Efficient Deployment of Spiking Neural Networks on SpiNNaker2 for DVS Gesture Recognition Using NIR

Sirine Arfa, Bernhard Vogginger, Chen Liu, Johannes Partzsch, Mark Schone and Christian Mayr

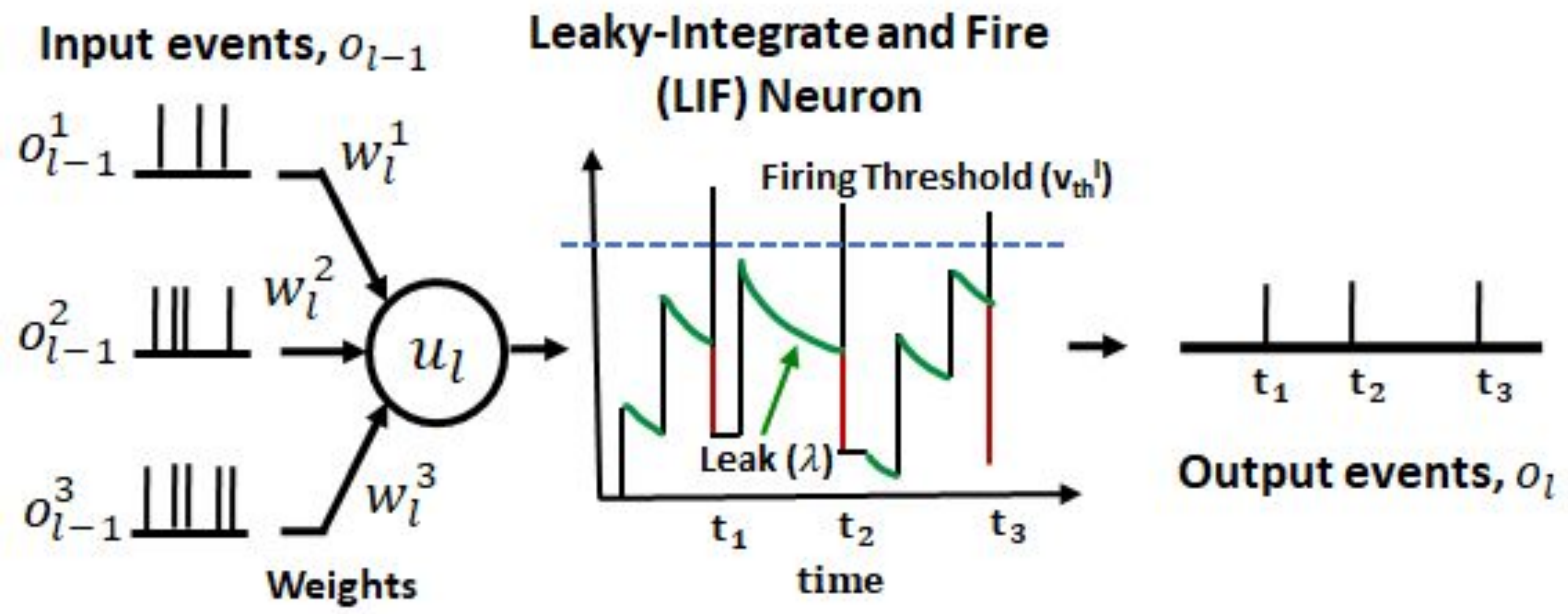
Chair of Highly-Parallel VLSI-Systems and Neuro-Microelectronics
Technical University of Dresden, Germany

25.03.2025

The Challenge of Neuromorphic AI

- SNNs promise **energy-efficient computing**, but hardware constraints make their deployment **challenging**.
- Quantization is essential, but current methods **often overlook neuron thresholds**.
- Can we reduce memory usage by applying quantization on SNNs while maintaining the accuracy?

*



*Kosta, A., & Roy, K. (2022). Adaptive-SpikeNet: Event-based Optical Flow Estimation using Spiking Neural Networks with Learnable Neuronal Dynamics. arXiv (09 2022). *arXiv preprint arXiv.2209.11741*

Fixed Precision for SNNs: A Necessary Trade-off

Why Fixed Precision?
Neuromorphic hardware demands lower bit-widths (e.g., int8) for efficiency.

The Challenge:
Lower precision can hurt model accuracy — it's a trade-off.

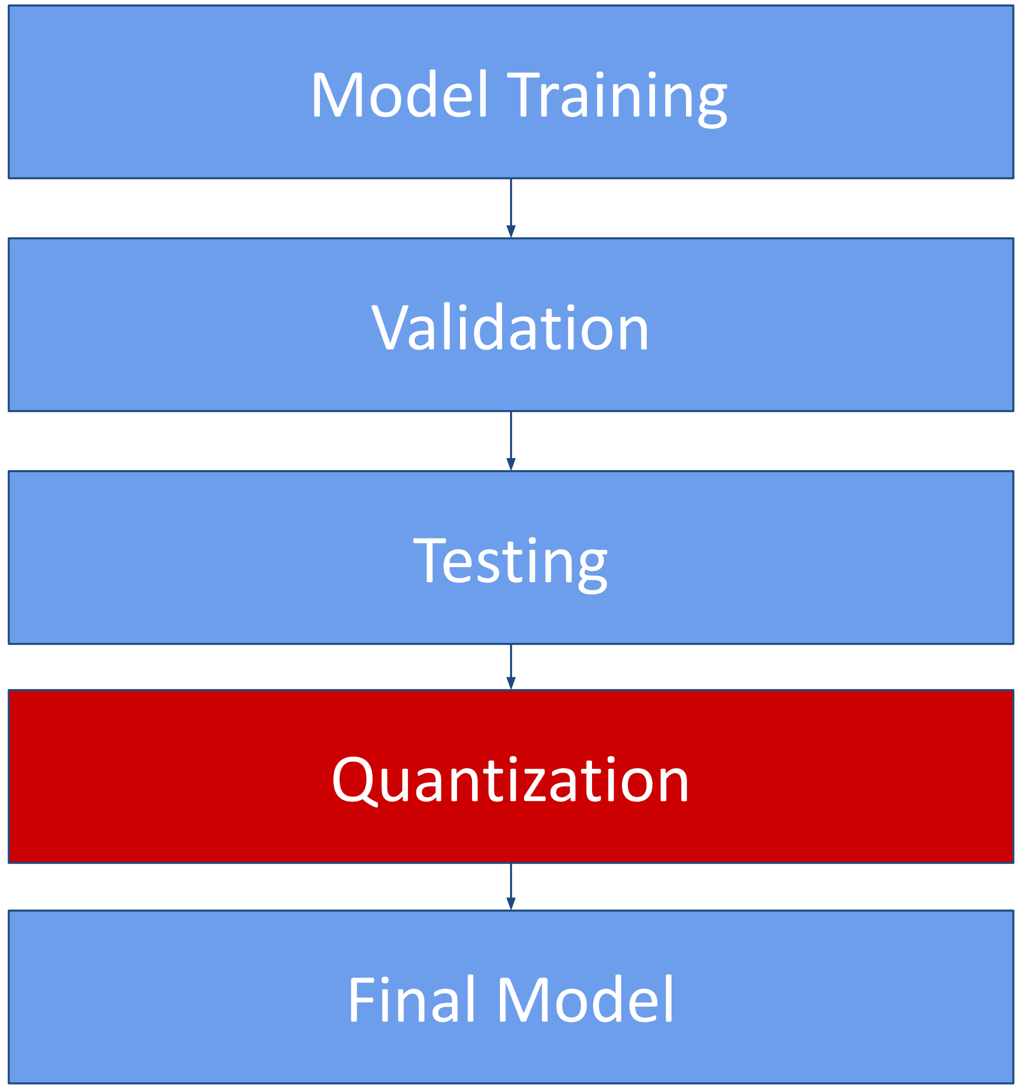
The Opportunity:
With smart techniques like pruning and threshold tuning, we can stay efficient **without** sacrificing performance.

