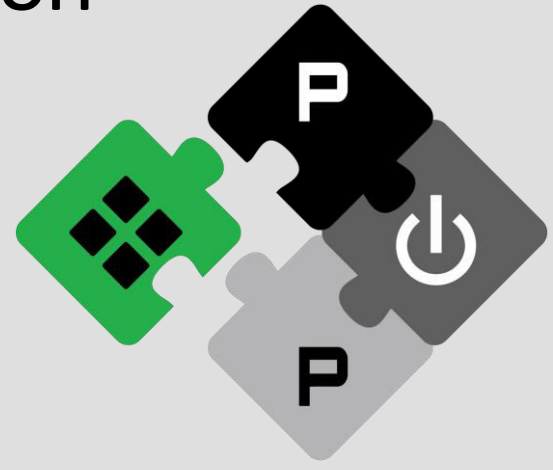
AICAS 2022 IEEE Advancing Technology for Humanity CAS ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA ETH zürich

Free Bits: Latency Optimization of Mixed-Precision Quantized Neural Networks on the Edge



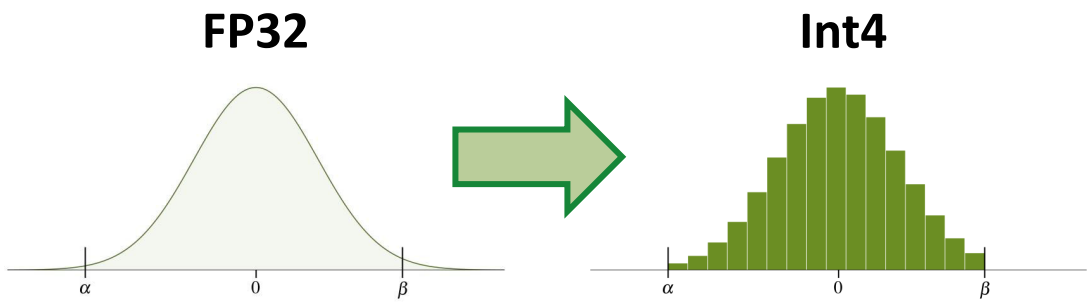
Georg Rutishauser[1] **georgr@iis.ee.ethz.ch**
Prof. Dr. Francesco Conti[2] f.conti@unibo.it
Prof. Dr. Luca Benini[1,2] lbenini@iis.ee.ethz.ch

[1]Integrated Systems Laboratory, ETH Zürich
[2]Università di Bologna

@pulp_platform
pulp-platform.org
youtube.com/pulp_platform

Edge AI: Quantization for Energy Efficiency

- **Insatiable hunger for data – number of IoT (sensor) nodes in use exploding: 83 Bn. devices by 2024! [1]**
- **On-device processing of collected data: Edge AI**
- **ML models tend to be large and compute-intensive**
  - **Particularly for difficult tasks**
- **Quantization:**



Less storage needed
Less memory needed
Lower inference latency

LOWER ENERGY/INF.!

[1]: https://www.juniperresearch.com/press/iot-connections-to-reach-83-bn-by-2024

Quantization Isn't Free Lunch!

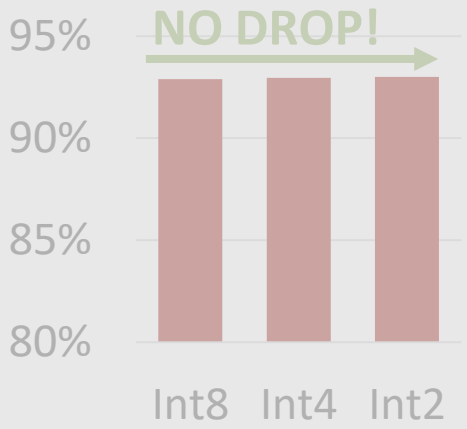
Q: What is the accuracy cost of quantization?
A: "It depends..."™ (On your task, network and quantization scheme)

"Easy" Task (CIFA

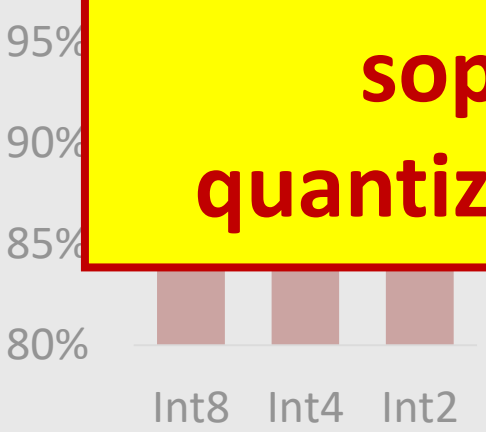
(ImageNet):

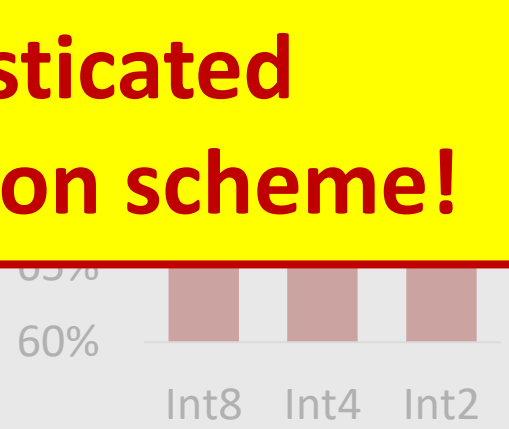
Want to solve "hard" tasks with "small" nets!
➔ We need a more sophisticated quantization scheme!

"Large" Net: VGG9
(~14M Param.)

