future collider environments increase event complexity and data-processing demands.

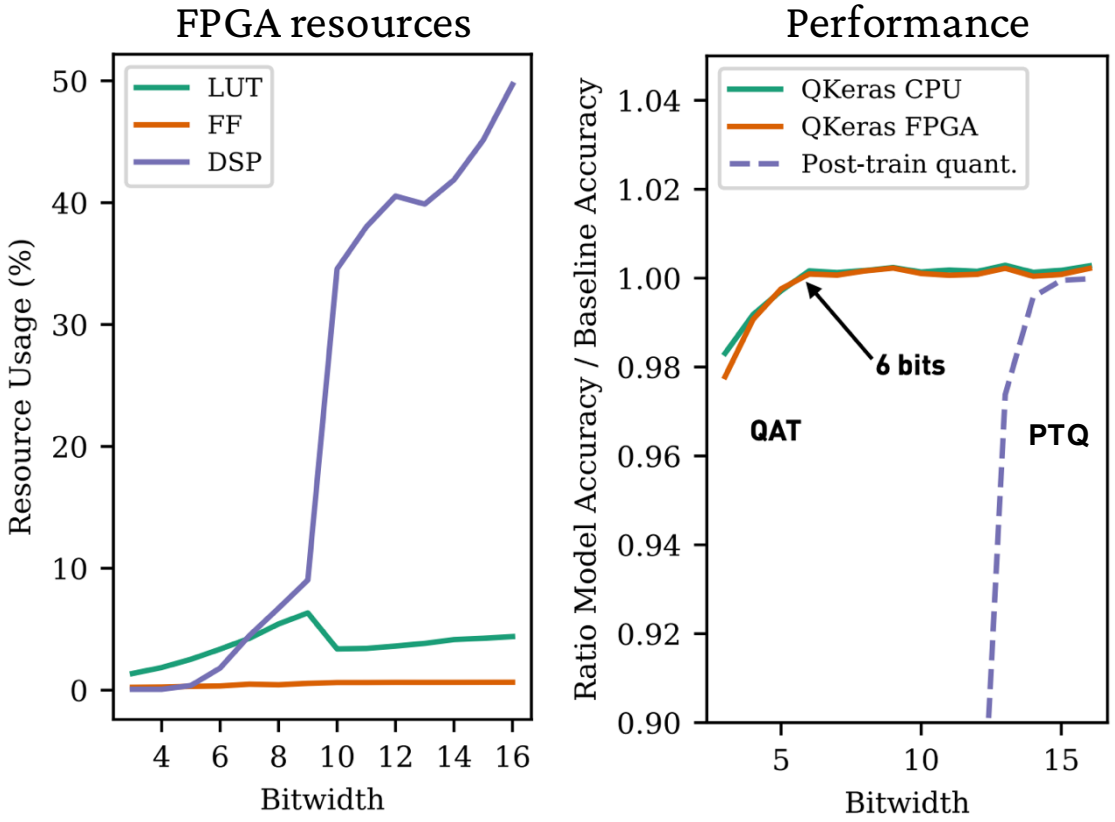




arXiv:2406.19522

Deploying ML on FPGAs

- FPGAs provide low-latency, high-throughput inference through parallelism and pipelining.
- Quantisation and pruning reduce numerical and computational complexity, lowering the hardware cost of ML inference.
- Quantisation-aware training (QAT) lets the model adapt to reduced precision, retaining performance better than post-training quantisation (PTQ) at aggressive bit widths.
- Different quantisation strategies offer different accuracy–hardware trade-offs.



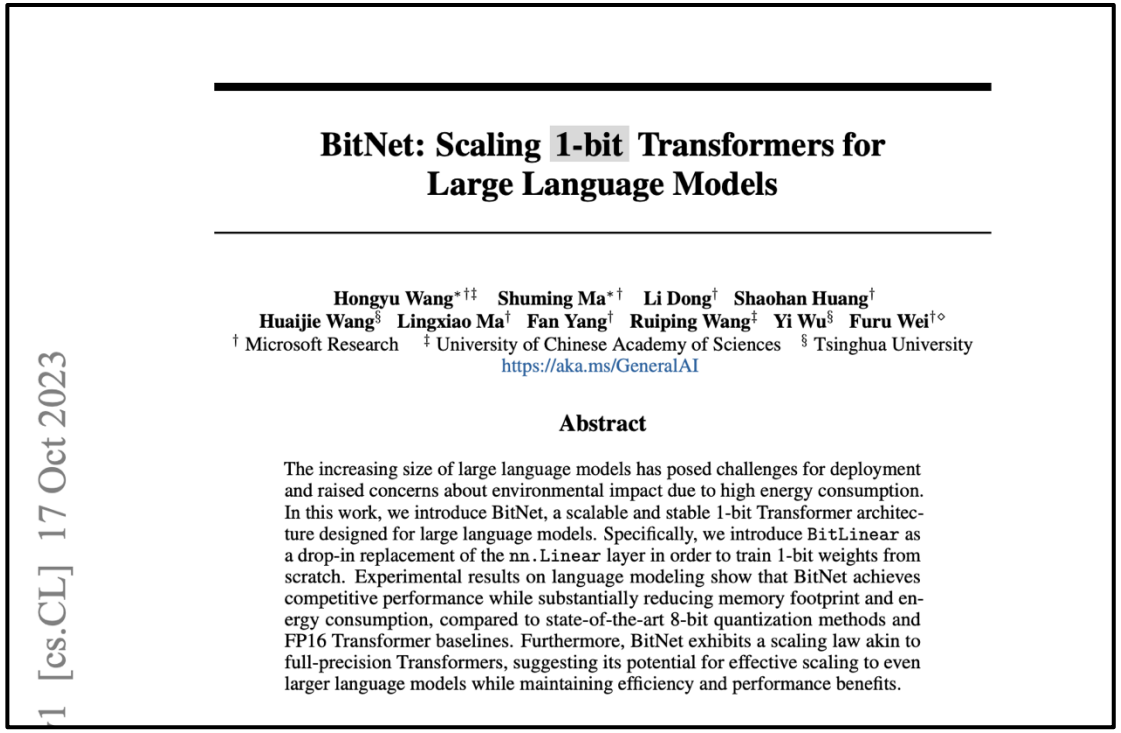
arXiv:2006.10159

BitNet: Binary-Weight Networks

- Training transformers from scratch with 1-bit weights to reduce memory and computational cost.
- BitNet remains competitive with conventional Transformer baselines and shows similar scaling behaviour as model size increases.
- Substantially lower estimated memory and arithmetic-energy requirements.

30B LLM: ~38x lower estimated arithmetic energy

2023 Binary → 2024 Ternary → 2024 SLMs→ 2025 BitHEP → 2026 FastML



BitNet: Scaling 1-bit Transformers for Large Language Models

Hongyu Wang*† Shuming Ma*† Li Dong† Shaohan Huang†
Huaijie Wang§ Lingxiao Ma† Fan Yang† Ruiping Wang‡ Yi Wu§ Furu Wei†◇
† Microsoft Research ‡ University of Chinese Academy of Sciences § Tsinghua University
https://aka.ms/GeneralAI

Abstract

The increasing size of large language models has posed challenges for deployment and raised concerns about environmental impact due to high energy consumption. In this work, we introduce BitNet, a scalable and stable 1-bit Transformer architecture designed for large language models. Specifically, we introduce `BitLinear` as a drop-in replacement of the `nn.Linear` layer in order to train 1-bit weights from scratch. Experimental results on language modeling show that BitNet achieves competitive performance while substantially reducing memory footprint and energy consumption, compared to state-of-the-art 8-bit quantization methods and FP16 Transformer baselines. Furthermore, BitNet exhibits a scaling law akin to full-precision Transformers, suggesting its potential for effective scaling to even larger language models while maintaining efficiency and performance benefits.

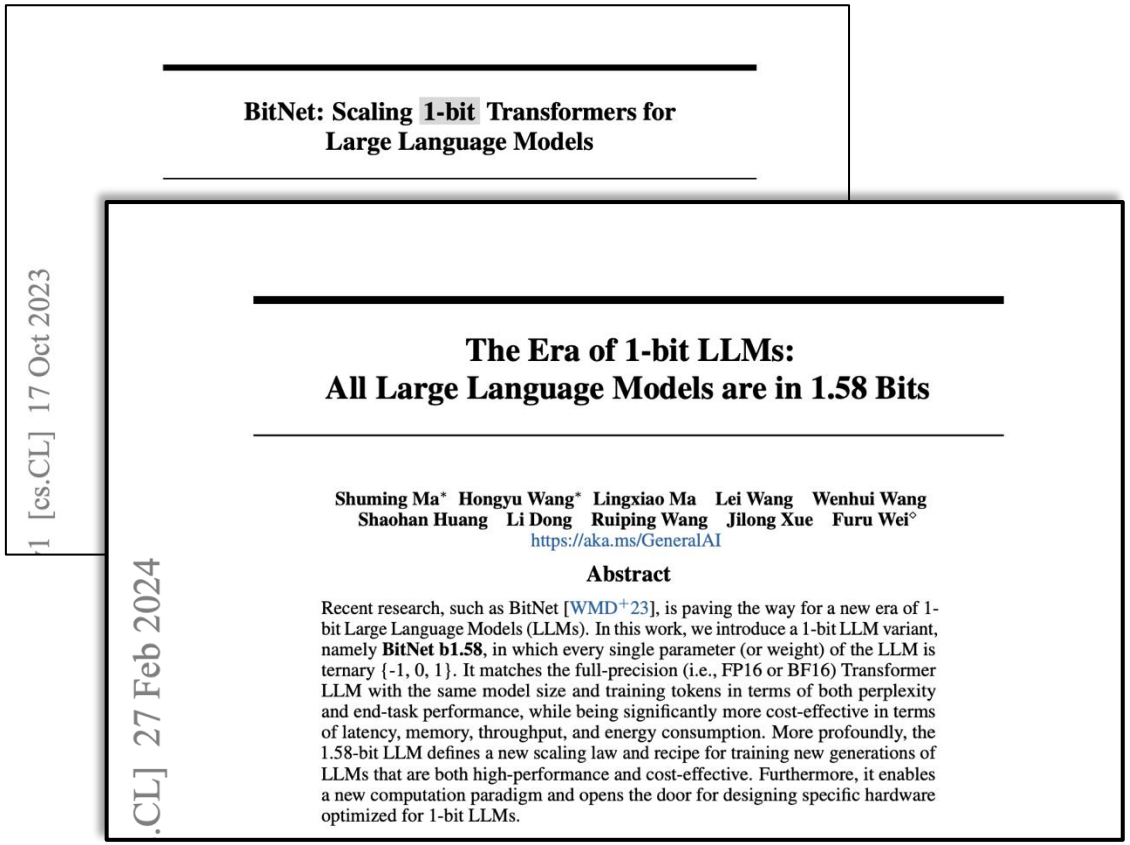
arXiv:2310.11453

BitNet b1.58: Adding Zero

2023 Binary → **2024 Ternary** → 2024 SLMs→ 2025 BitHEP → 2026 FastML

- Binary is highly restrictive. BitNet b1.58 adds a zero state:
  $W_q \in \{-, 1, 0 + 1\}, \log_2(3) \approx 1.58$ bits
- At matched model size and training budget, b1.58 approaches or matches FP16/BF16 Transformer performance.
- The zero state adds sparsity while retaining ultra-low precision.

Memory: 7.89 → 2.22 GB (**3.55× lower**)
Latency: 5.07 → 1.87 ms (**2.71× faster**)



BitNet: Scaling 1-bit Transformers for Large Language Models

[cs.CL] 17 Oct 2023

The Era of 1-bit LLMs:
All Large Language Models are in 1.58 Bits

Shuming Ma* Hongyu Wang* Lingxiao Ma Lei Wang Wenhui Wang
Shaohan Huang Li Dong Ruiping Wang Jilong Xue Furu Wei°
https://aka.ms/GeneralAI

Abstract

Recent research, such as BitNet [WMD+23], is paving the way for a new era of 1-bit Large Language Models (LLMs). In this work, we introduce a 1-bit LLM variant, namely **BitNet b1.58**, in which every single parameter (or weight) of the LLM is ternary {-1, 0, 1}. It matches the full-precision (i.e., FP16 or BF16) Transformer LLM with the same model size and training tokens in terms of both perplexity and end-task performance, while being significantly more cost-effective in terms of latency, memory, throughput, and energy consumption. More profoundly, the 1.58-bit LLM defines a new scaling law and recipe for training new generations of LLMs that are both high-performance and cost-effective. Furthermore, it enables a new computation paradigm and opens the door for designing specific hardware optimized for 1-bit LLMs.

.CL] 27 Feb 2024

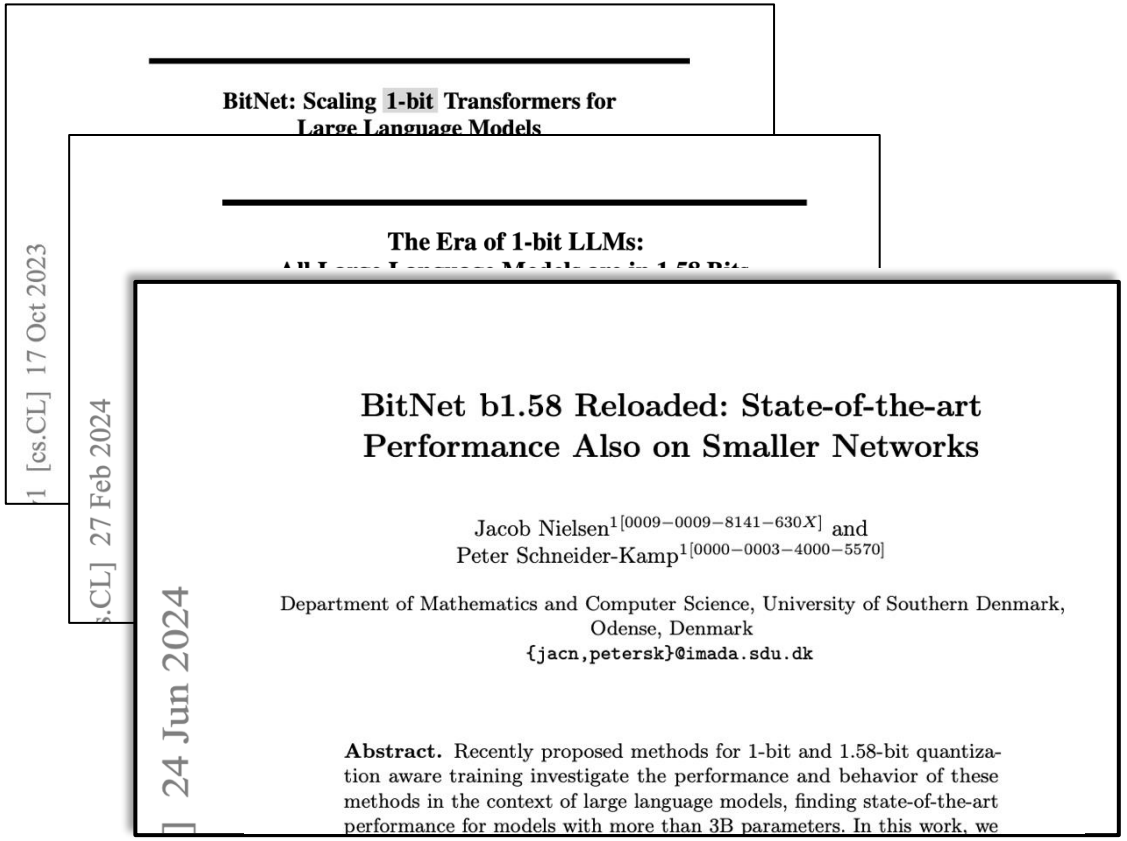
arXiv:2402.17764

BitNet b1.58 Reloaded

- BitNet is not limited to billion-parameter LLMs.
- Tested on much smaller language and vision models: ~100k–48M parameters.
- Performance remains competitive at small scale, with vision models matching or surpassing comparable same-size baselines.