Our Goal:
Find the sweet spot between memory savings and model accuracy.

Balancing Accuracy and Efficiency in Quantized SNNs

Post-Training Quantization (PTQ) vs. Quantization Aware Training (QAT)

- **PTQ**: Simpler, faster, but loses accuracy.

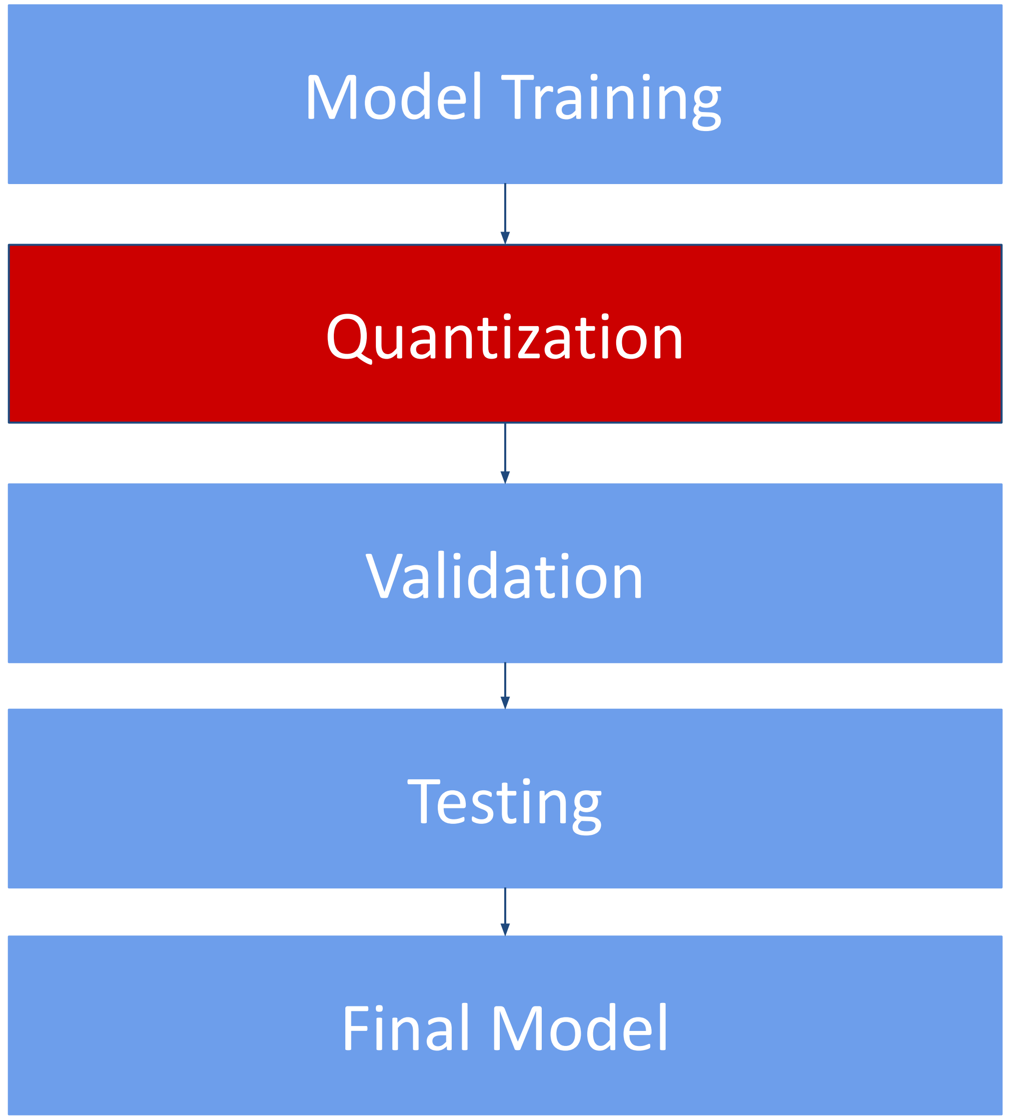


- Train model in full precision
- Quantize weights only at runtime

Balancing Accuracy and Efficiency in Quantized SNNs

Post-Training Quantization (PTQ) vs. Quantization Aware Training (QAT)

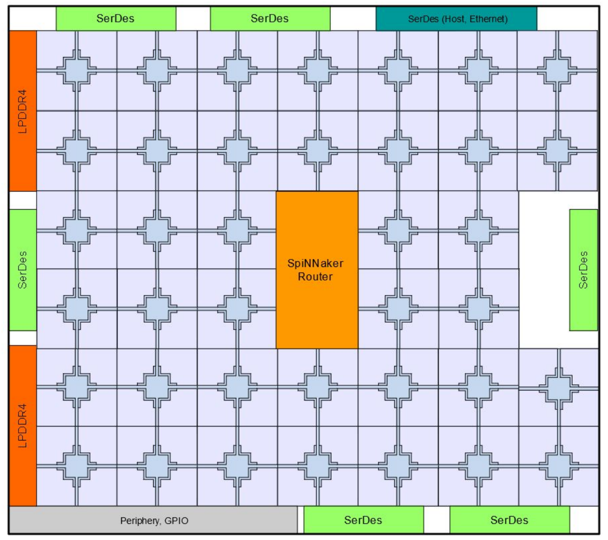
- **QAT**: Trains with quantization, achieving better accuracy but with higher training cost.