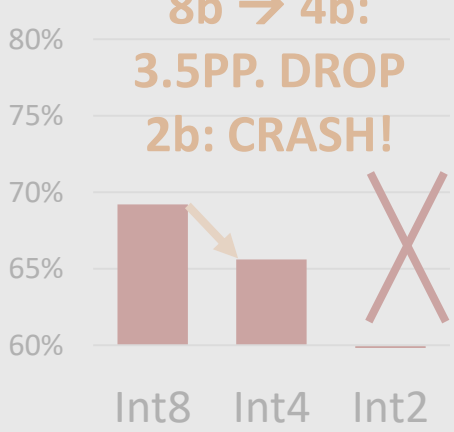


"Sr





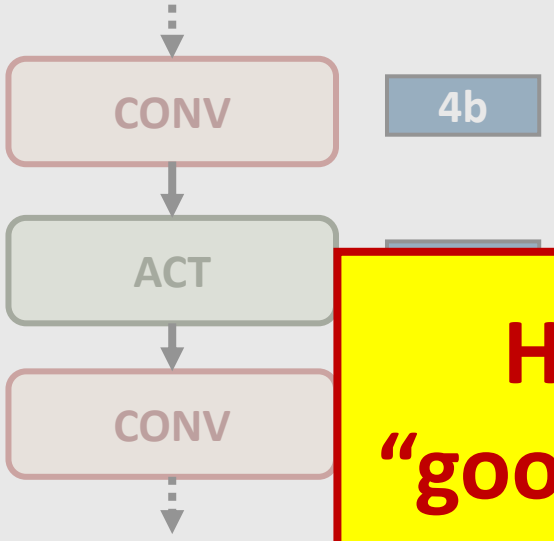
"Small" Net: MNv1
(~2.6M Param.)



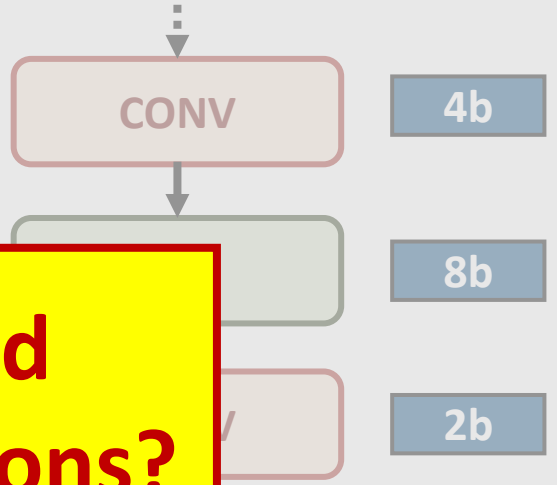
[2] Esser et al., "Learned Step Size Quantization"

Fine-Grained Quantization: Mixed-Precision Networks

Homogeneous Quantization (4b)



Mixed-Precision Quantization



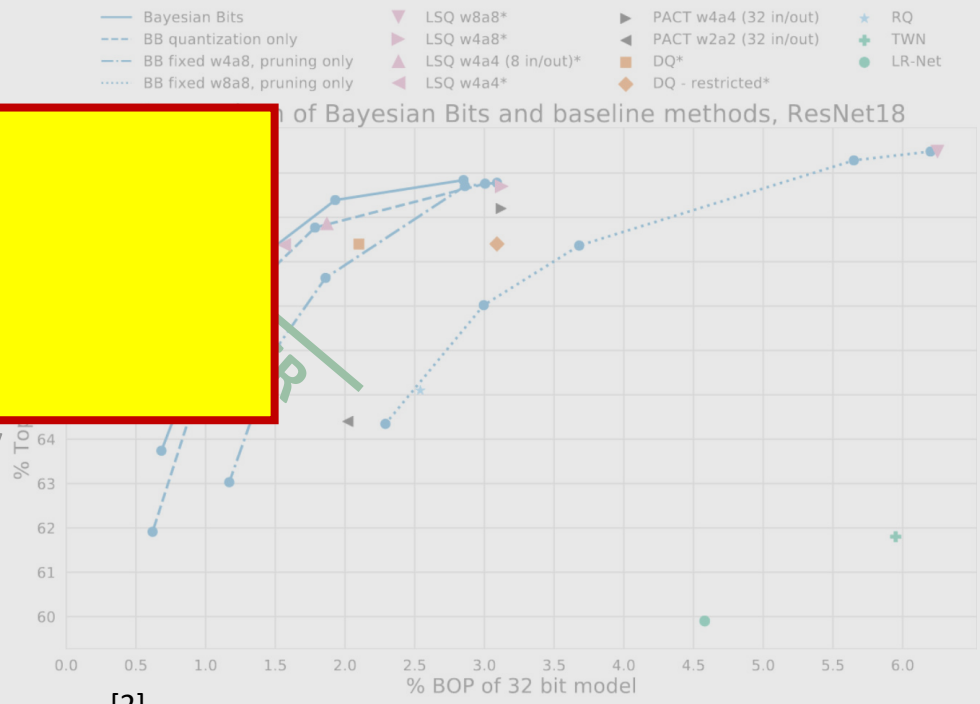
How can we find "good" configurations?

- Each weight and activation tensor is quantized to a different precision
- Maximize improvement in target metric (latency, model size, memory footprint), minimize accuracy drop
- But: Exponential number of configurations (MNv1: 2×10^{25})

Mixed-Precision Search:

- **Goal: map Pareto front of accuracy-cost trade-off**
  - Real-world cost: **latency**, model size, memory footprint...
  - **System inference energy: proportional to latency!**
- **Differentiable search**
  - E.g.: Bayesian Bits [1]
  - Idea: Regularize task los
  - Commonly uses "binary operations" (BOPs) as cost proxy
  - Layer BOPs: $\#BOPs_l = b_{in,l} \times b_{wt,l} \times \#MACs_l$
    - BOPs = "Precision per MAC"

JOB DONE?
NO!