BitNet can work well far below LLM scale

2023 Binary → 2024 Ternary → **2024 SLMs** → 2025 BitHEP → 2026 FastML



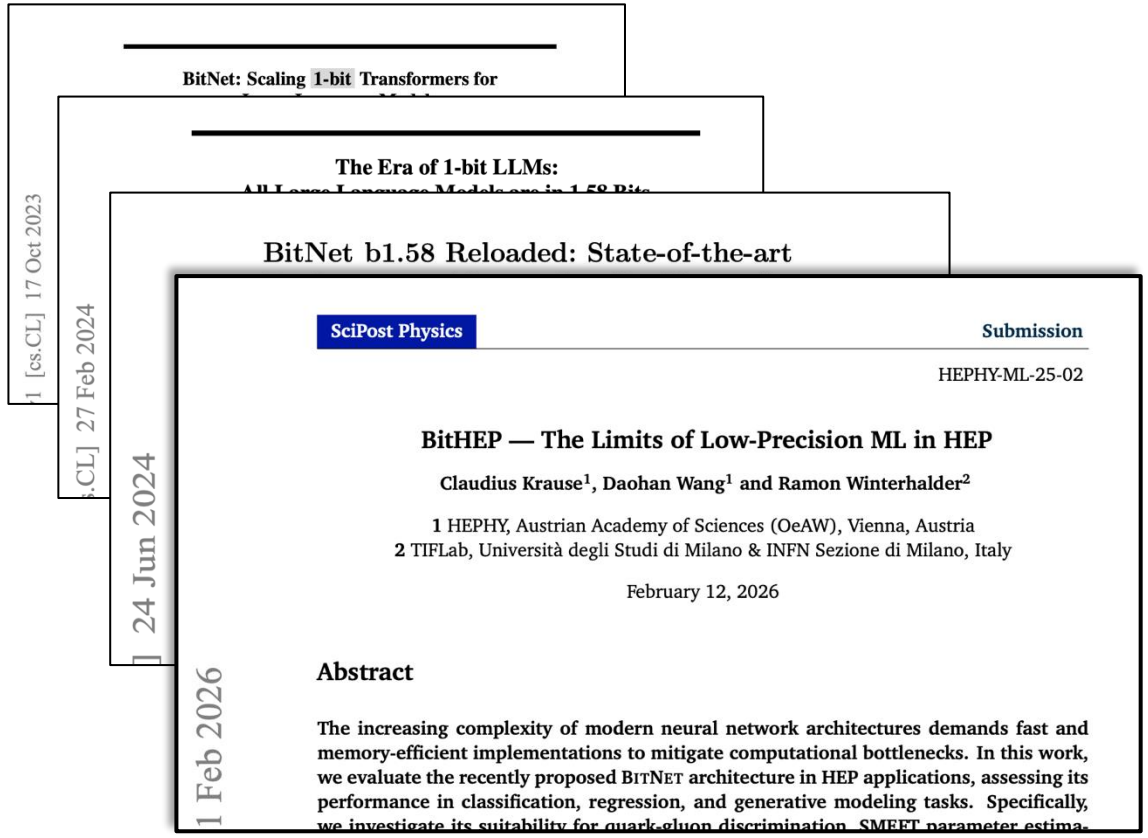
arXiv:2407.09527

BitHEP: Applying BitNet to HEP

- Various tasks: quark/gluon classification, SMEFT regression and detector generative modelling.
- Performance is task-dependent: strongest and most consistent for classification, with more mixed behaviour for regression and generation.
- Large theoretical compute reductions are possible.

Equivalent compute proxy: ~41-6% of baseline

2023 Binary → 2024 Ternary → 2024 SLMs → **2025 BitHEP** → 2026 FastML

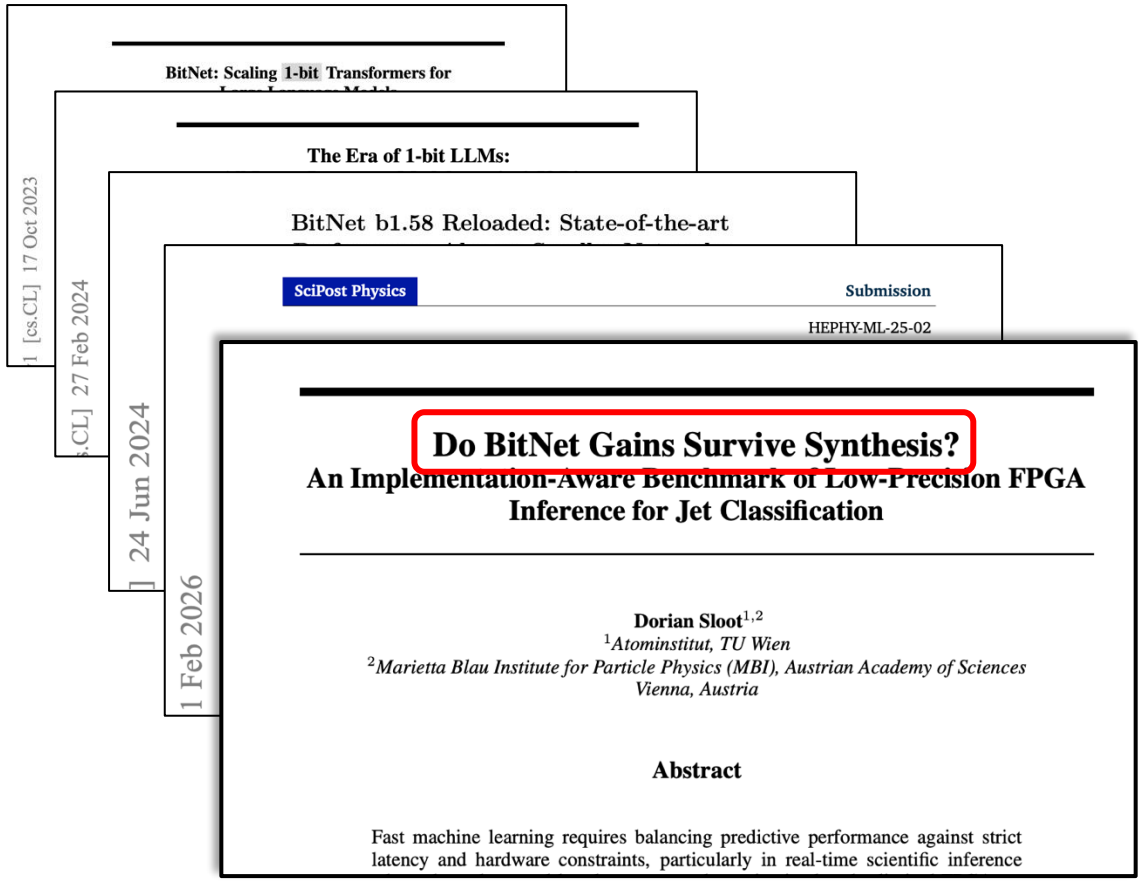


SciPostPhys.20.2.038

Real-Time BitHEP

- Previous BitNet studies show gains in predictive performance and theoretical/software efficiency, but stop short of hardware synthesis.
- Bit width is only part of the story: scaling, sparsity, scheduling and conversion affect the realised hardware cost.
- This work synthesises BitNet and benchmarks it against established low-precision methods under identical FPGA constraints.

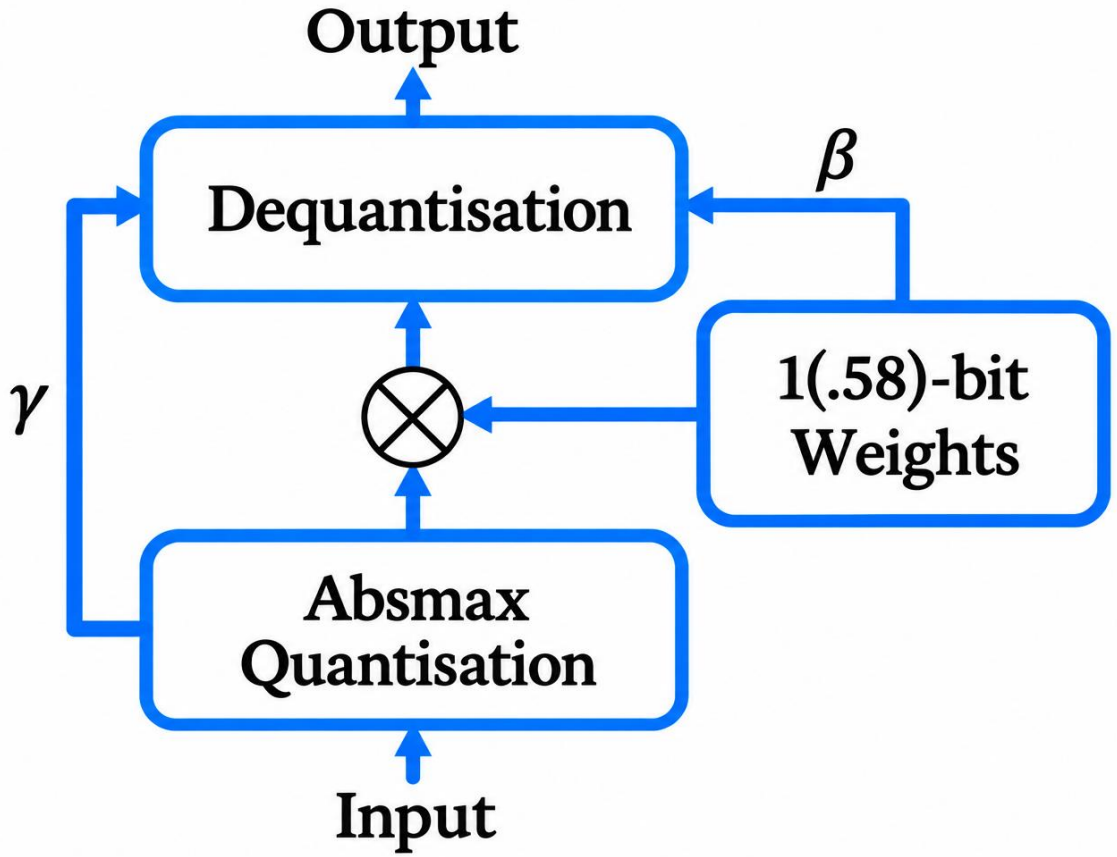
2023 Binary → 2024 Ternary → 2024 SLMs → 2025 BitHEP → **2026 FastML**



OpenReview: huKsxMpIIa

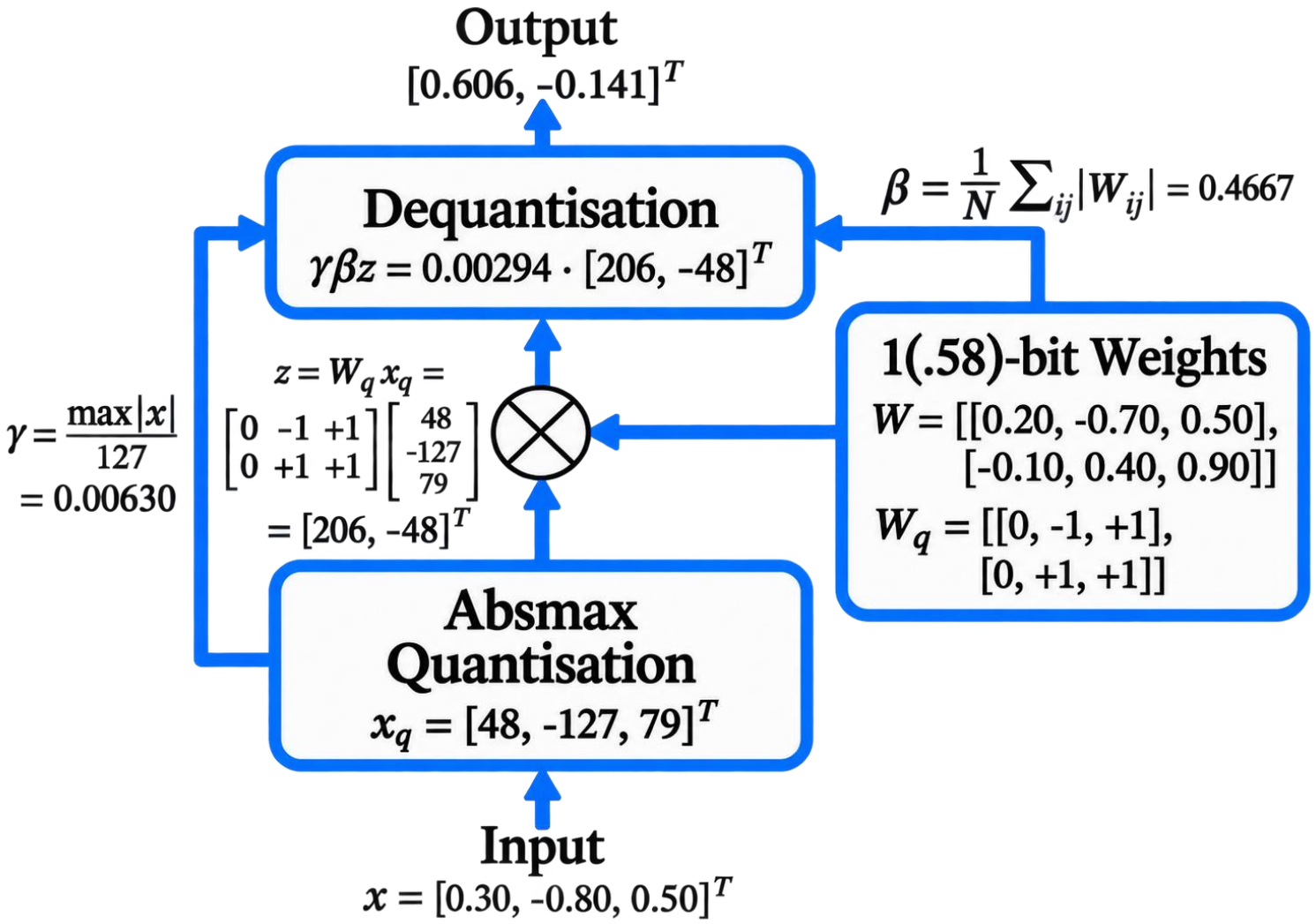
How does BitNet work?

- BitLinear quantises weights to binary or ternary values, replacing general multiplications with sign changes, additions and zeros.
- Scaling factors restore the dynamic range lost through aggressive quantisation and help retain predictive performance.
- The hardware benefit is therefore not guaranteed: scaling, quantisation logic and the resulting synthesised datapath can introduce additional cost.