Key Question: Beyond precision reduction, what other considerations are critical when quantizing SNNs for effective implementation on neuromorphic systems?

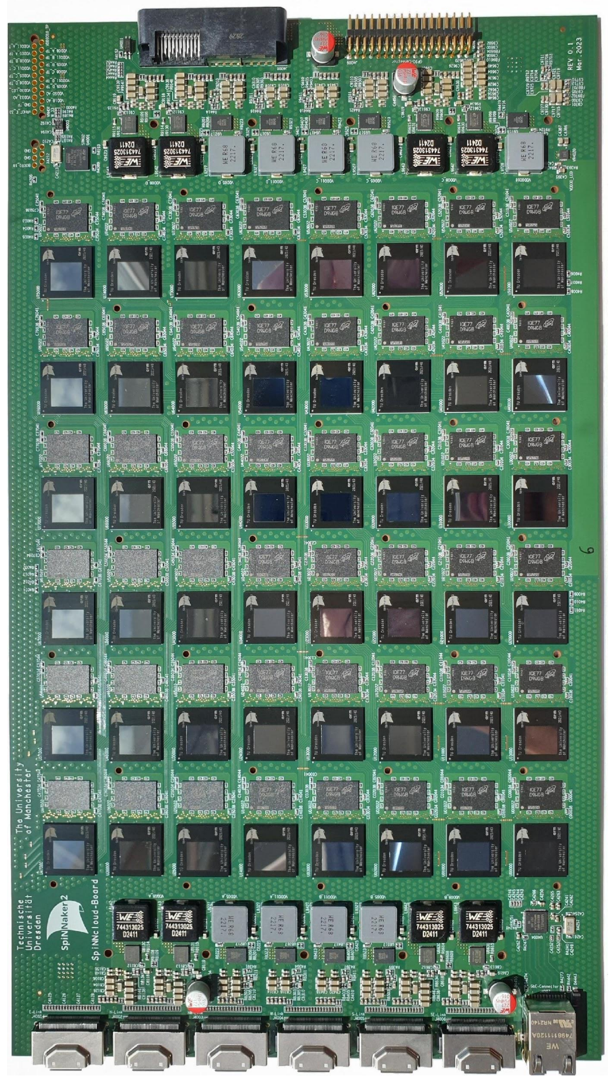
The SpiNNcloud Neuromorphic Supercomputer
World‘s largest brain-inspired computer

SpiNNaker2 Chip



- 152 processing elements with low-power ARM Cores and DNN accelerators
- SpiNNaker router & 6 chip-to-chip interfaces for event-based communication
- 22FDX CMOS by GLOBALFOUNDRIES
- 0.5 V operation using Raciycs ABB IP

SpiNNcloud Board



48 SpiNNaker2 chips
with 2 GB DRAM each

Card frame

With 18 boards and water cooling

The „SpiNNcloud“

5 million core machine at TU Dresden:
8 racks with 5 card frames each.