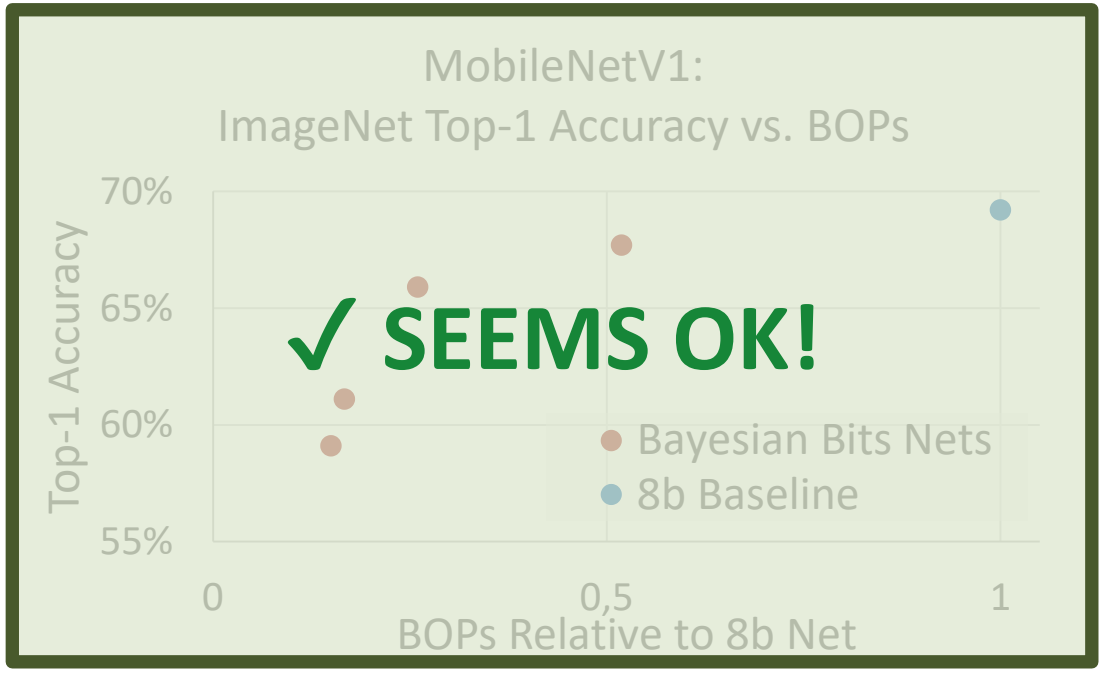


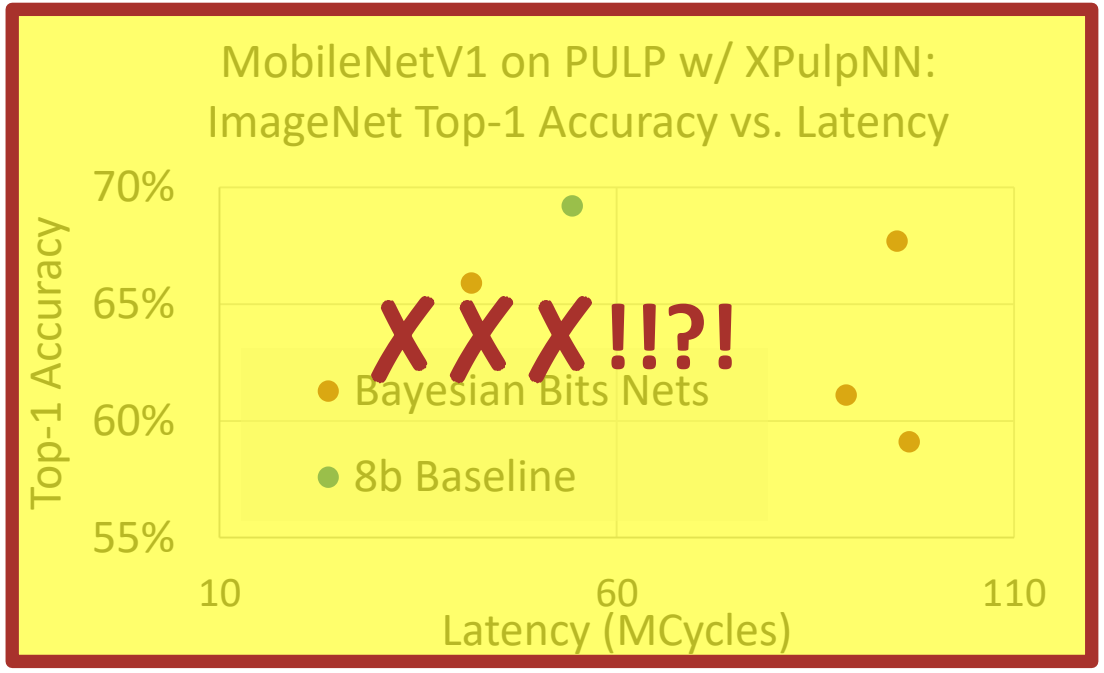
[2]

[1] M. van Baalen et al., Bayesian Bits: Unifying Quantization and Pruning

BOPs Don’t Correspond to Inference Latency

- **Let’s find & deploy a mixed-precision network to mixed-precision hardware!**
  - Algorithm: Bayesian Bits