BitLinear: A Toy Example

- BitLinear quantises weights to ternary values.
- Scaling factors restore the dynamic range.
- The hardware benefit is not guaranteed.

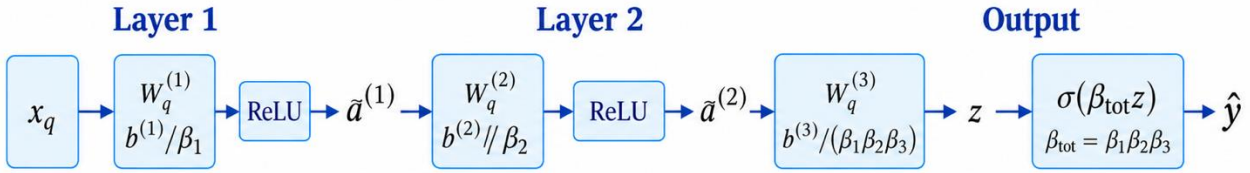


BitNet Scaling for FPGA Inference

- Dynamic activation scaling (γ): per-event AbsMax requires runtime reduction and rescaling.
- Static weight scaling: for $\beta > 0$, propagate the scale through ReLU and compensate downstream biases.
- Output folding: absorb the accumulated β_{tot} into the final sigmoid, eliminating explicit hidden-layer β multipliers.

$$u = wx + b = (\beta w_q)(\gamma x_q) + b = \beta(w_q x_q) + b = \beta(w_q x_q + b/\beta) = \beta(w_q x_q + b_q) = \beta u_q$$

$$\beta > 0$$
$$\text{ReLU}(\beta u_q) = \beta\,\text{ReLU}(u_q)$$

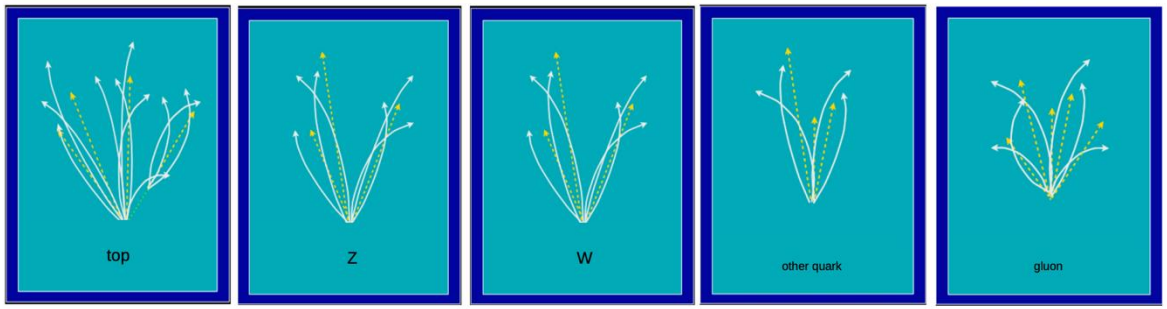


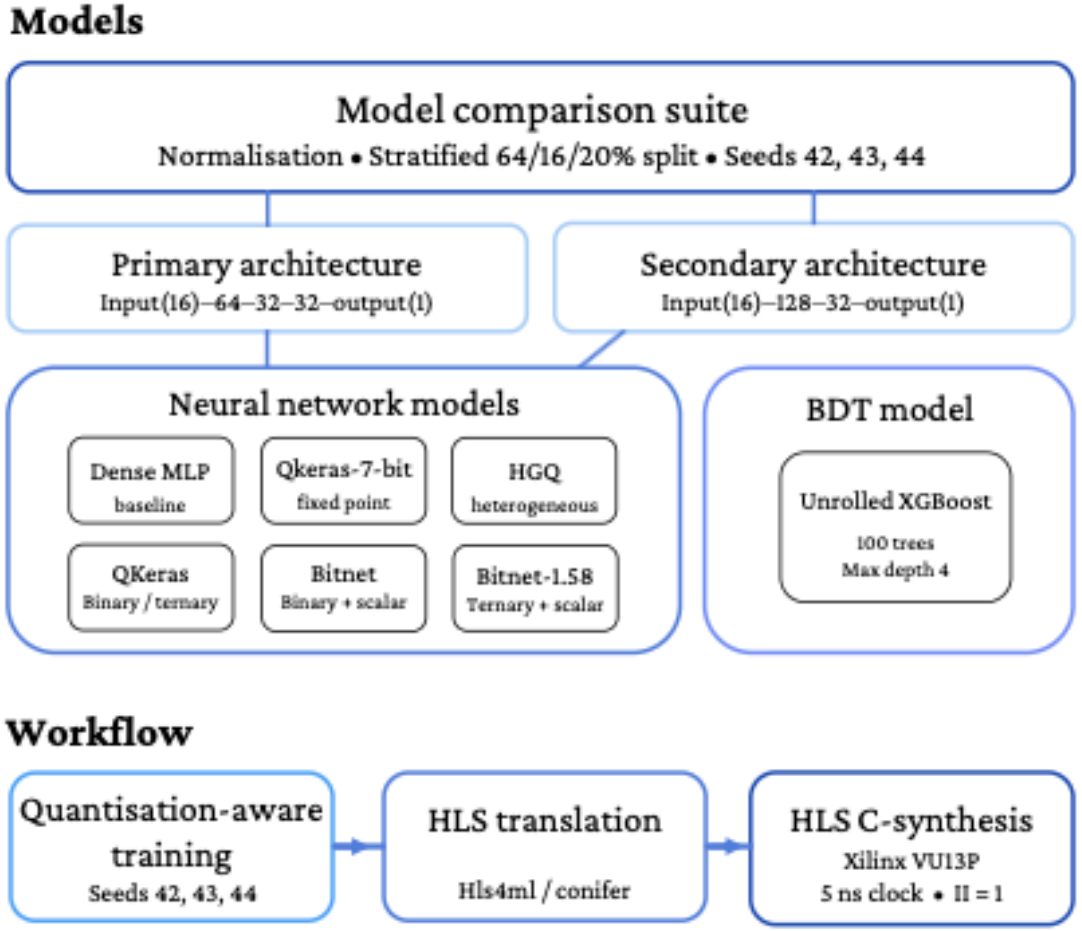
$$\tilde{a}^{(1)} = \text{ReLU}\left(W_q^{(1)} x_q + b^{(1)}/\beta_1\right)$$
$$\tilde{a}^{(2)} = \text{ReLU}\left(W_q^{(2)} \tilde{a}^{(1)} + b^{(2)}/\beta_2\right)$$
$$z = W_q^{(3)} \tilde{a}^{(2)} + b^{(3)}/(\beta_1\beta_2\beta_3)$$
$$\hat{y} = \sigma((\beta_1\beta_2\beta_3) z)$$

Push β through ReLU, divide downstream biases, absorb β_{tot} into the final sigmoid.

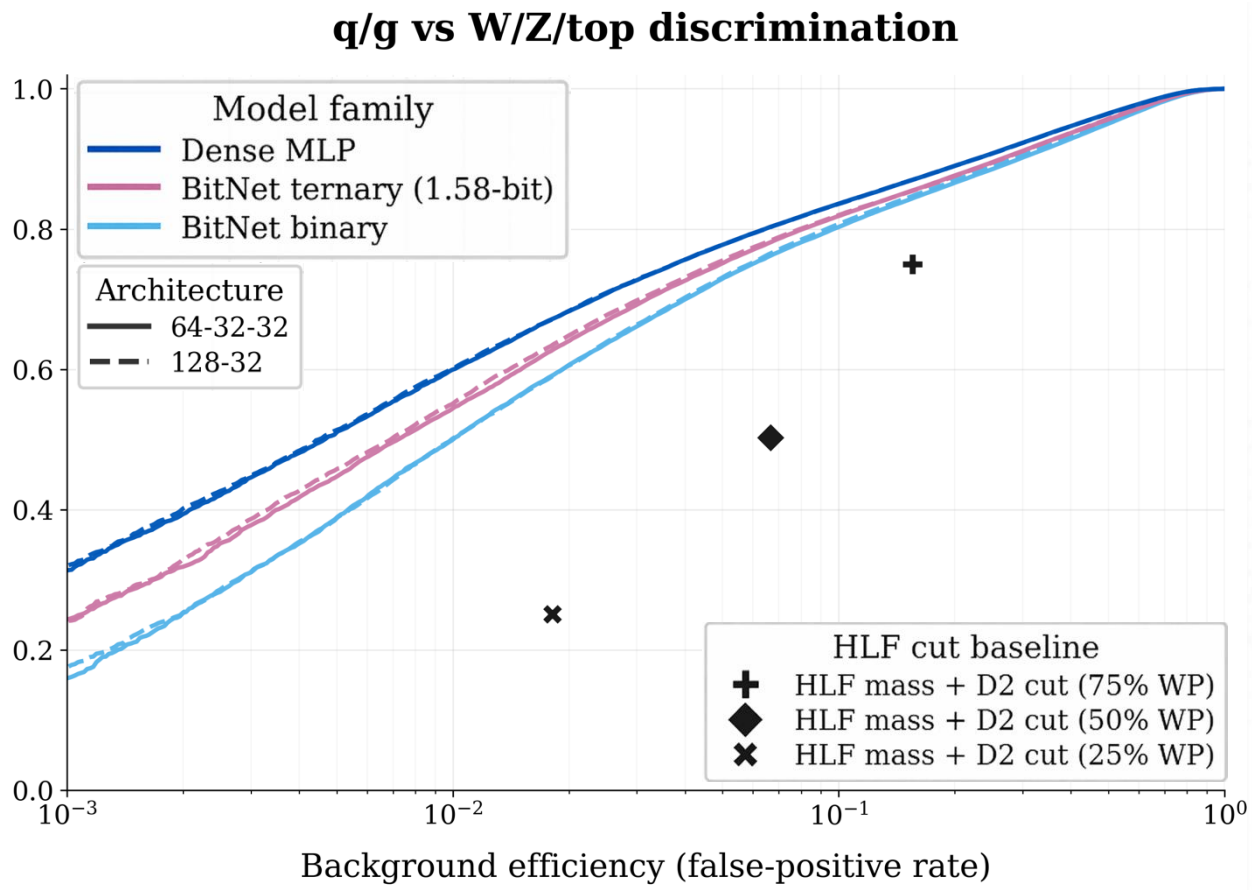
A Controlled Benchmark

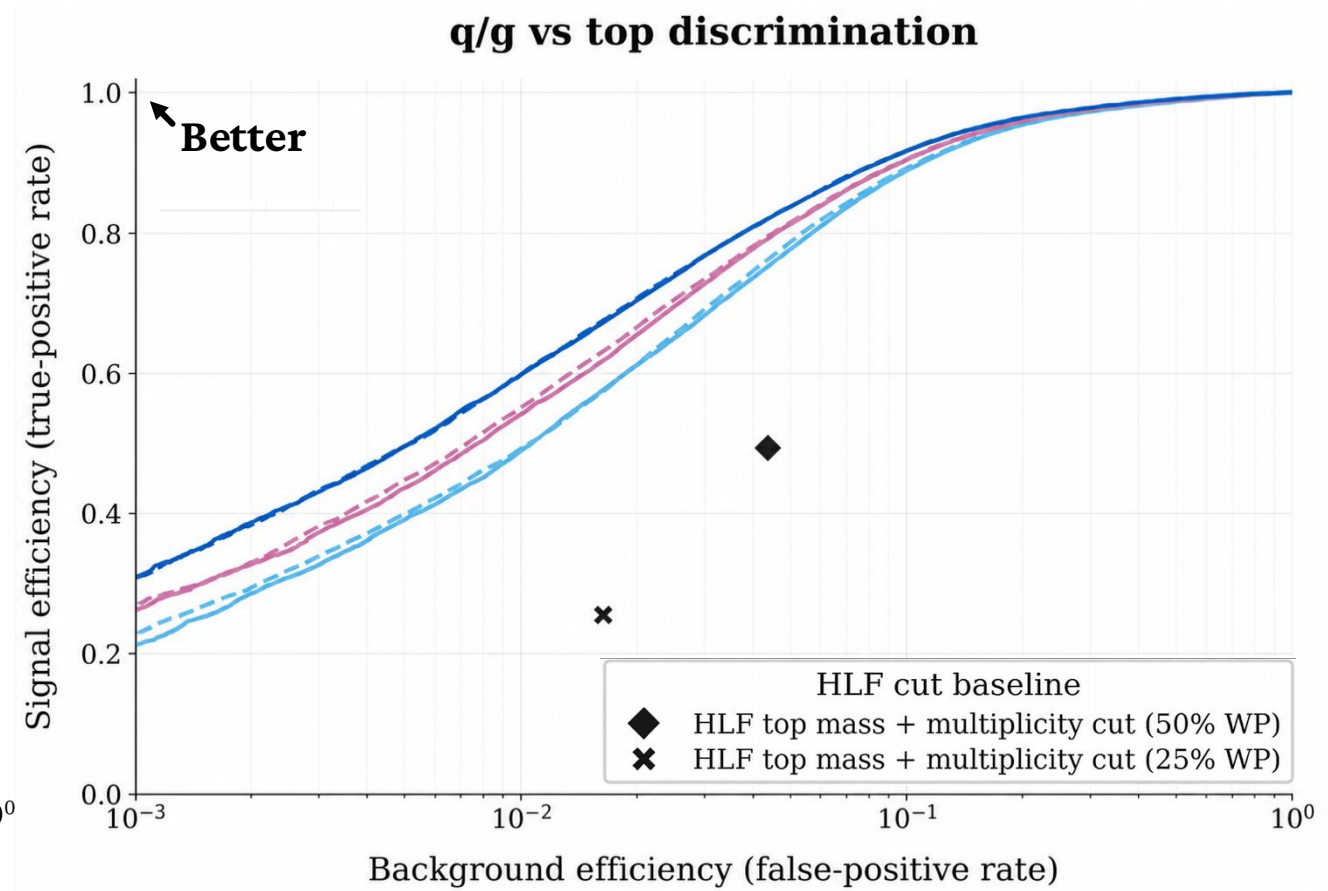
- Public hls4ml LHC Jets HLF dataset (OpenML):
  - ~830k balanced jets,
  - 16 high-level observables,
  - Five classes.
- 3 Tasks:
  - q/g vs. $W/Z/t$,
  - q/g vs. top,
  - Multiclass classification.