The SpiNNcloud Neuromorphic Supercomputer
Software Stack

NIR

Data flow graphs
Event-based computing

ONNX

Spiking neural networks

Generic Compute

Deep learning

S2 FrontendCommon

S2 Runtime

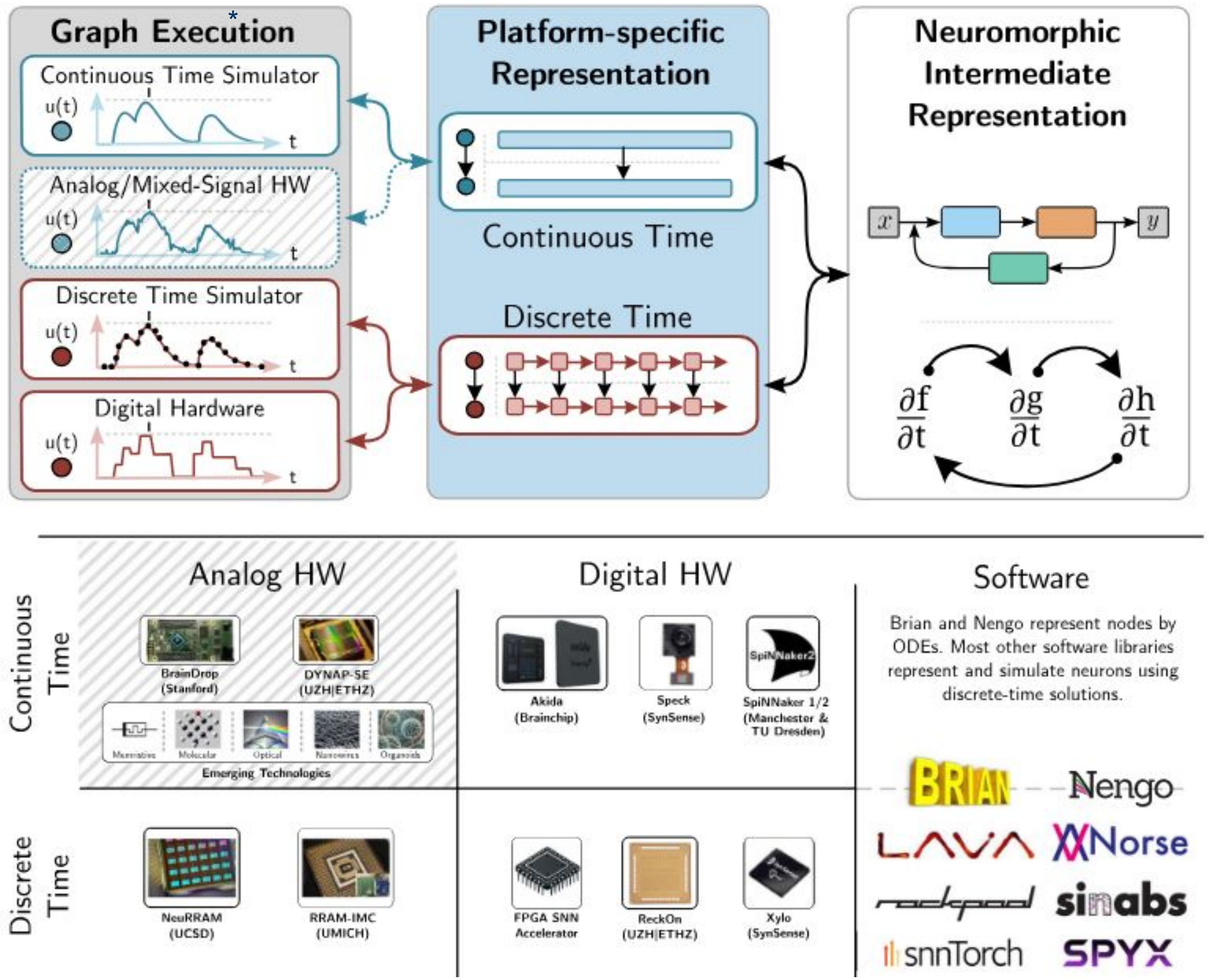
SpiNNaker2 Chip / Board

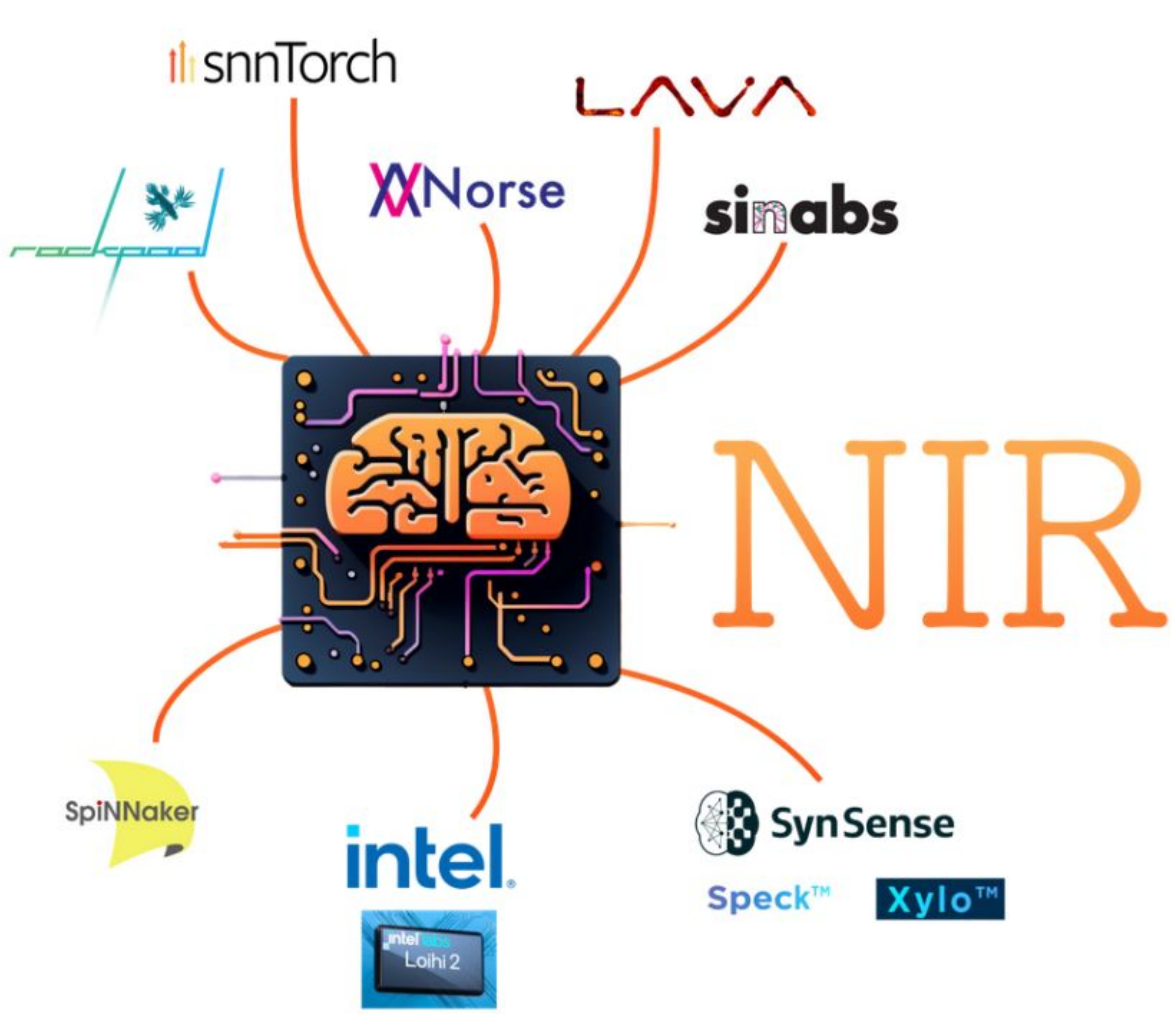
Chip Application Software

Further details on Friday:

Tutorial: **SpiNNaker2**
Tutorial: Beyond Neural Simulation

Neuromorphic IR: A Common Language for Brain-Inspired Computing





https://neuroir.org/

TECHNISCHE UNIVERSITÄT DRESDEN

*Pedersen, J.E., Abreu, S., Jobst, M. *et al.* Neuromorphic intermediate representation: A unified instruction set for interoperable brain-inspired computing. *Nat Commun* **15**, 8122 (2024). https://doi.org/10.1038/s41467-024-52259-9