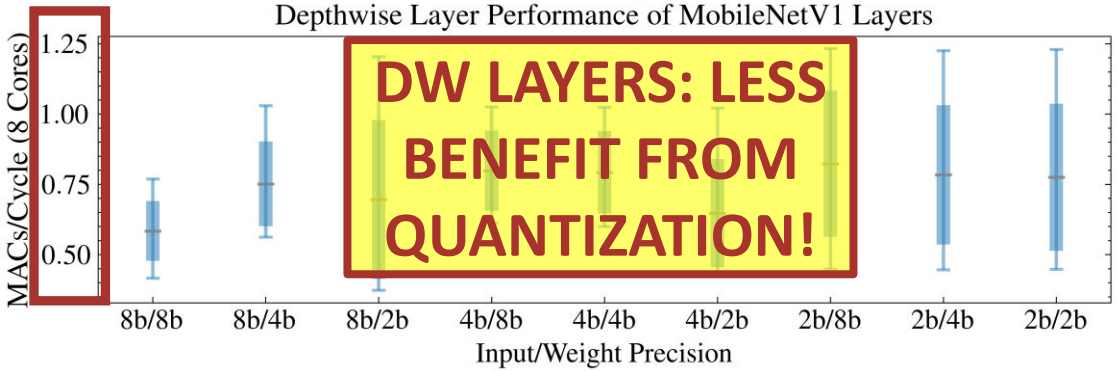


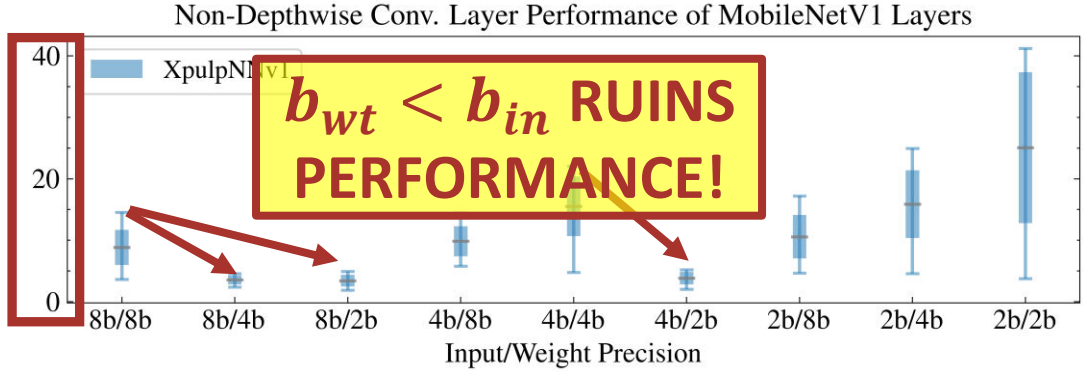


On real systems, latency is not proportional to BOPs!

The BOP-Latency Gap

Profiling ALL MNv1 layers in ALL supported precisions (~1-2h):





Counterintuitive: some layers run faster in higher precisions!

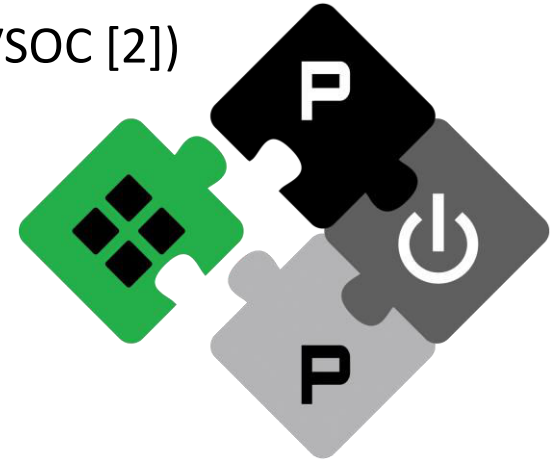
Free Bits

- **Increasing their precision ("Free Bits") gives us:**
  - Lower Latency
  - Higher Accuracy
- **Setup:**
  - Assume layer-wise execution → $L_{net} \approx \sum_l L_l$
  - Given network architecture and target platform, find latency-optimized MP configurations