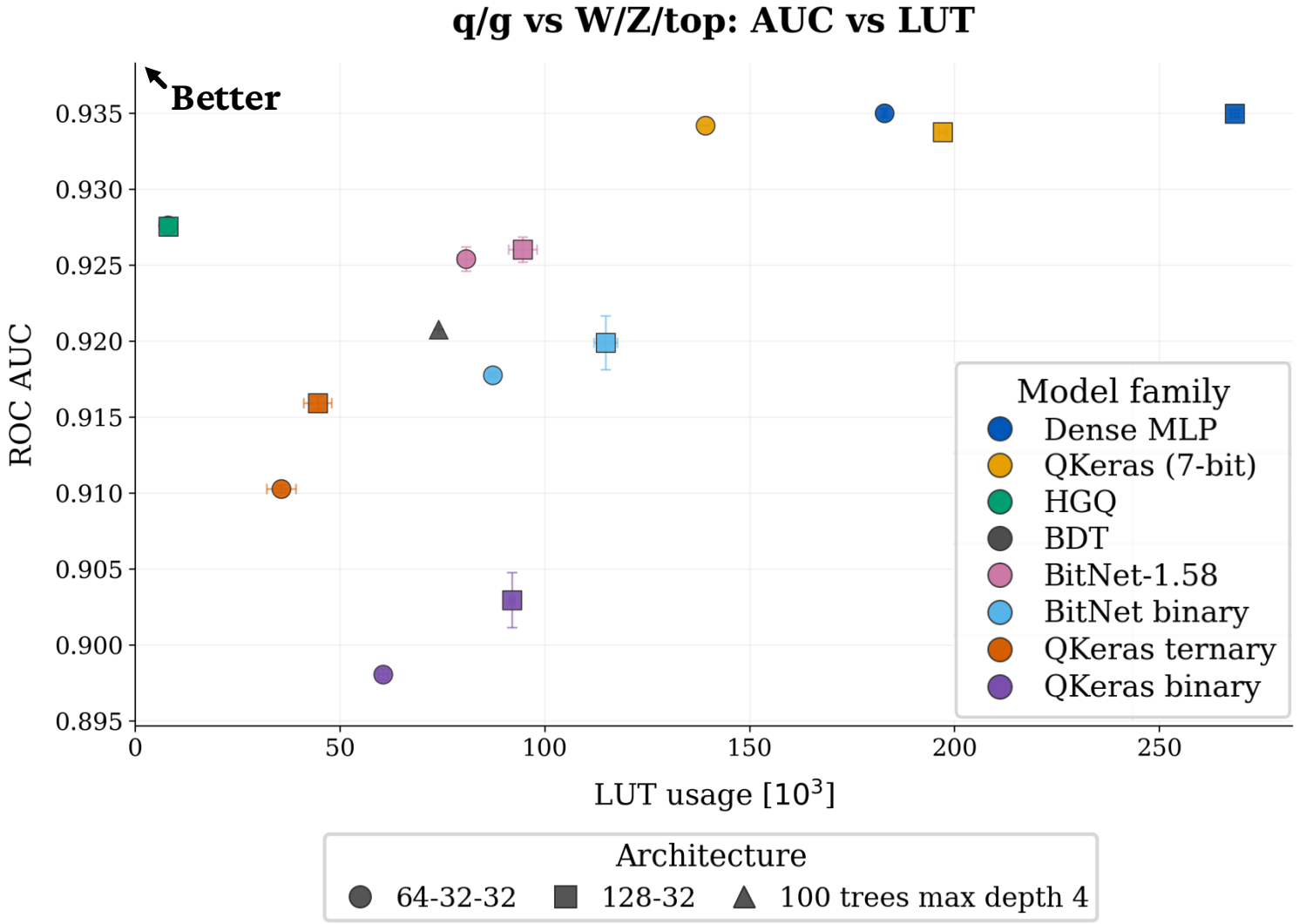


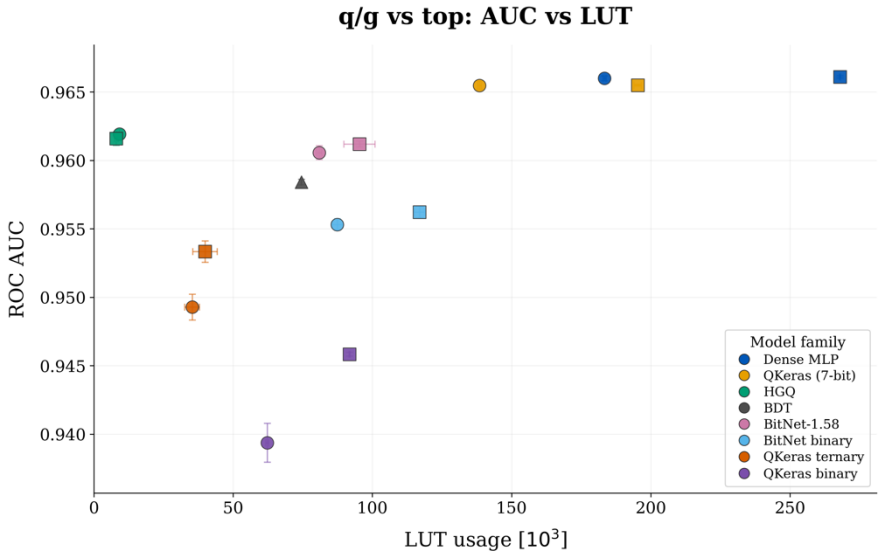
Predictive Performance

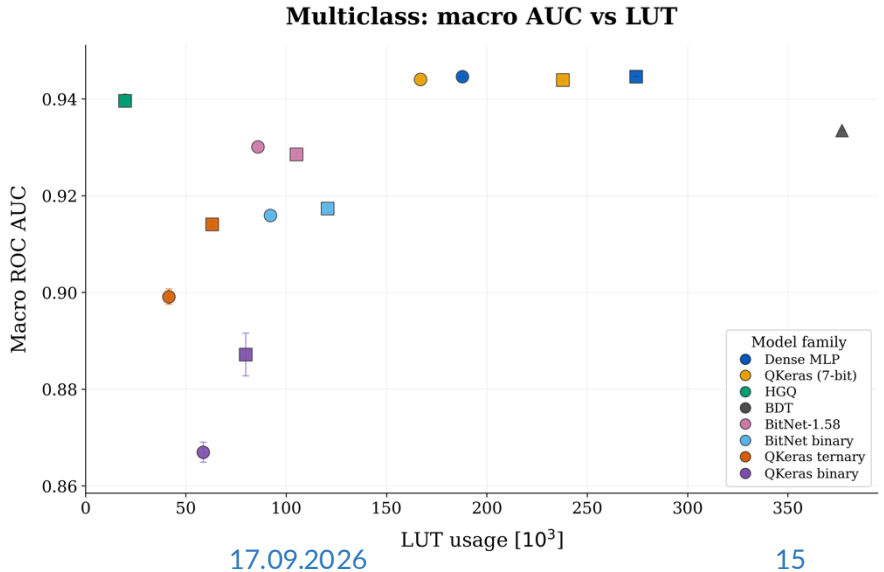




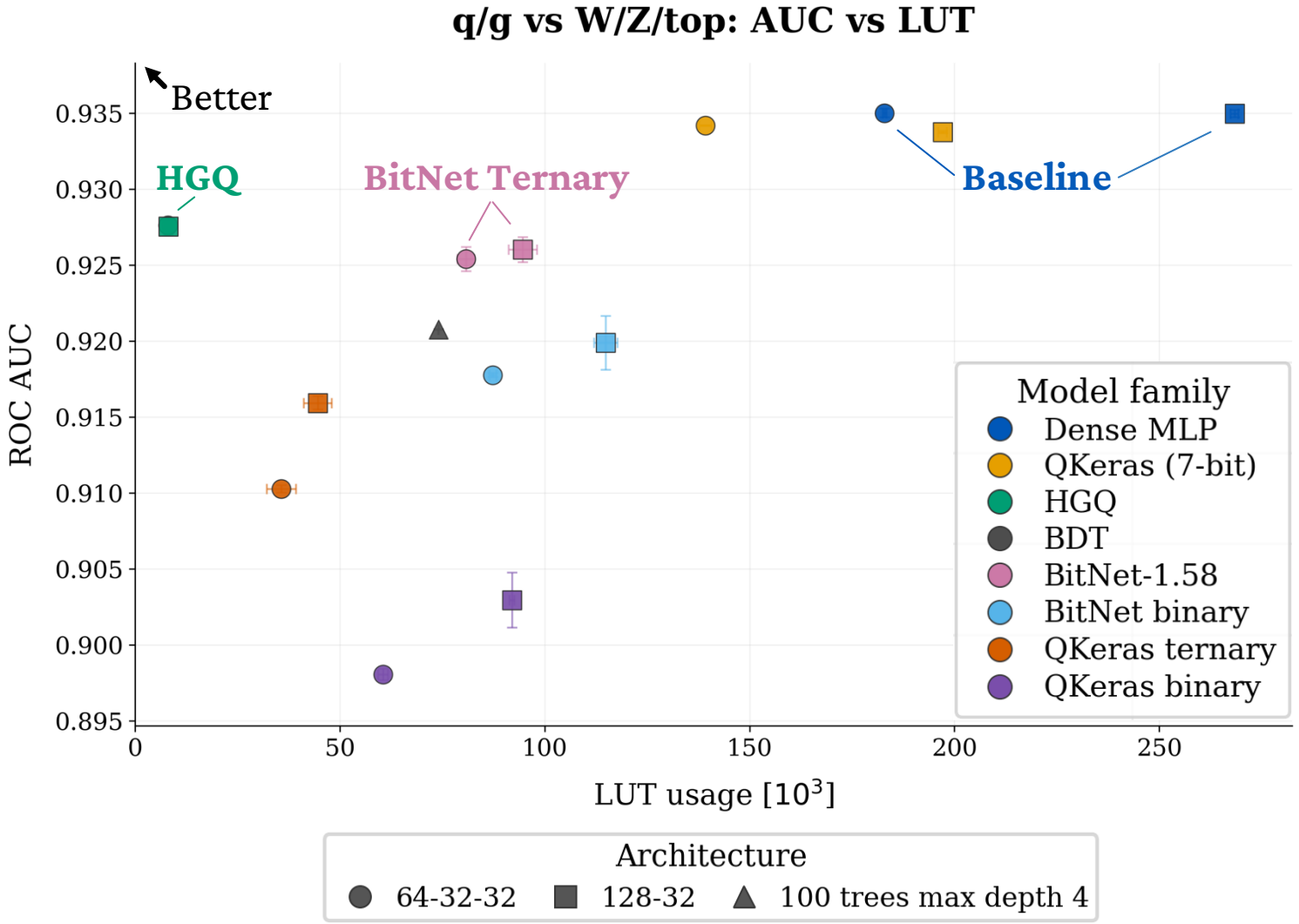
Performance vs. FPGA Resources

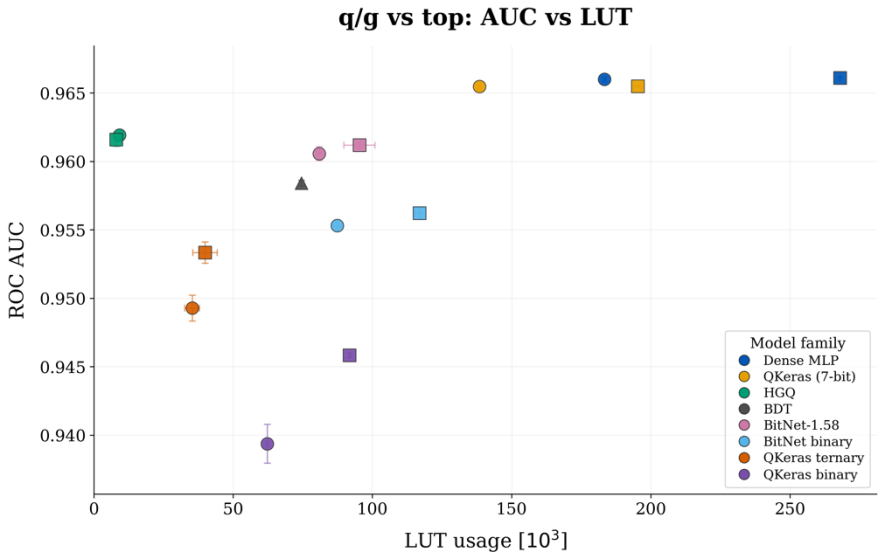


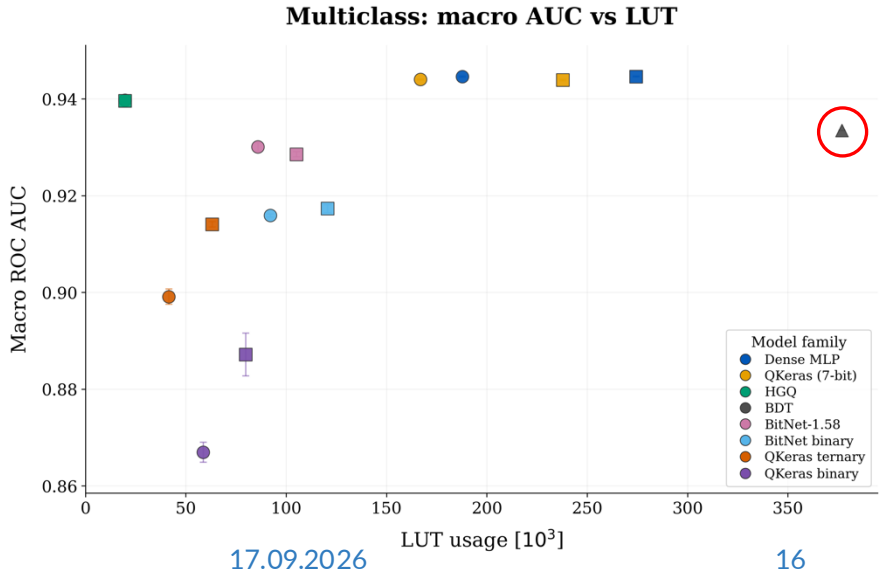




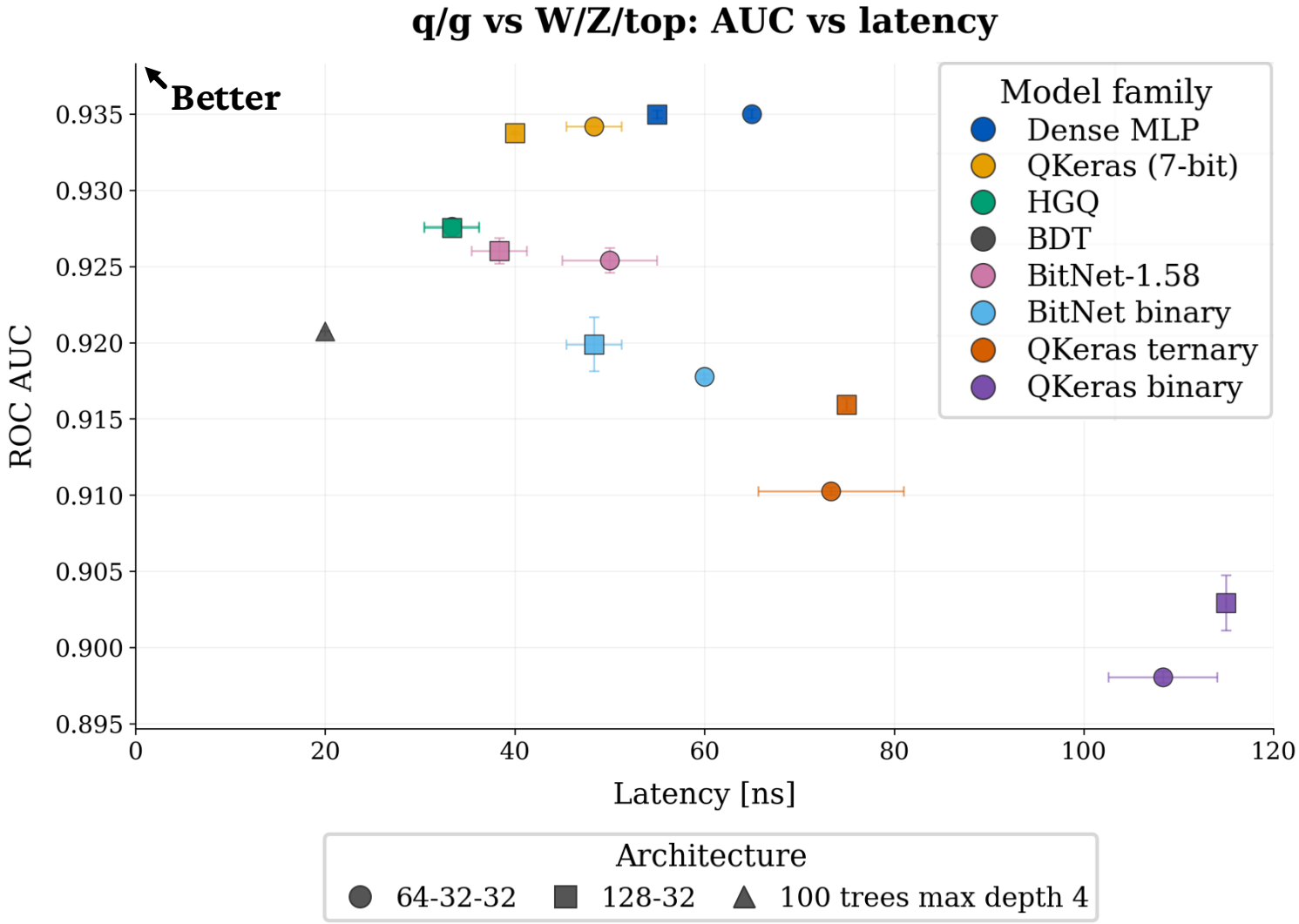
Performance vs. FPGA Resources

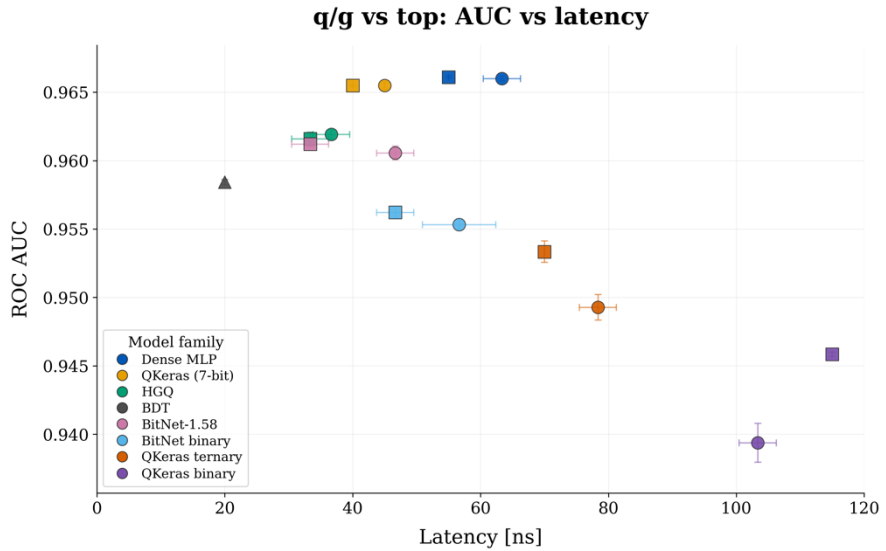


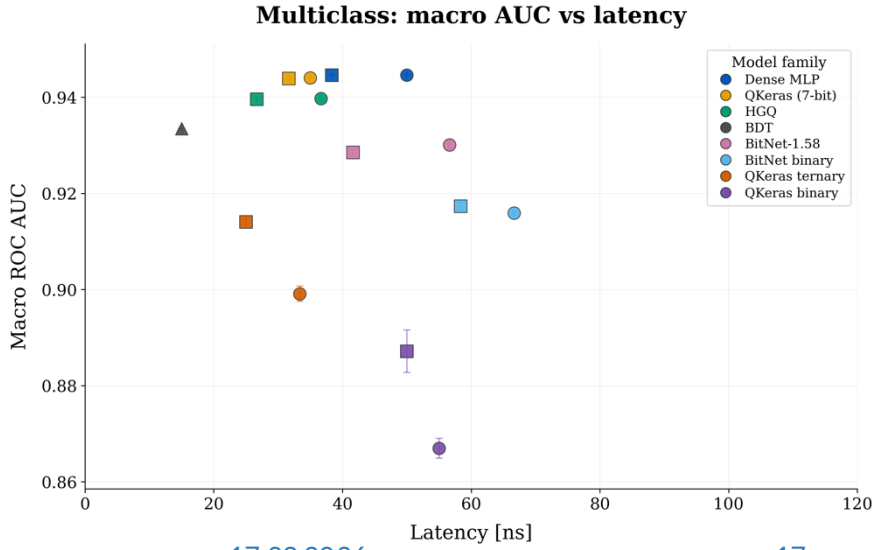




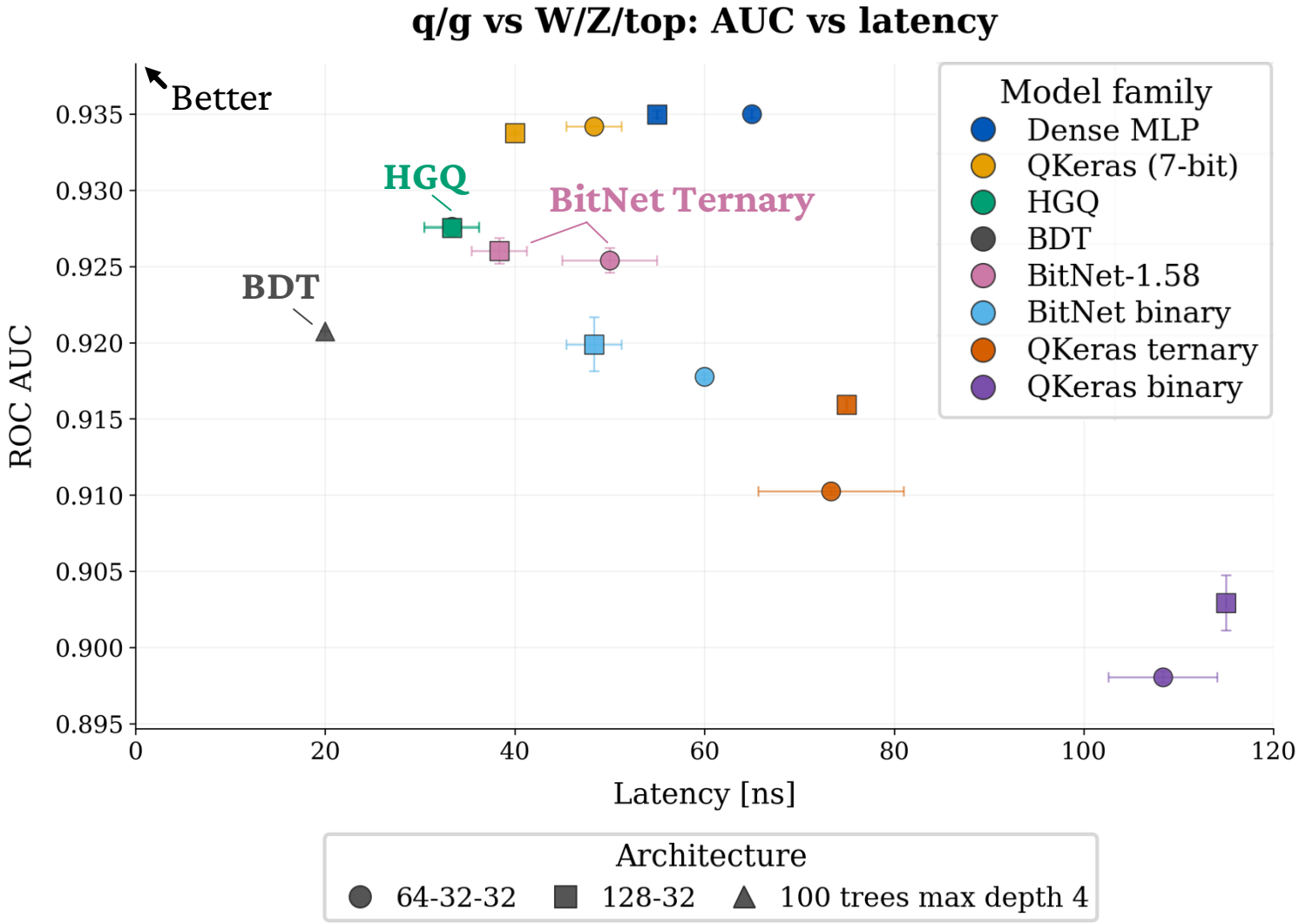