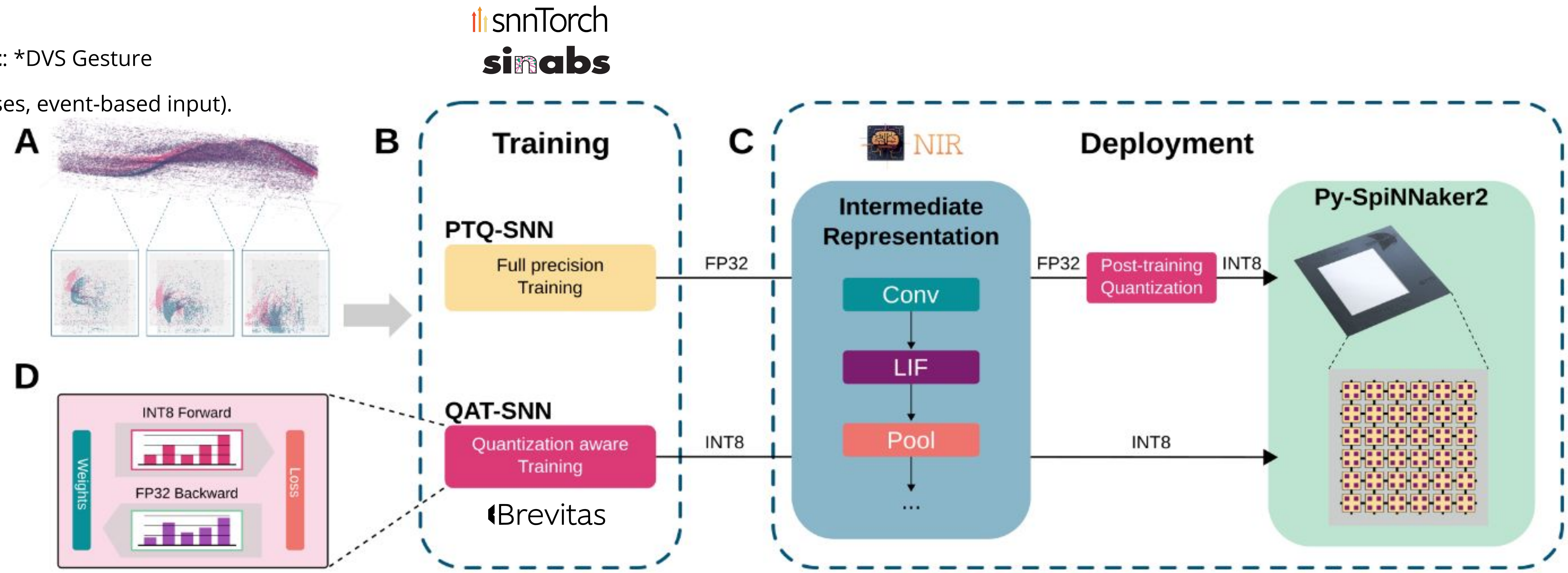
NICE 2025

DRESDEN concept

From Training to Deployment: Our Pipeline

Dataset: *DVS Gesture

(11 classes, event-based input).

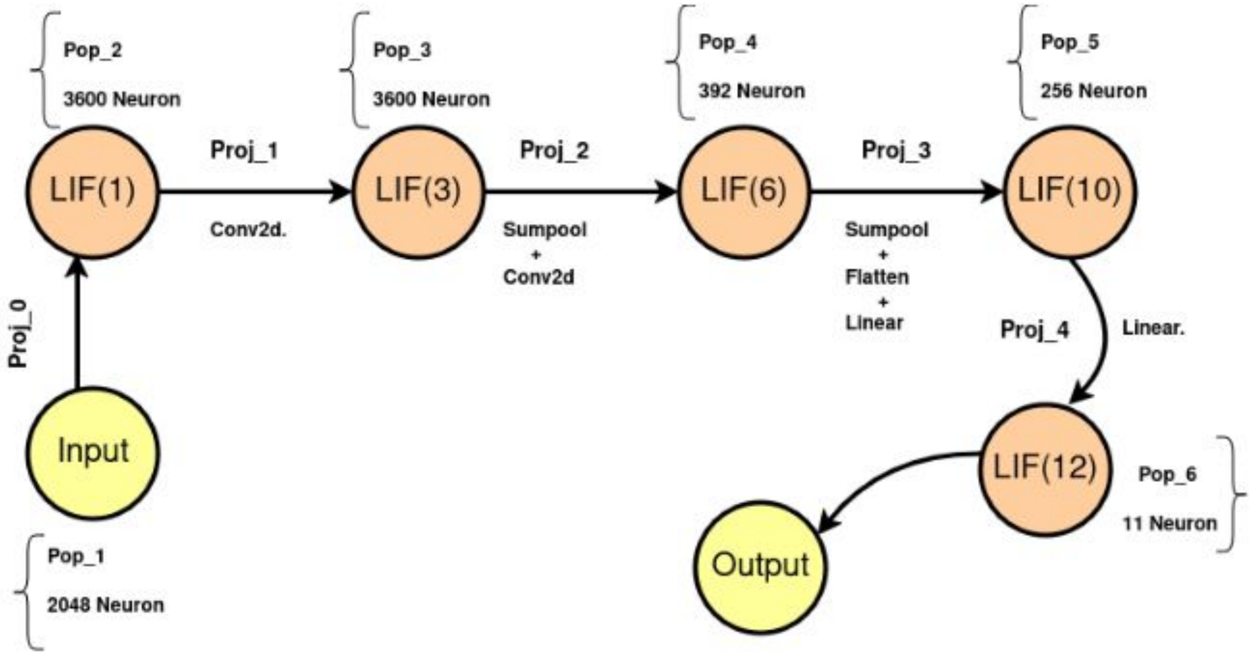


- Backpropagation Through Time
- Surrogate Gradients
- 200 epochs

*Amir, Arnon, et al. "A low power, fully event-based gesture recognition system." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