FREE BITS:
Increase a layer's precision if it decreases latency

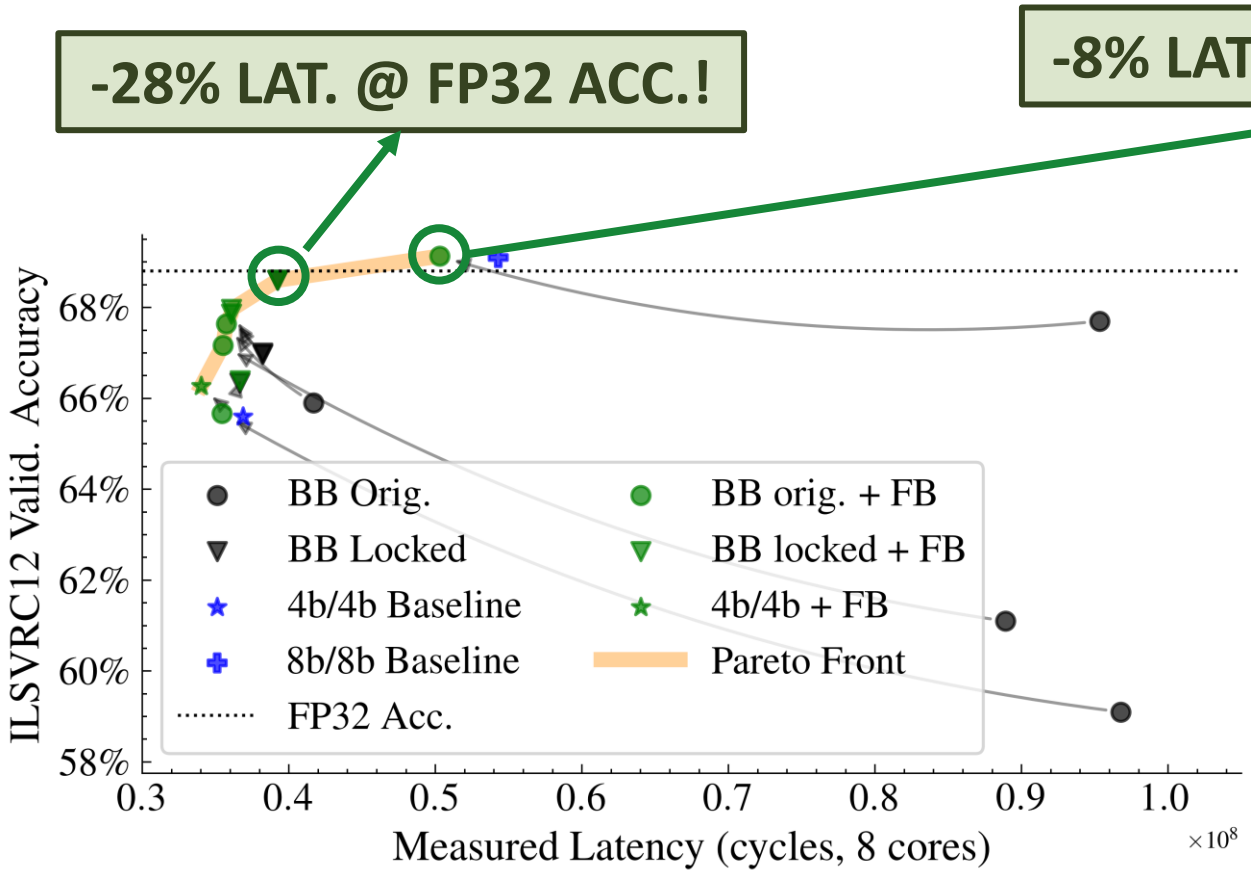
Experiments

- **Networks: MobileNetV1-224-0.75, MobileNetV2-224-1.0**
- **Task: ImageNet, 1000-class classification**
- **Target platforms: PULP systems with different ISA Extensions**
  - **P**arallel **U**ltra **L**ow **P**ower: Multicore RISC-V MCUs with specialized DSP ISA extensions
  - Computation on **cluster** of 8 RISC-V cores
  - 512 KiB main L2 memory, 64 KiB cluster L1 memory
  - Networks mapped with DORY [1] and run on a cycle-accurate simulator (GVSOC [2])
  - 3 RISC-V ISA Extensions (2 shown in this presentation):
    - XPulpV2 – SIMD 8-bit arithmetic, post-increment load/store, ...
    - XPulpNN – XPulpV2 + SIMD sub-byte operations on equal-precision operands
  - 2 variants of Bayesian Bits
    - Vanilla – as in original paper
    - Locked – weight and activation precisions constrained to be equal



[1] https://github.com/pulp-platform/dory
[2] https://github.com/pulp-platform/pulp-sdk

Results: MobileNetV1 on XPulpNN



- Locked precisions:
  Dominant wrt. baselines
- Free Bits heuristic:
  - ✓ **Improves all but 2 configs**
  - ✓ **Improves 4b/4b baseline**
  - ✓ **6/8 configurations are Pareto-dominant!**

MobileNetV1 on Different ISA Extensions

	XPulpNNV1: SW Unpacking		XPulpV2: 8-bit SIMD only	
Network	**Lat. vs. 8b**	**Accuracy**	**Lat. vs. 8b**	**Accuracy**
Baseline: 8b	+0%	69.1%	+0%	69.1%
Baseline: 4b				
4b + Free Bits				
<0.5 pp. acc. drop				

Even 8-bit-only ISA profits from mixed precision!

Conclusion: Use Your Empirical Platform Knowledge!

- **BOPs are not a sufficient proxy for complexity on real platforms**
- **Free Bits:**
  - Start with DNAS-based, HW-agnostic precision search
  - **Profiling-based** heuristic: Increase precision where it benefits latency
  - **Improves accuracy *and* latency** of MP networks on PULP systems
  - Applicable to **any platform**
- **Benchmarks: MobileNetV1 and MobileNetV2 on PULP**

  ✓-28% latency @ FP32 accuracy for MobileNetV1 w/ XPulpNN ISA

  ✓Latency improvement on 8-bit only ISA (XPulpV2)

  ✓Homogeneously quantized networks (8b, 4b) dominated

PULP Platform

Open Source Hardware, the way it should be!

Georg Rutishauser georgr@iis.ee.ethz.ch
Prof. Dr. Francesco Conti f.conti@unibo.it
Prof. Dr. Luca Benini lbenini@iis.ee.ethz.ch

Institut für Integrierte Systeme – ETH Zürich
Gloriastrasse 35
Zürich, Switzerland

DEI – Universitá di Bologna
Viale del Risorgimento 2
Bologna, Italy

@pulp_platform

pulp-platform.org

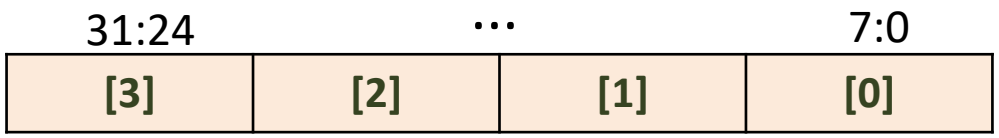
youtube.com/pulp_platform

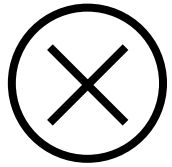
ETH zürich ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Unpacking Mismatched Operands