Performance vs. Latency

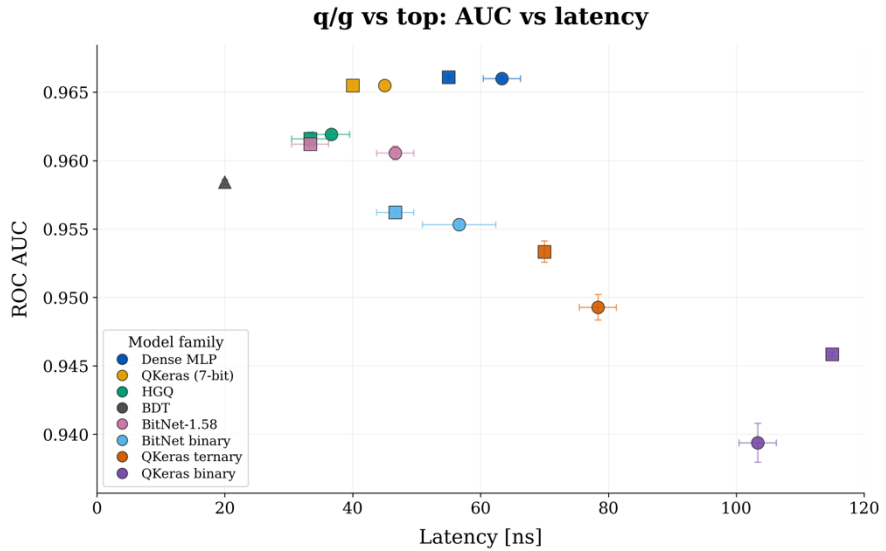


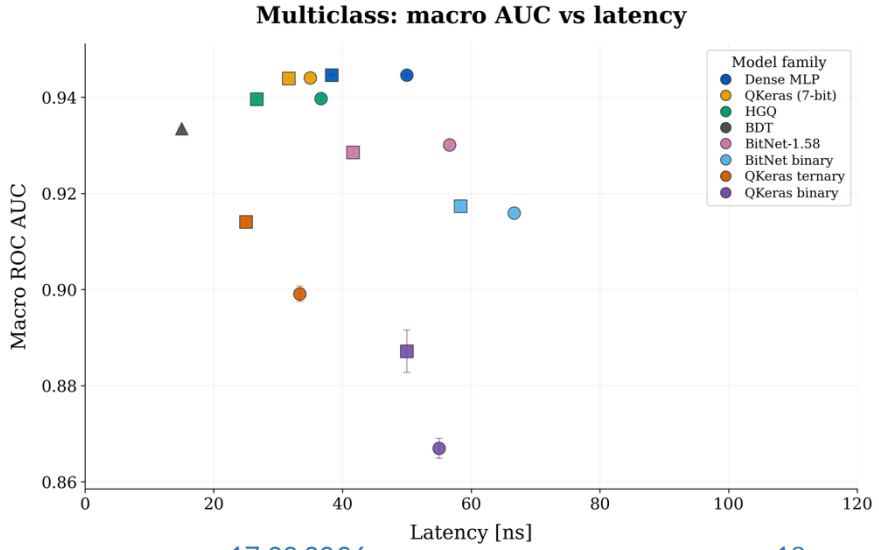




Performance vs. Latency



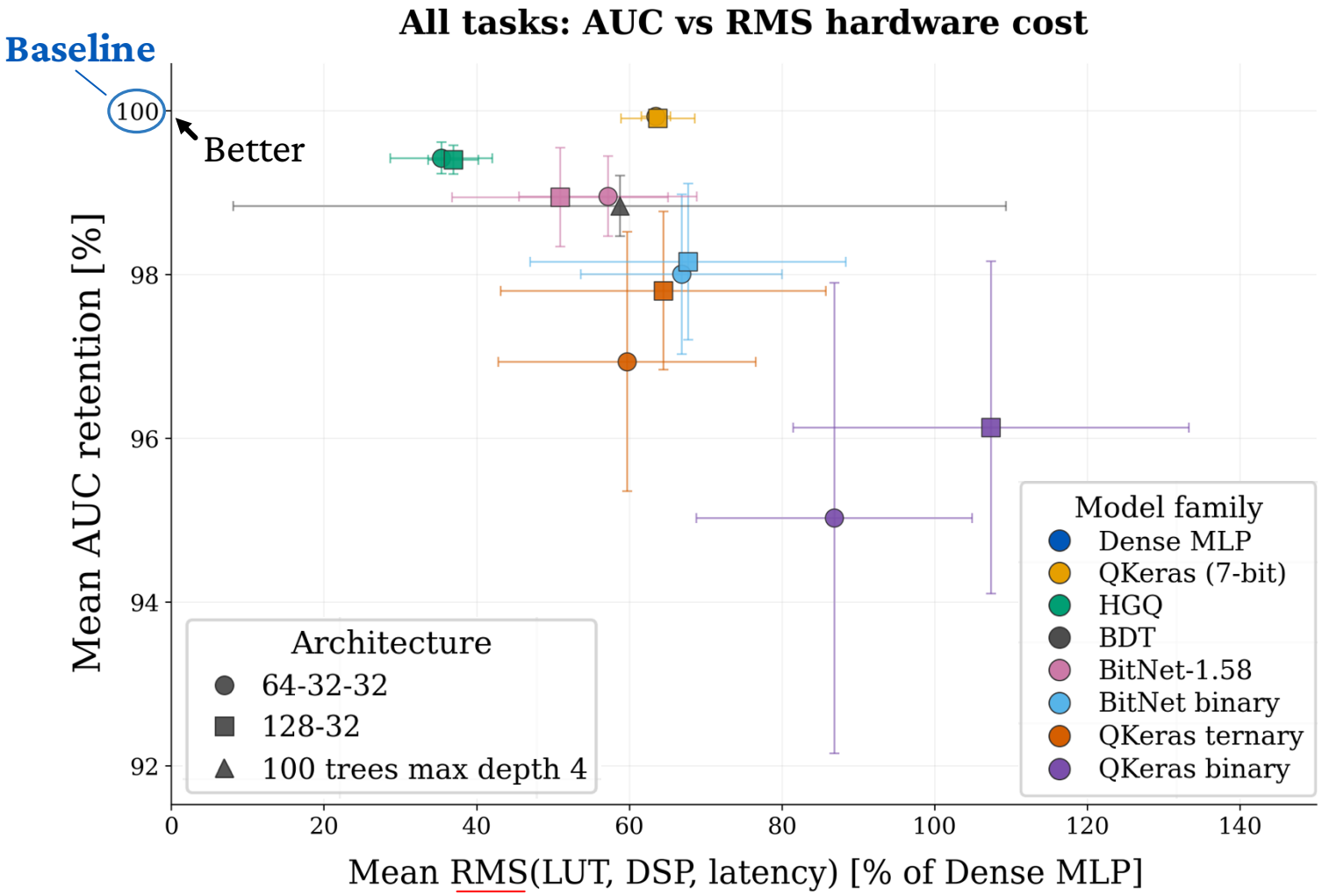




Combined Pareto Frontier

- HGQ achieves strong resource efficiency through learned heterogeneous precision and hardware-cost regularisation.
- BitNet-1.58 remains a strong ultra-low-bit operating point:
  - Ternary weights enable efficient storage and streaming,
  - Discrete structure can aid model interpretability studies.