Experimental Setup: Scaling Deep SNNs on SpiNNaker2

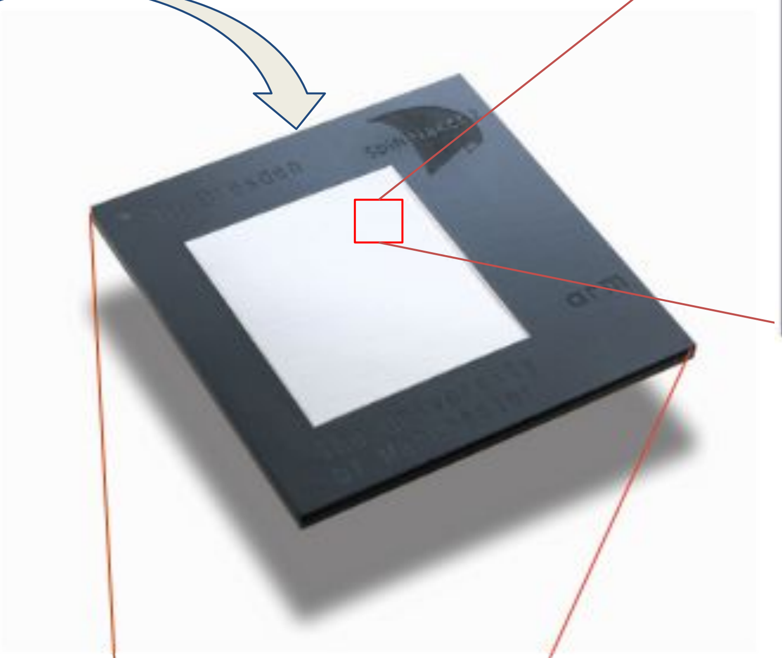
Partitioning per PE



SpiNNaker2 Network

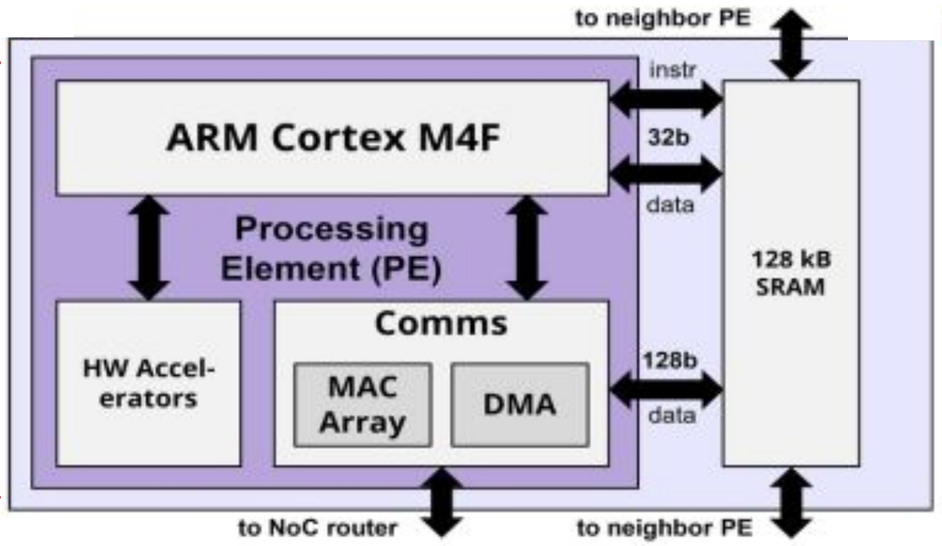
Population	*Max Neurons*
LIF(1)	900
LIF(3)	900
LIF(6)	980
LIF(10)	16
Input	17