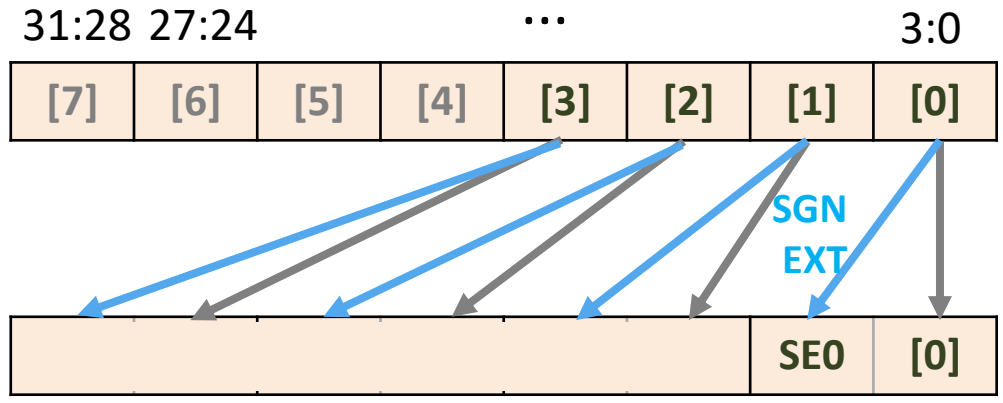
- Packed-SIMD operations on operands of different precisions: Lower-precision operand must be “unpacked” to to the higher precision
- Example: MAC of 8-bit and 4-bit operands

Operand A: 4x8 bits





Operand B: 8x4 bits

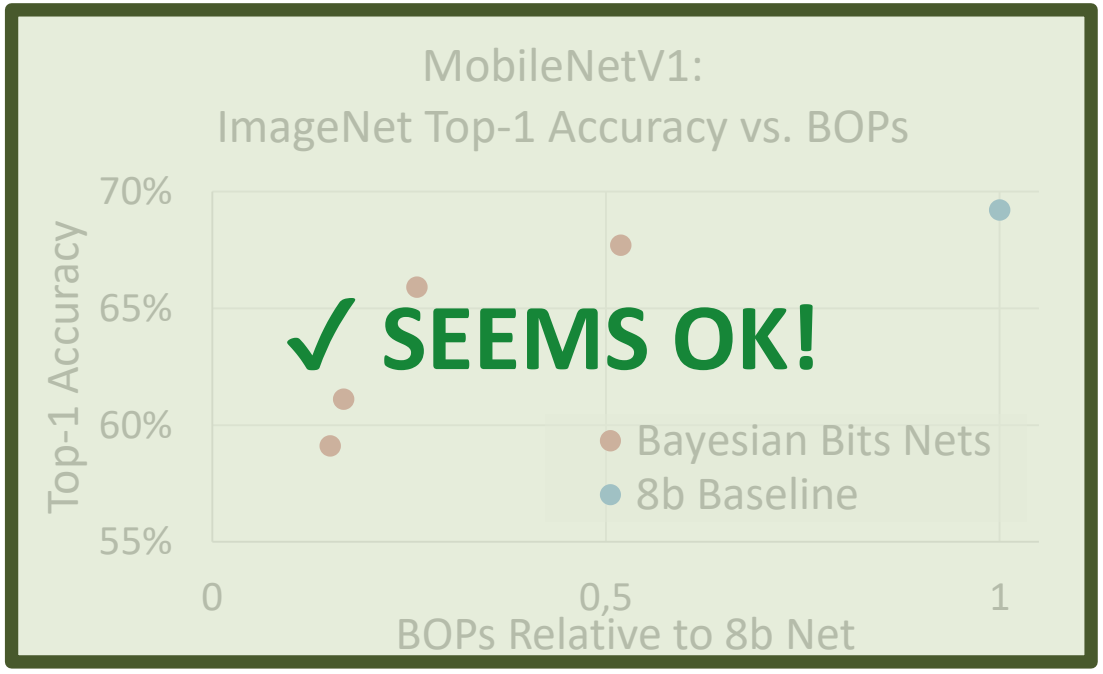


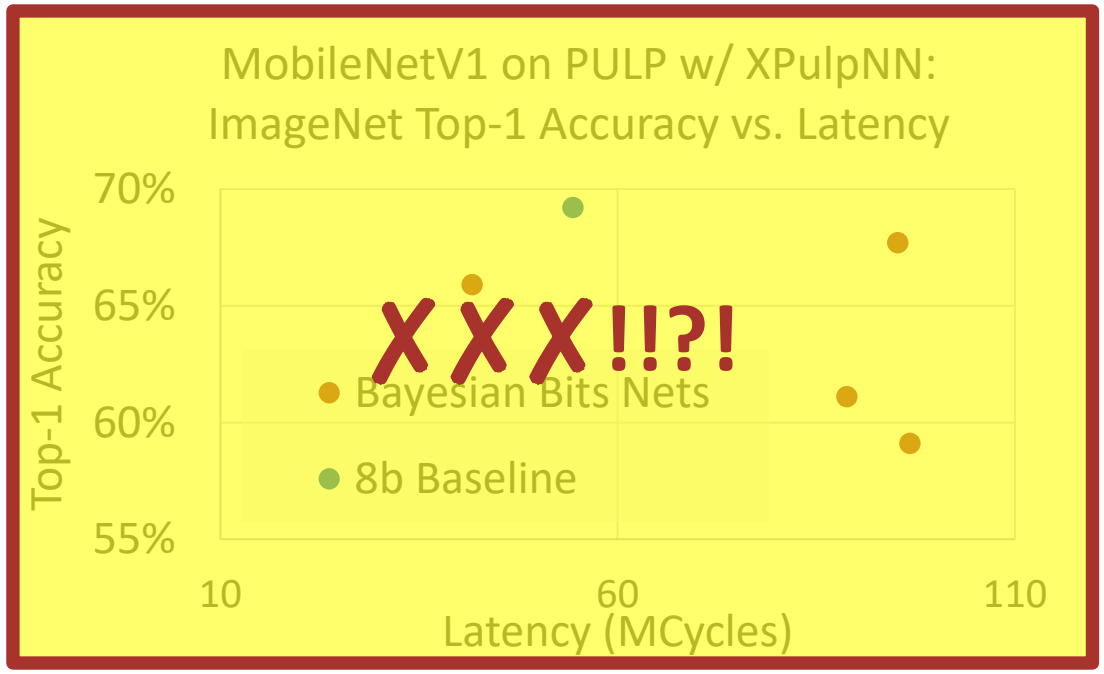
Unpacked B: 4x8 bits – ready to MAC with A!