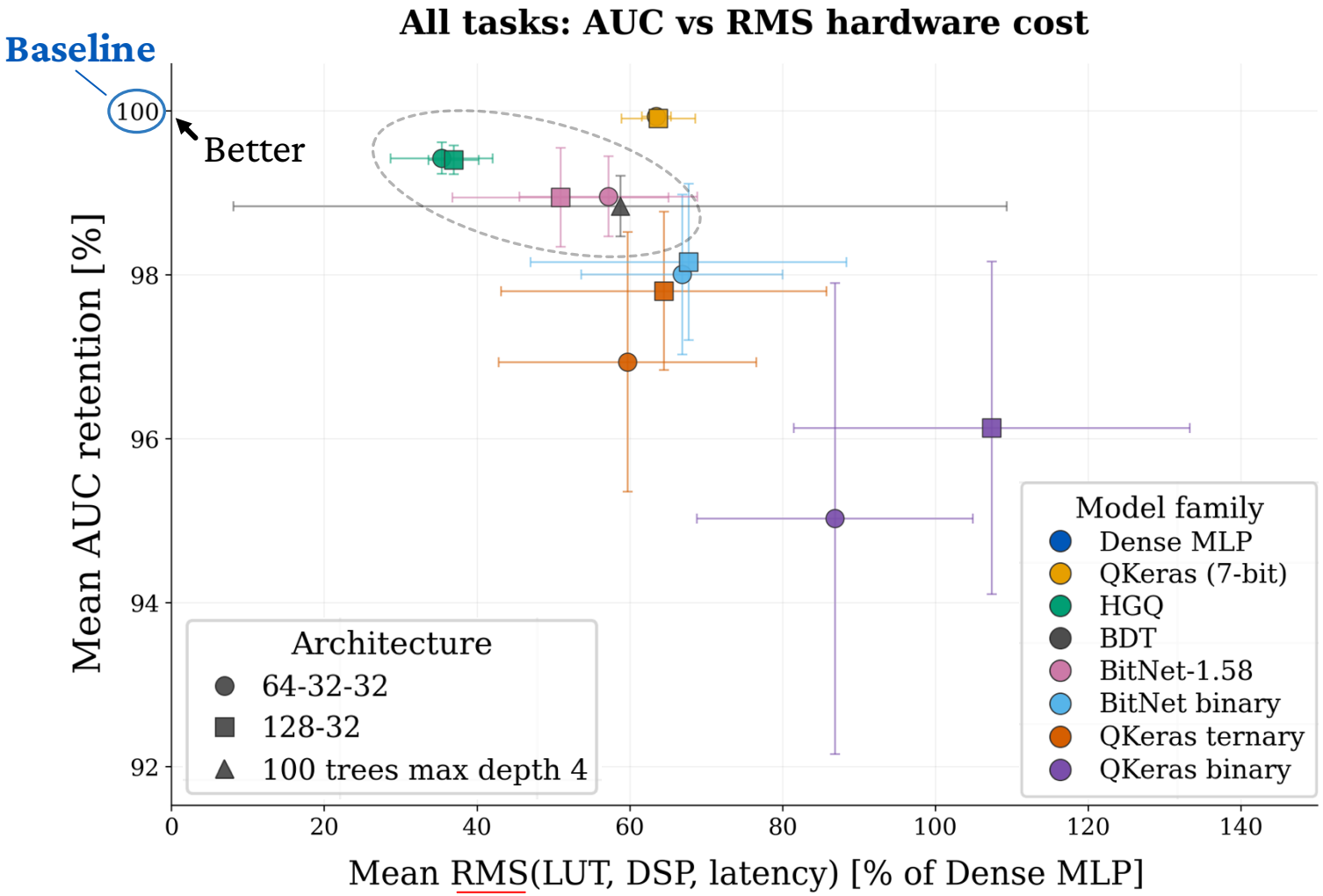
$$RMS_{hw} = \sqrt{\frac{r_{LUT}^2 + r_{DSP}^2 + r_{latency}^2}{3}}$$



Combined Pareto Frontier

- HGQ achieves strong resource efficiency through learned heterogeneous precision and hardware-cost regularisation.
- BitNet-1.58 remains a strong ultra-low-bit operating point:
  - Ternary weights enable efficient storage and streaming,
  - Discrete structure can aid model interpretability studies.

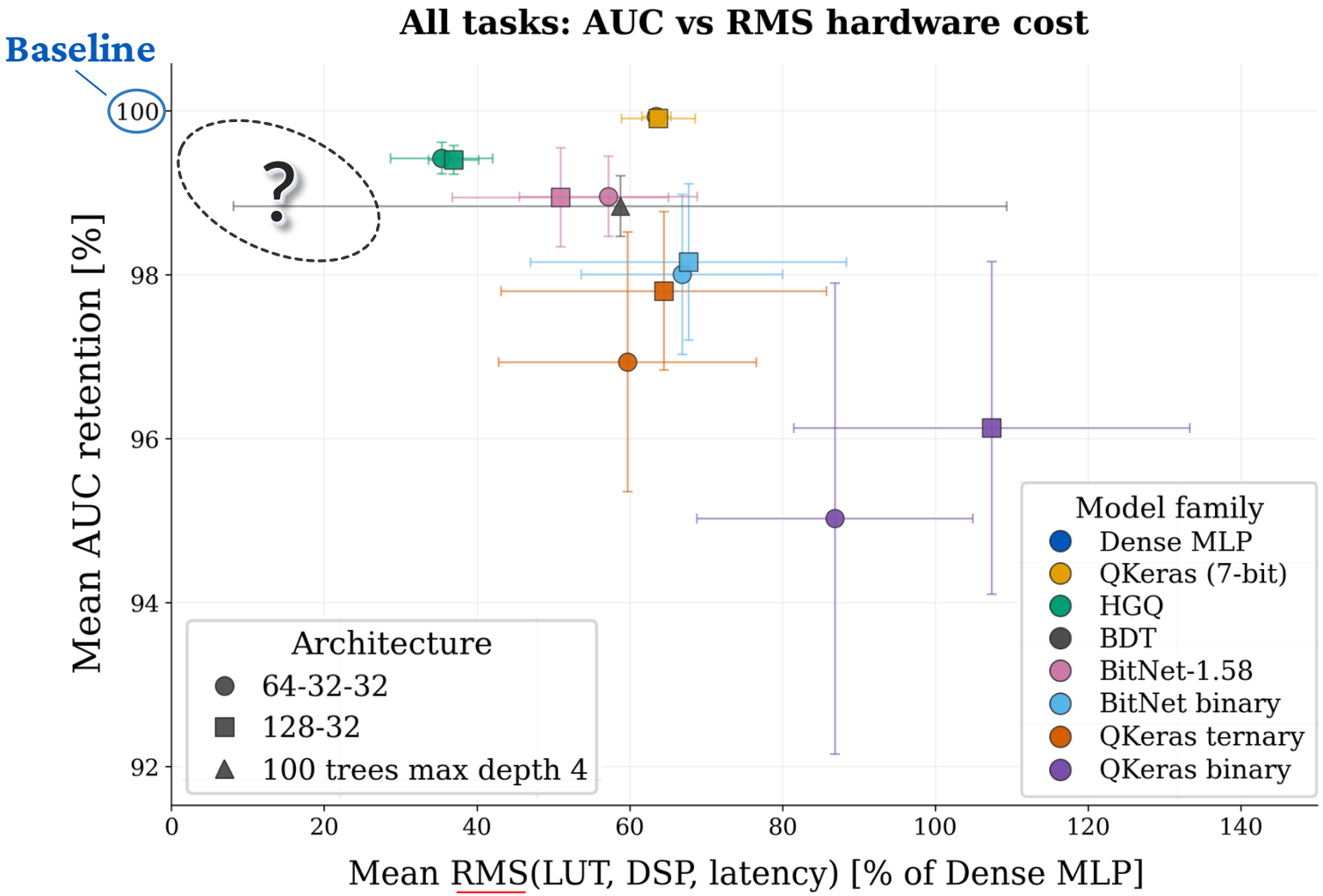
$$RMS_{hw} = \sqrt{\frac{r_{LUT}^2 + r_{DSP}^2 + r_{latency}^2}{3}}$$



Combined Pareto Frontier

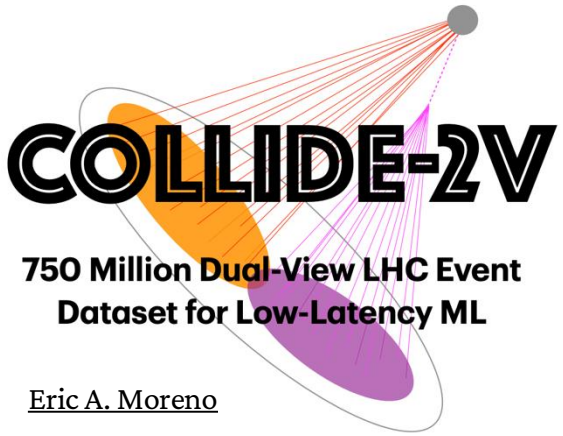
- HGQ achieves strong resource efficiency through learned heterogeneous precision and hardware-cost regularisation.
- BitNet-1.58 remains a strong ultra-low-bit operating point:
  - Ternary weights enable efficient storage and streaming,
  - Discrete structure can aid model interpretability studies.

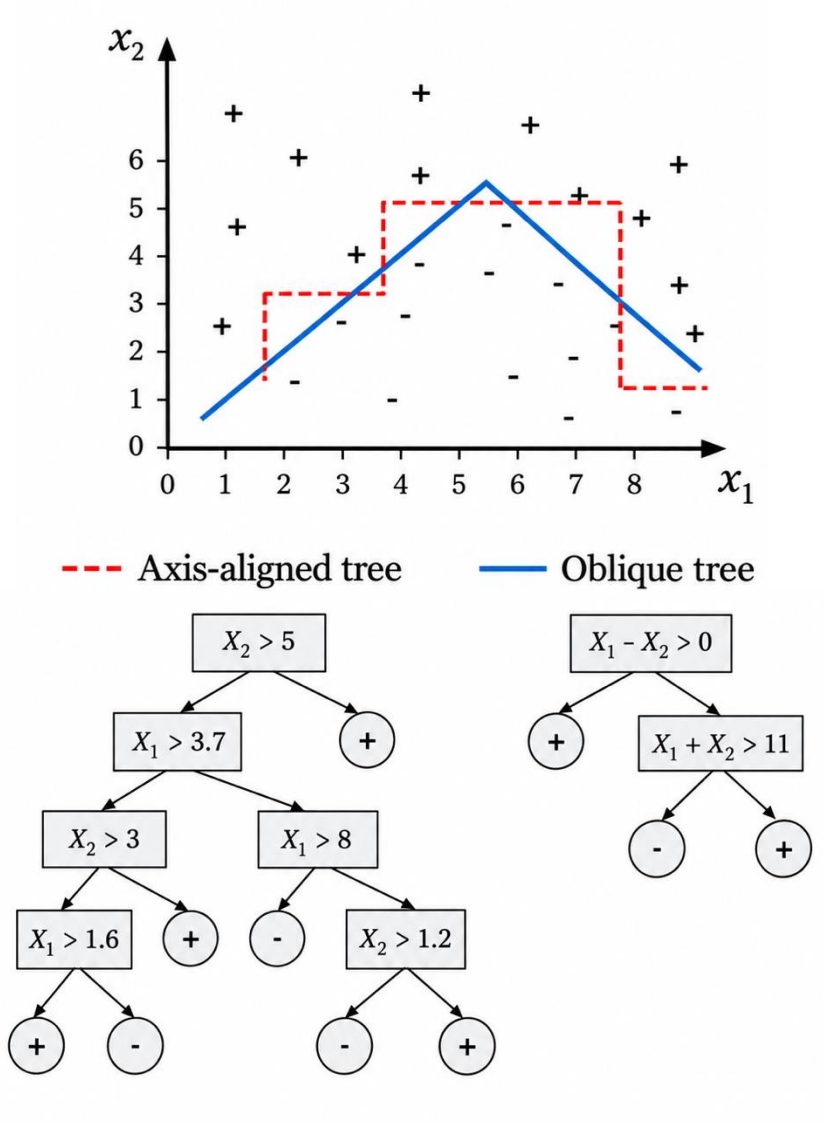
$$RMS_{hw} = \sqrt{\frac{r_{LUT}^2 + r_{DSP}^2 + r_{latency}^2}{3}}$$



Outlook

- Quintary BitNet: explore {-2, -1, 0, 1, 2} beyond binary and ternary precision.
- Broader model space: BitNet Transformers, oblique trees, and other low-latency architectures.
- Beyond jet tagging: various additional datasets, trigger-like tasks, and input representations.





Take-Home Messages

1. **BitNet scaling improves predictive performance over QKeras counterparts.**

2. **Low precision does not guarantee efficient hardware implementation.**

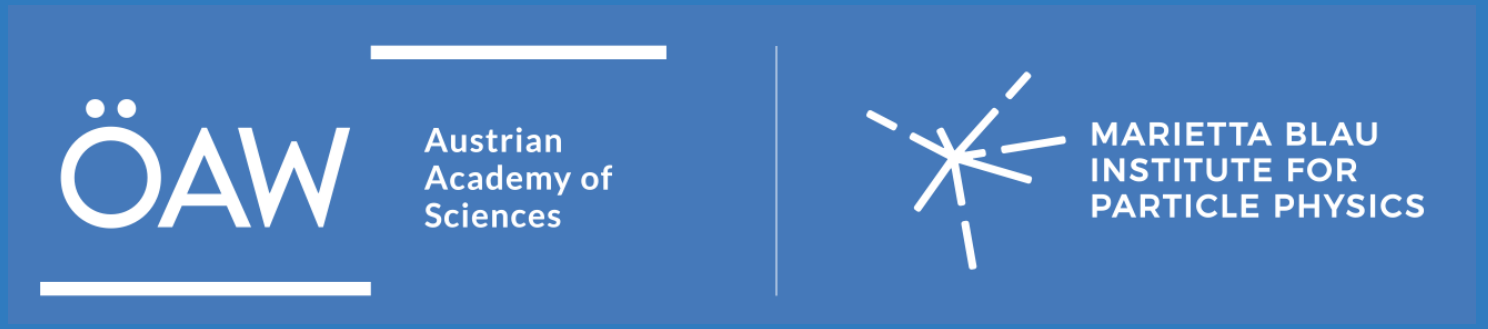
3. **There is no universal best model:**

QKeras 7-bit – best performance retention
HGQ – best neural-network resource efficiency
Unrolled BDT – lowest synthesised latency
BitNet-1.58 – strongest ultra-low-bit operating point

github.com/nairods/fastml_bitnet_benchmark

Thank you!