Py-spinnaker 2



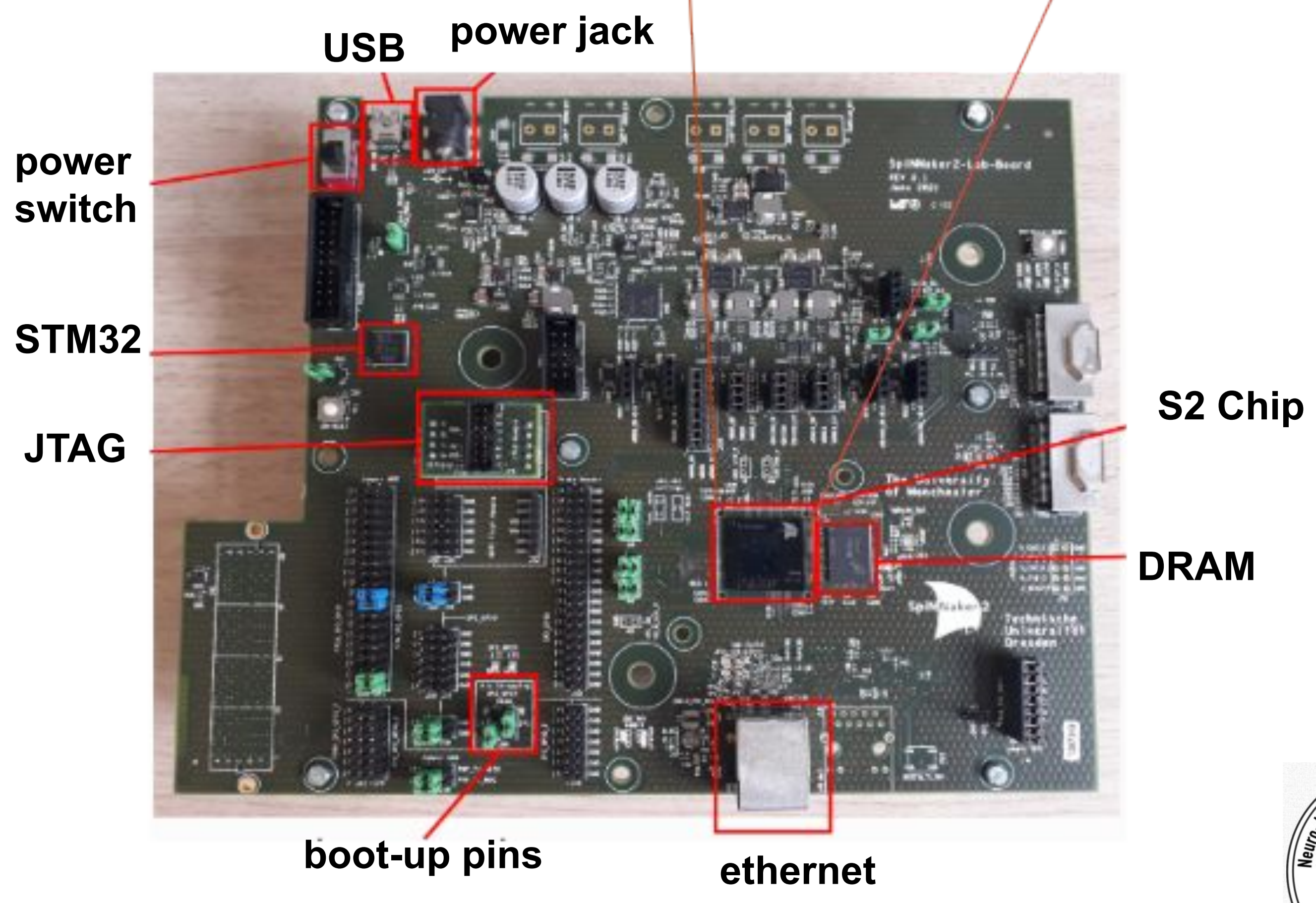
SpiNNaker2 Chip
152 PE

SpiNNaker2 PE



PE Constraints

Neuron Model	*Max Neurons per PE*
LIF Conv2d	1024
LIF Neuron	250
Spike List	500



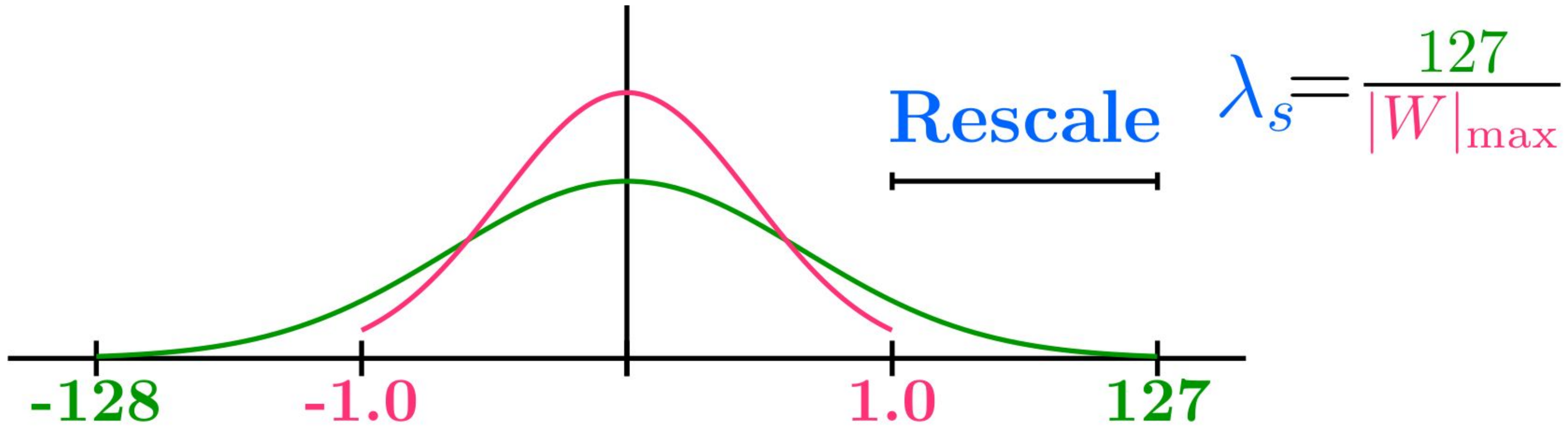
TECHNISCHE
UNIVERSITÄT
DRESDEN

NICE 2025

DRESDEN
concept

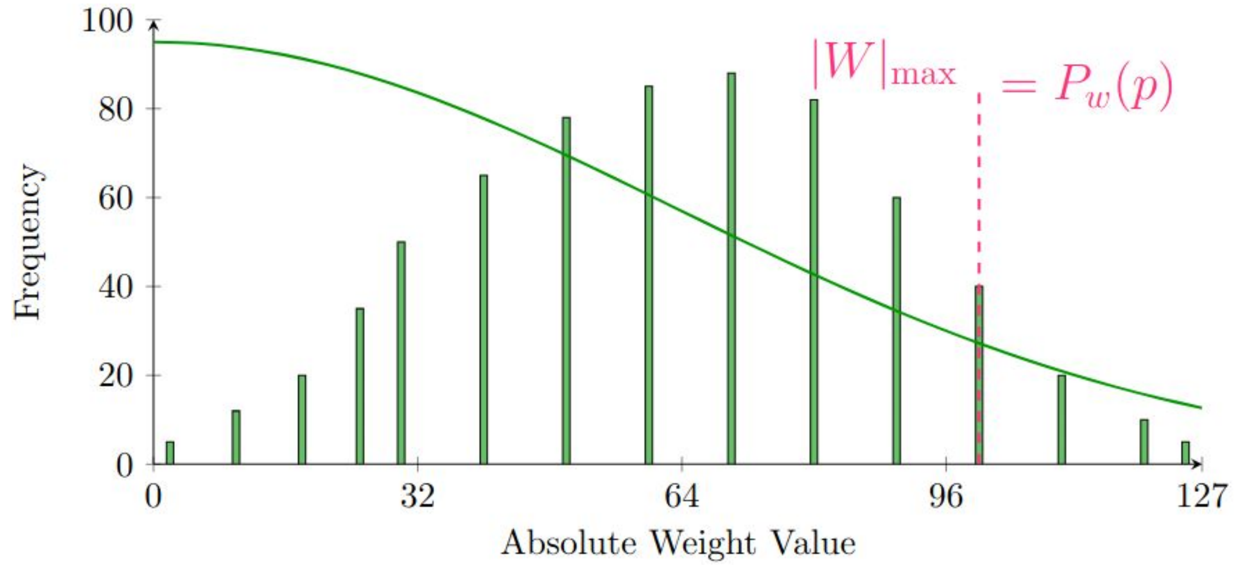
From Training to Deployment: PTQ / P-SNN

1. Compute Scaling Factor



Rescale **weights** from floating-point [-1.0,1.0] to integer [−128,127] using a scaling factor **λs**.

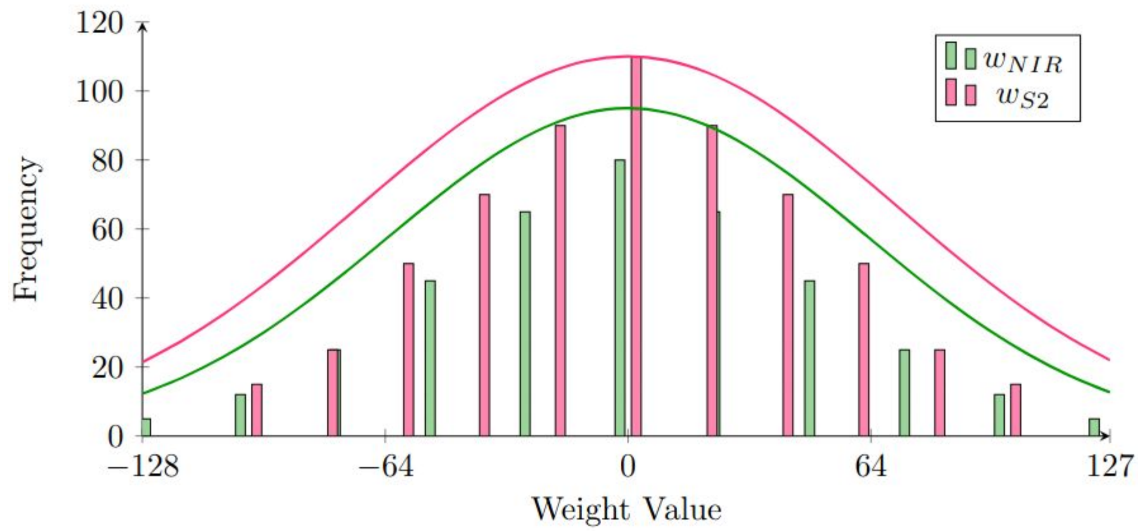
2. Percentile-Based Maximum



Estimate the **maximum absolute weight** using a **percentile function** to reduce the effect of outliers.