SOFTWARE UNPACKING → RUNTIME OVERHEAD!

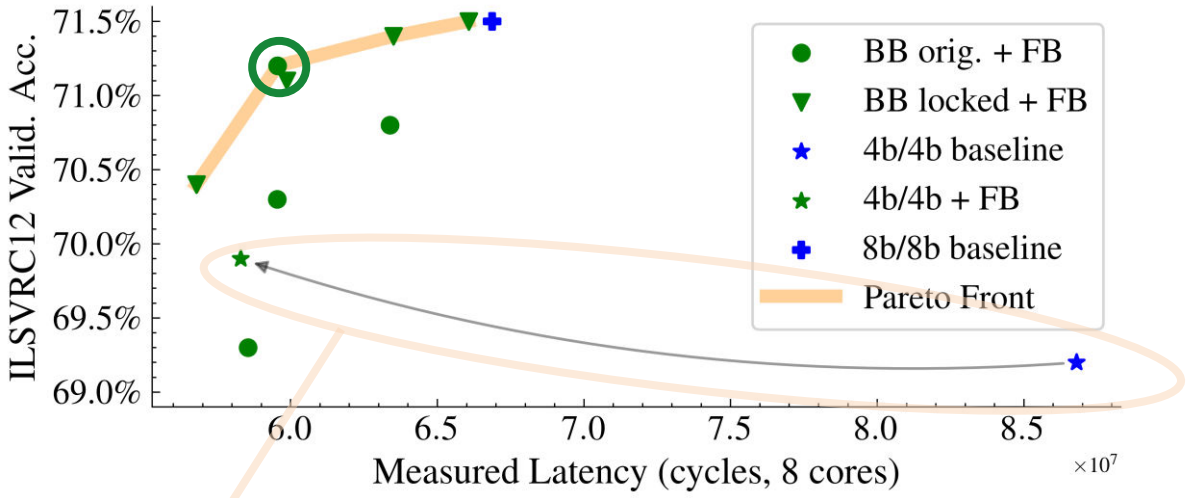
BOPs Don’t Correspond to Inference Latency

- **Our target platforms: PULP systems**
  - **P**arallel **U**ltra **L**ow **P**ower – multi-core, near-threshold MCUs
  - Computation on **cluster:** 8× 32-bit RISC-V cores with custom ISA Extensions
  - **XPulpNN:** vectorized sub-byte MAC operations – $16 \times 2b / 8 \times 4b / 4 \times 8b$ MACs/cycle/core
  - Let’s try to find a good mixed-precision MobileNetV1 with Bayesian Bits!





Results: MobileNetV2 on XPulpNN



Q: Why is the 4b/4b baseline so slow?
A: Outputs of adder nodes are always 8b
→ precision mismatch in next layer

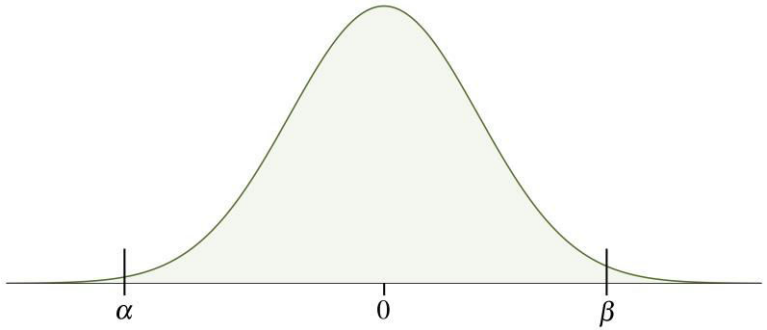
- ✓ **Free Bits dominates baselines**
- ✓ **-11% latency @ -0.3 pp. accuracy**