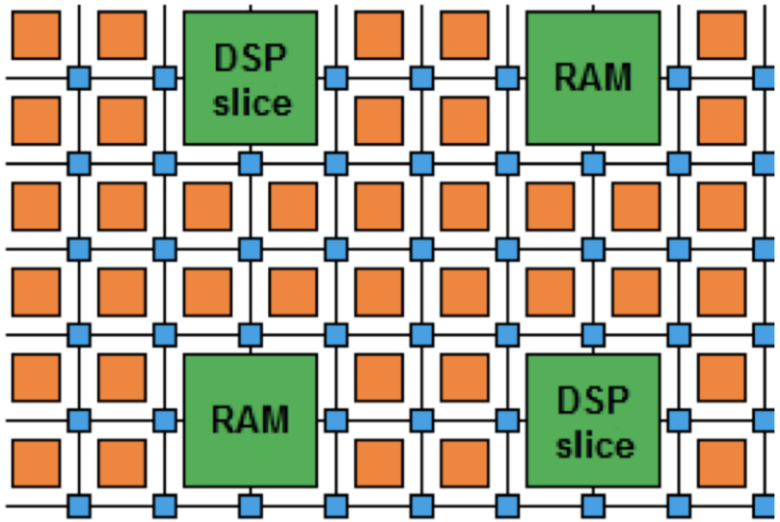

TU
WIEN


ÖAW
Austrian
Academy of
Sciences
MARIETTA BLAU
INSTITUTE FOR
PARTICLE PHYSICS


SCIES
4
FREE

Field Programmable Gate Array (FPGA)

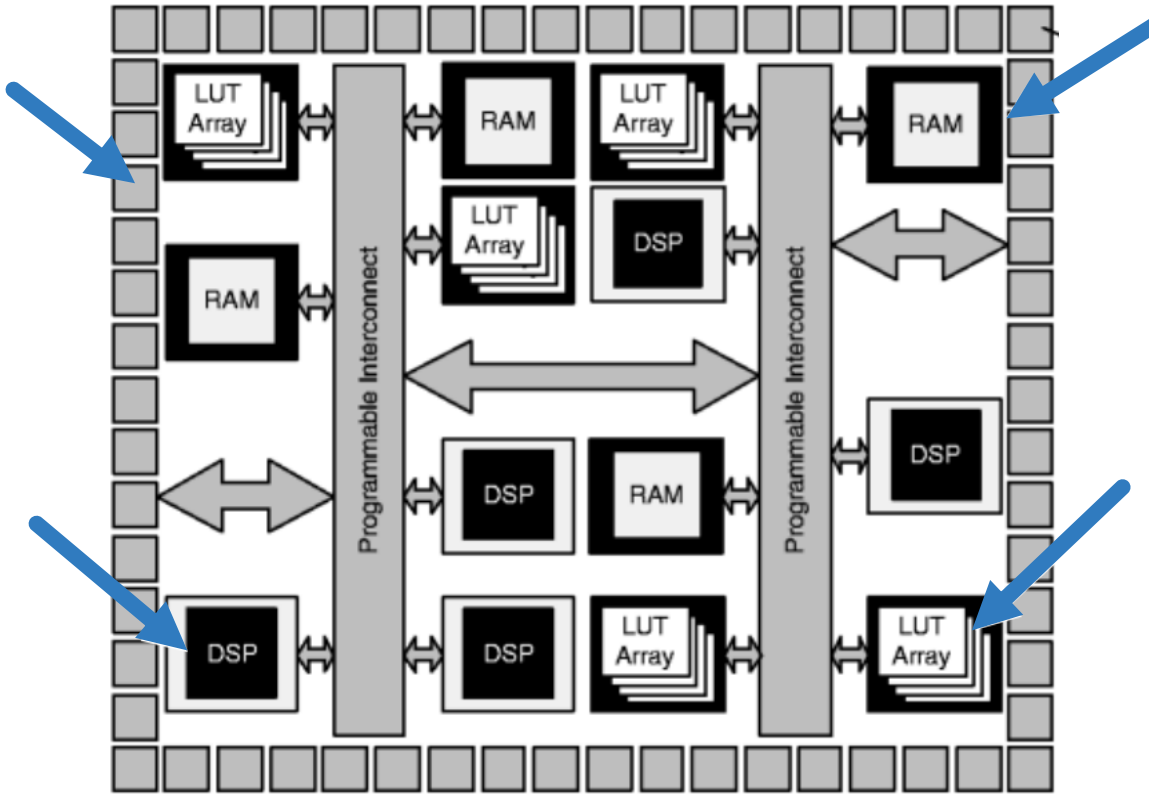
- Reprogrammable integrated circuits
- It contains many configurable building blocks connected by programmable routing.
- Low latency (parallelisation) & High throughput (pipelining).
- Originally for prototyping ASICs, but now also for embedded systems, real-time processing, and high-performance computing.



FPGA Logic Blocks

High-speed I/O and transceivers allow direct communication with external systems through multi-gigabit links.

Digital Signal Processors O(5,000) units: specialised arithmetic units for operations such as multiplication and accumulation.

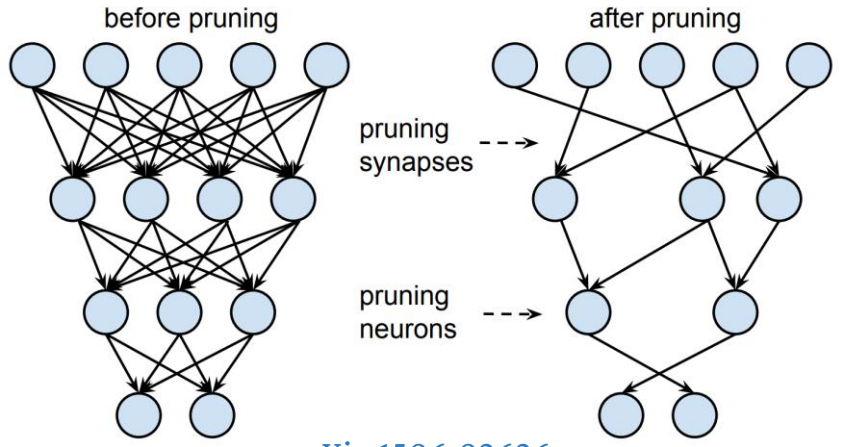


Block RAM (BRAM) O(2,000) units: provides fast on-chip memory for buffers, FIFOs, lookup tables, and local storage.

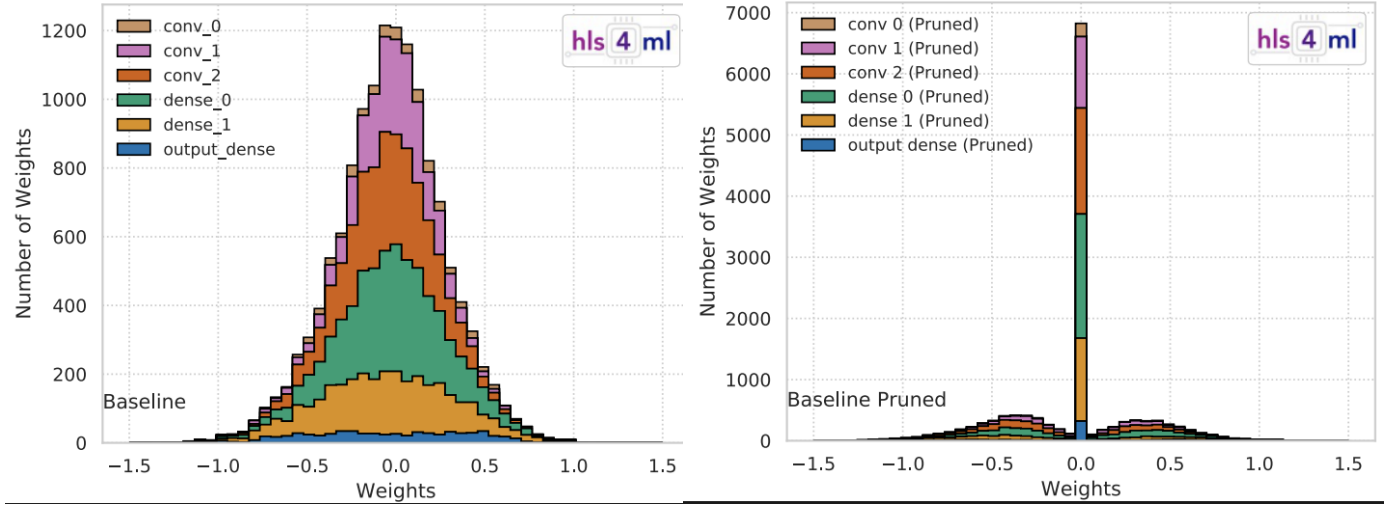
LUTs and flip-flops O(1) million units: implement control logic, simple arithmetic, and pipelined data paths.

Pruning Techniques

- Lottery Ticket Hypothesis:
  - There exists an optimal sparse network WITHIN each dense network.
- Unstructured pruning:
  - Remove individual weights/synapses; high sparsity, irregular pattern.
- Structured pruning:
  - Remove neurons/channels/layers; lower flexibility, better hardware efficiency.
- Scoring:
  - Magnitude- or gradient/sensitivity-based.



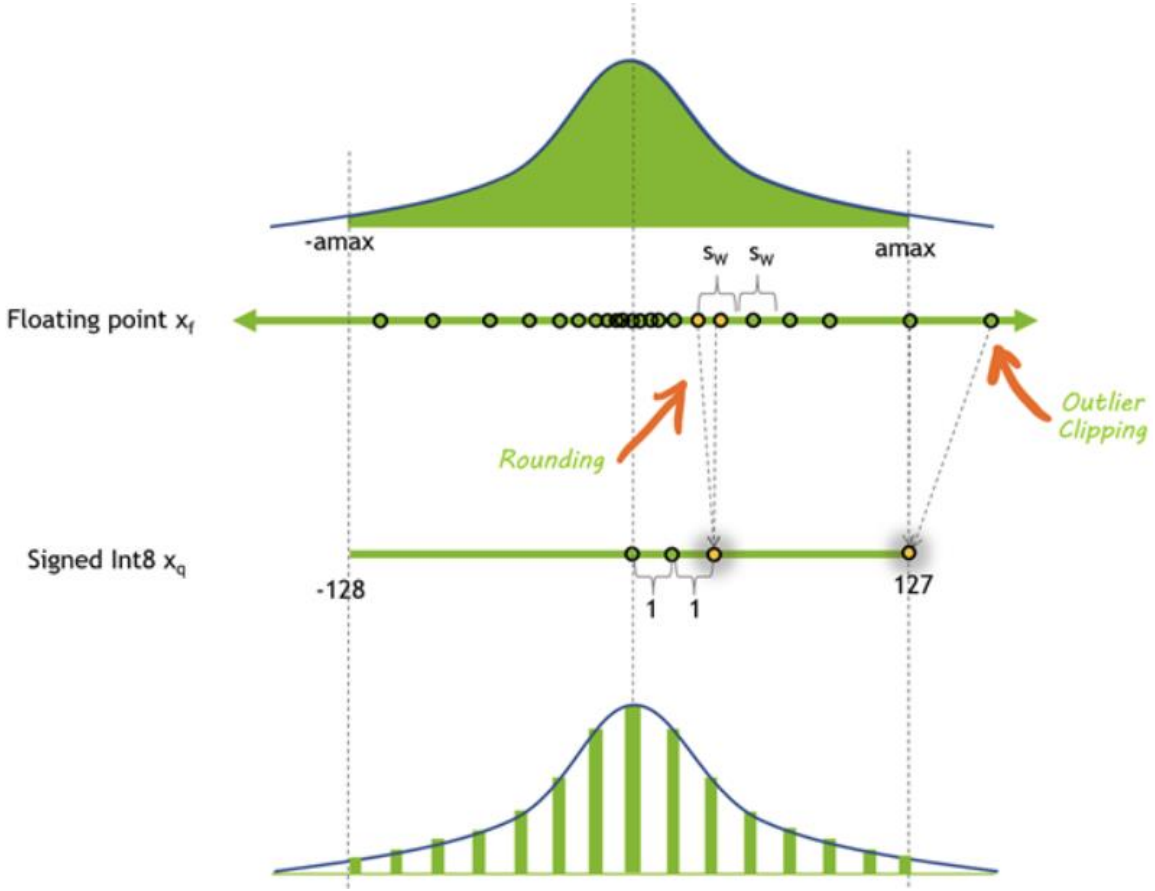
arXiv:1506.02626



Quantisation

- Floating points is too expensive for FPGA deployment in terms of memory, bandwidth, and resources:
  - FP32: 4B numbers in [-3.4e38, +3.4e38]
- Quantisation maps floating-point values to a smaller discrete set of numbers:
  - Signed Int8: 2^8=256 numbers in [-128,127]

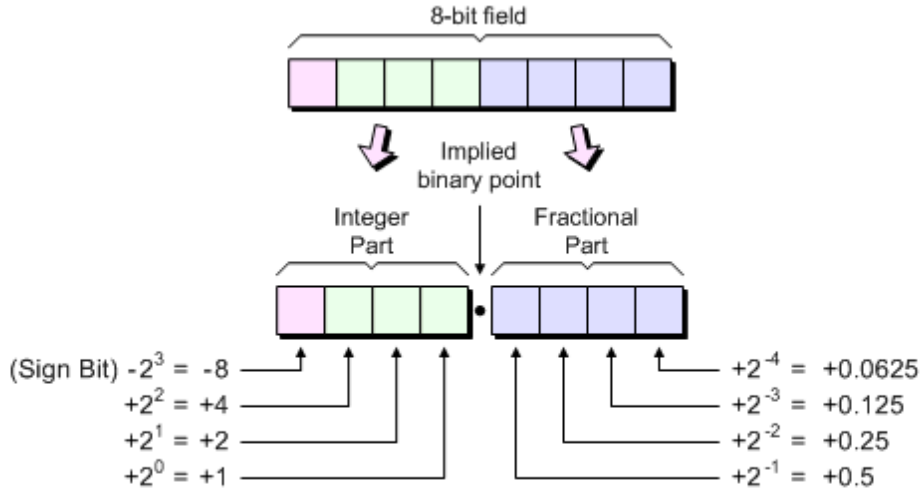
- $x_q = \text{Clip}(\text{Round}(\frac{x_f}{\text{scale}}))$



NVIDIA Developer

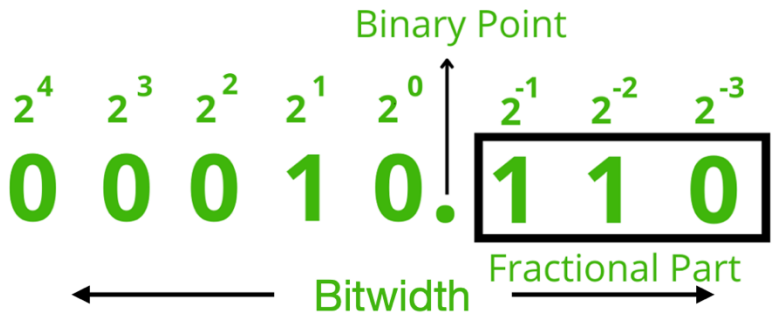
Fixed Point Precision ⟨W,I⟩

- Express fractions with integers:
  - W = bitwidth
  - I = integer bits
  - F = W− I (−1) = fractional bits
- Trade off:
  - range (integer bits) vs precision (fractional bits)
  - Range = $[-2^{I}, (2^{I}-2^{-F})]$
  - Precision = 2^{-F}
- E.g: Fixed_point<8,5>
  - $2^{4} \times 0 + 2^{3} \times 0 + 2^{2} \times 0 + 2^{1} \times 1 + 2^{0} \times 0 + 2^{-1} \times 1 + 2^{-2} \times 1 + 2^{-3} \times 0 = 2.75$

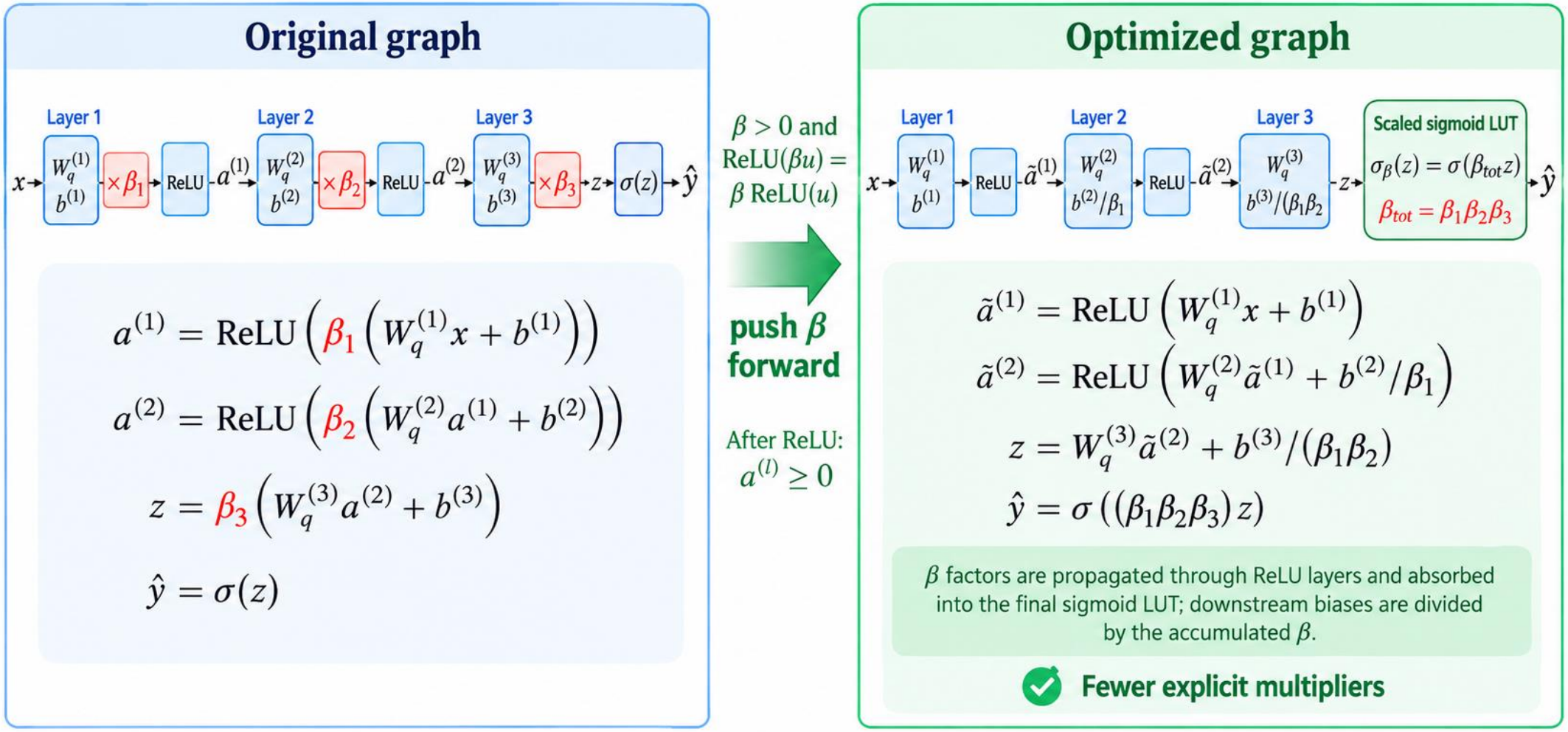


8-bit signed binary 4.4 fixed-point representation

The Databus

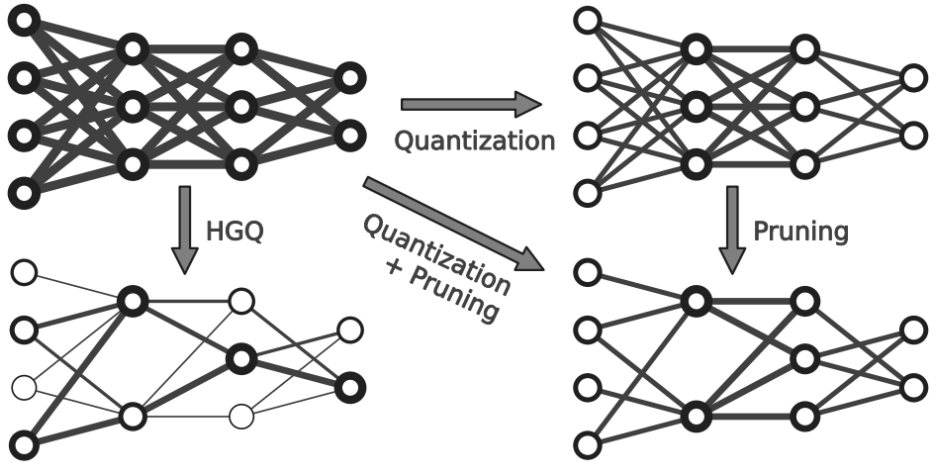


Folding β Through the Network



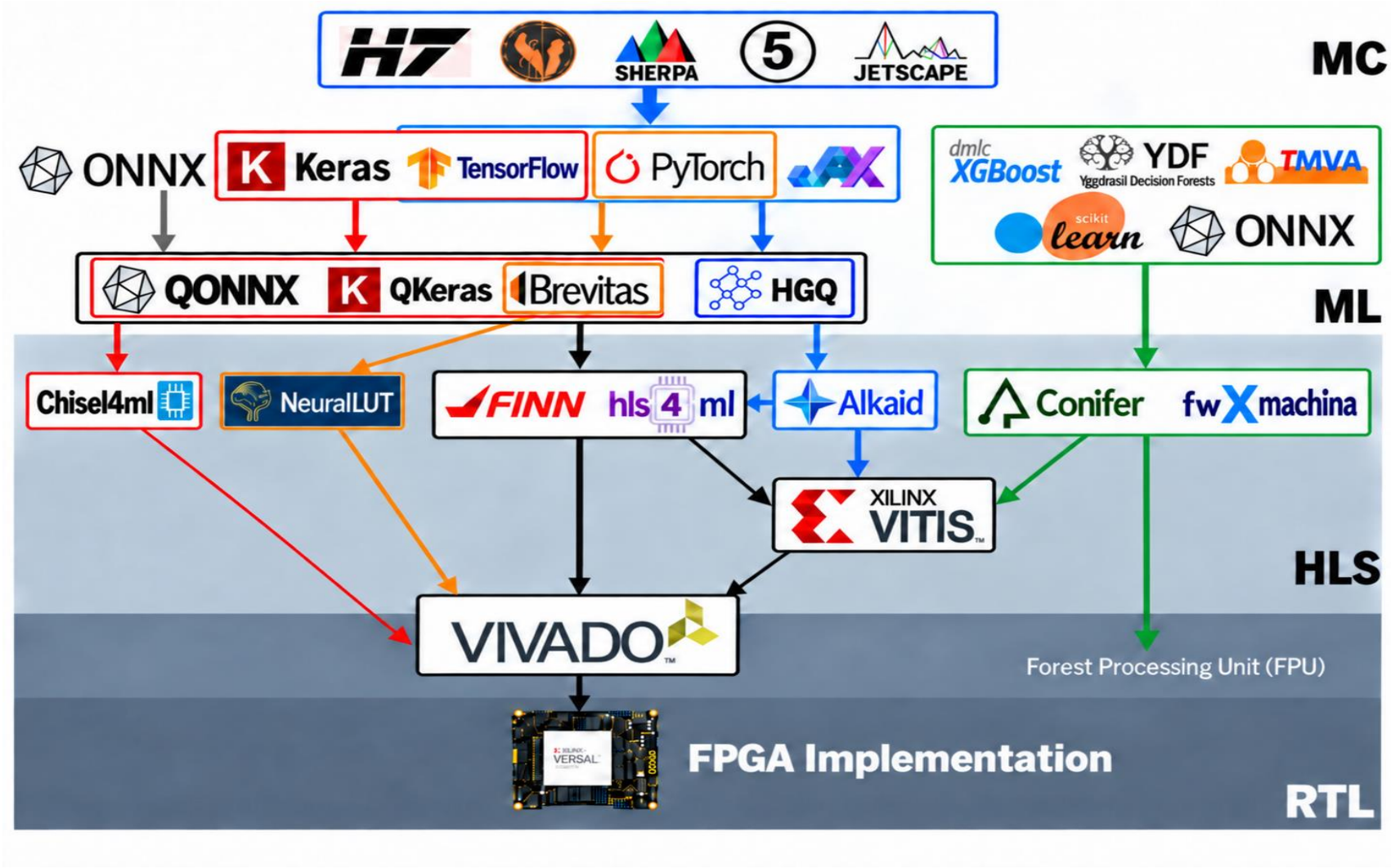
High Granularity Quantization (HGQ)

- HGQ learns precision per quantised node, rather than applying a uniform bit width across the network.
- Weights and activations receive locally optimised precisions, using more bits where they improve performance and fewer where they are unnecessary.
- Hardware-cost regularisation guides this allocation, favouring cheaper representations while preserving predictive performance.



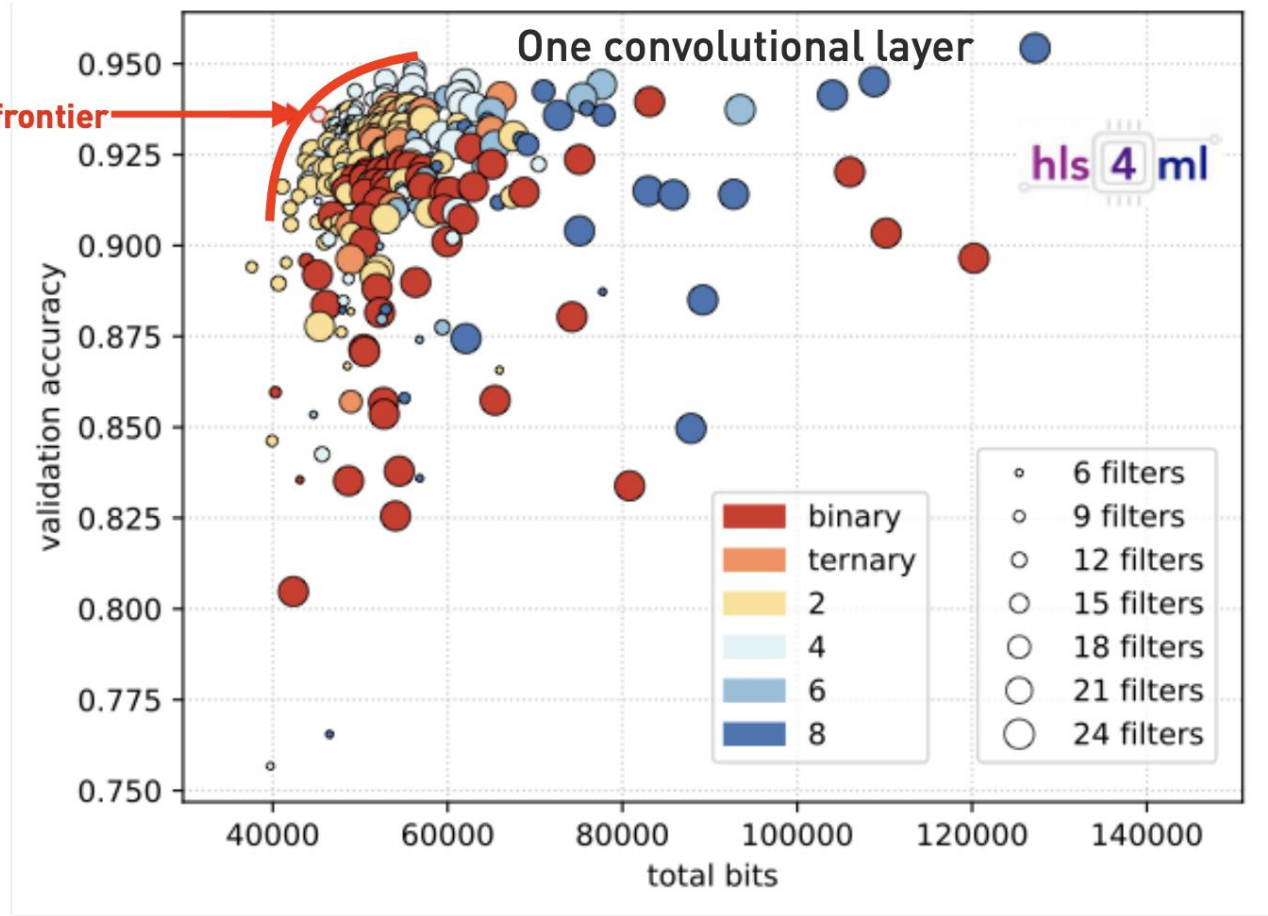
arXiv:2405.00645

FastML ToolChain



Evaluation of FastML Models

- Accuracy vs model size
- Total bits = (#parameters × bits per parameter)
- Set of solutions that give the best trade-offs between multiple objectives → Pareto front
- Highest accuracy for a given resource budget



arXiv:2101.05108

Model Comparison: q/g vs. W/Z/t

Model	Accuracy	AUC	Latency [ns]	LUT [10^3]	DSP
Dense MLP	*0.8618 ± 0.0003*	*0.9350 ± 0.0003*	*65.0 ± 0.0*	*182.9 ± 0.4*	*3651*
QKeras (7-bit)	**0.8611 ± 0.0004**	**0.9342 ± 0.0002**	48.4 ± 2.9	139.3 ± 1.5	981
HGQ	0.8538 ± 0.0012	0.9276 ± 0.0002	33.4 ± 2.9	**8.0 ± 0.8**	0
QKeras binary	0.8327 ± 0.0052	0.8981 ± 0.0004	108.4 ± 5.8	60.6 ± 1.2	0
QKeras ternary	0.8417 ± 0.0031	0.9103 ± 0.0002	73.4 ± 7.7	35.7 ± 3.6	0
BitNet binary	0.8445 ± 0.0007	0.9178 ± 0.0001	60.0 ± 0.0	87.3 ± 0.8	0
BitNet-1.58	0.8519 ± 0.0007	0.9254 ± 0.0008	50.0 ± 5.0	80.7 ± 2.1	0
BDT (unrolled)	0.8472 ± 0.0003	0.92075 ± 0.00003	**20.0 ± 0.0**	74.1 ± 0.1	0

Table shows the 64–32–32 architecture; values are mean ± SD across three seeds.