Full Results: MNv1, MNv2/XPV2, XPNNv1, XPNNv2

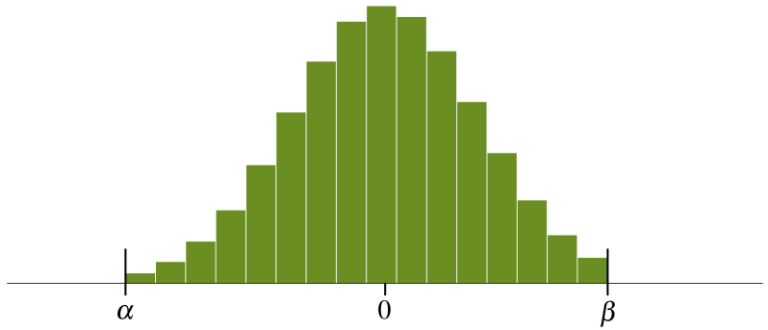
Acc. Margin	ISA	MobileNetV1		MobileNetV2	
		Lat. vs. 8b	Acc.	Lat. vs. 8b	Acc.
8b Baseline	*all*	+0%	69.1%	+0%	71.5%
0.5 pp.	XPv2	−5.5%	69.3%	−3.4%	71.0%
	XPNNv1	−27.9%	68.6%	−10.9%	71.2%
	XPNNv2	−28.6%	68.6%	−15.3%	71.0%
1.5 pp.	XPv2	−5.5%	69.3%	−6.3%	70.7%
	XPNNv1	−34.4%	67.6%	−15.1%	70.4%
	XPNNv2	−35.1%	67.6%	−15.3%	71.0%
4b + FB	XPv2	−3.5%	67.7%	−7.7%	70.9%
	XPNNv1	−37.1%	66.3%	−12.8%	69.9%
	XPNNv2	−39.8%	66.6%	−25.7%	69.6%
4b Baseline	XPv2	+49.9%	65.6%	+37.0%	69.3%
	XPNNv1	−32.3%	65.6%	+48.9%	69.3%
	XPNNv2	−38.3%	65.6%	−23.4%	69.3%

Integer Quantization can Help…

FP32



Int4

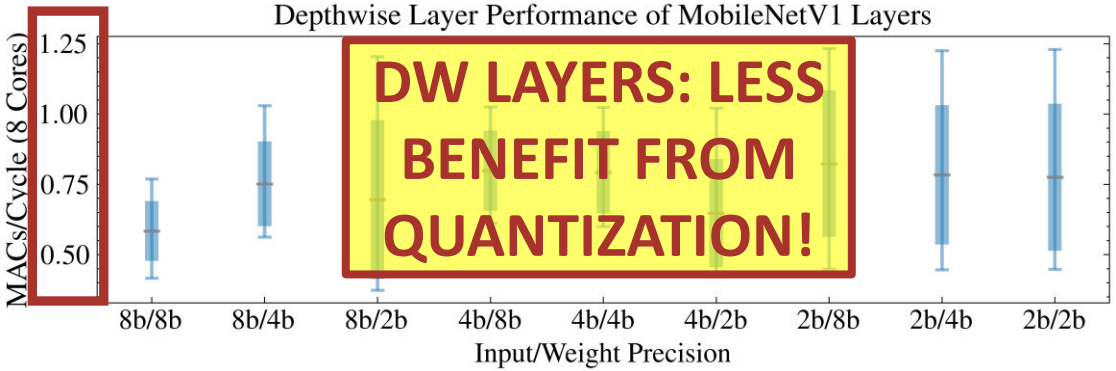


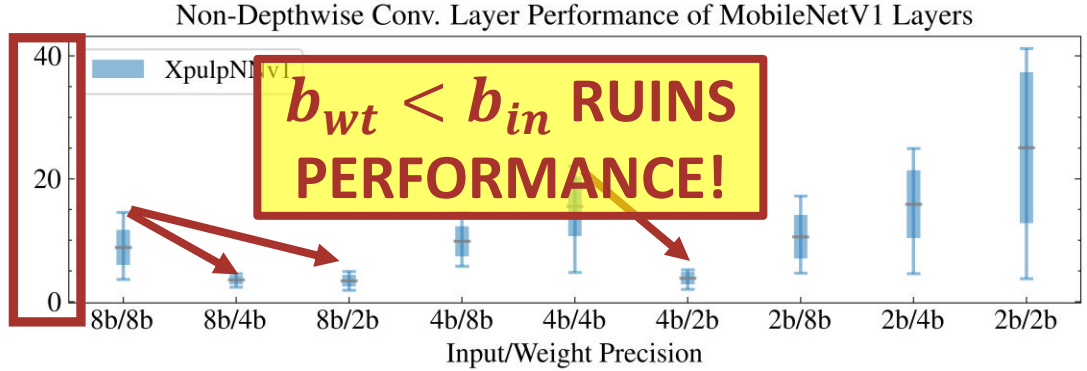
- Represent weights & activations as low-bitwidth integers
- Run inference using integer arithmetic only
- Benefits:
  - Smaller model size → Less storage needed
  - Smaller activation size → Less memory needed
  - More ops/cycle → Lower inference latency

LOWER ENERGY/INF.!

The BOP-Latency Gap

- **Profiling ALL MNv1 layers in ALL supported precisions (~1-2h):**





- **On real systems, latency is not proportional to BOPs!**
  - HW constraints: equal-precision operands required
    - Overhead to unpack lower-precision operands
    - E.g. NVidia INT4 Tensor cores, **XPulpNN**
  - SW non-idealities:
    - Some kernels may see limited benefit from low-precision operands
    - Performance depends on tiling strategy

Edge AI: Energy Efficiency is Everything

- **Insatiable hunger for data – number of IoT (sensor) nodes in use exploding: 83 Bn. devices by 2024! [1]**
- **On-device processing of collected data: Edge AI**
- **This scaling can only be sustained if nodes become even more:**
  - Cheap → MCU-class systems
  - Powerful → Optimization for ML algorithms
  - Versatile → battery-powered → **energy efficiency is everything!**
- **ML models tend to be large and compute-intensive**
  - **Particularly for difficult tasks**

ML-
opt

[1]: https://www.juniperresearch.com/press/iot-connections-to-reach-83-bn-by-2024