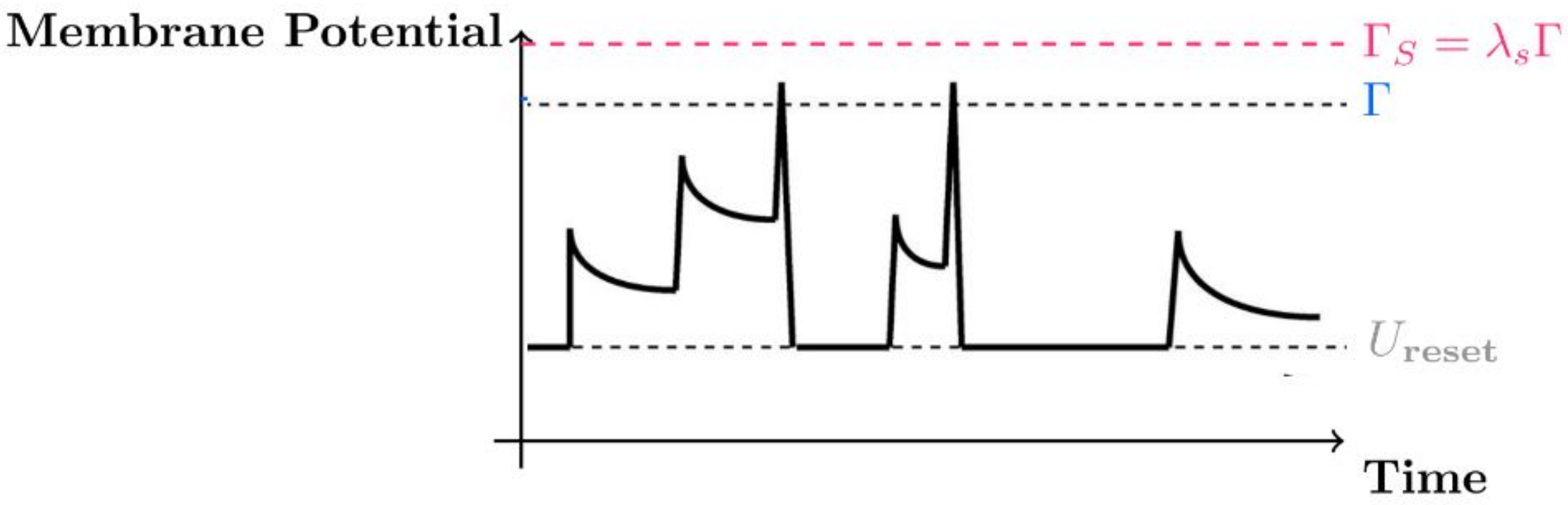
3. Scale and Quantize Weights

$$w_{S2} = \lambda_s w_{NIR}$$



Apply the **scaling factor** and convert the weights to 8-bit integers.

4. Adjust Neuron Thresholds



Scale **neuron thresholds** to maintain spiking behavior under quantized weights.

TECHNISCHE UNIVERSITÄT DRESDEN

NICE 2025

DRESDEN concept

PTQ: Precise but Sensitive

- This approach ensures that 8-bit weights and scaled thresholds maintain compatibility and precision on SpiNNaker2.

High Percentile

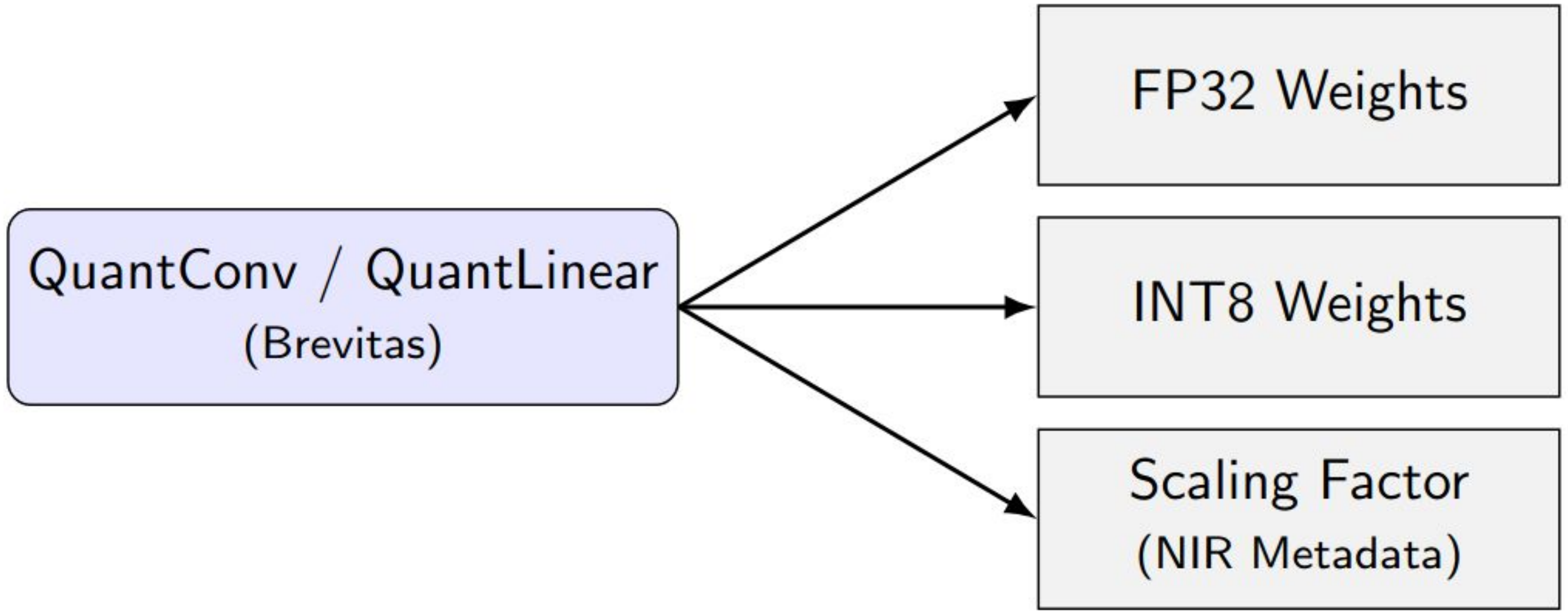
- ✅ Preserves full weight range
- ❌ Sensitive to outliers → may reduce precision

Lower Percentile

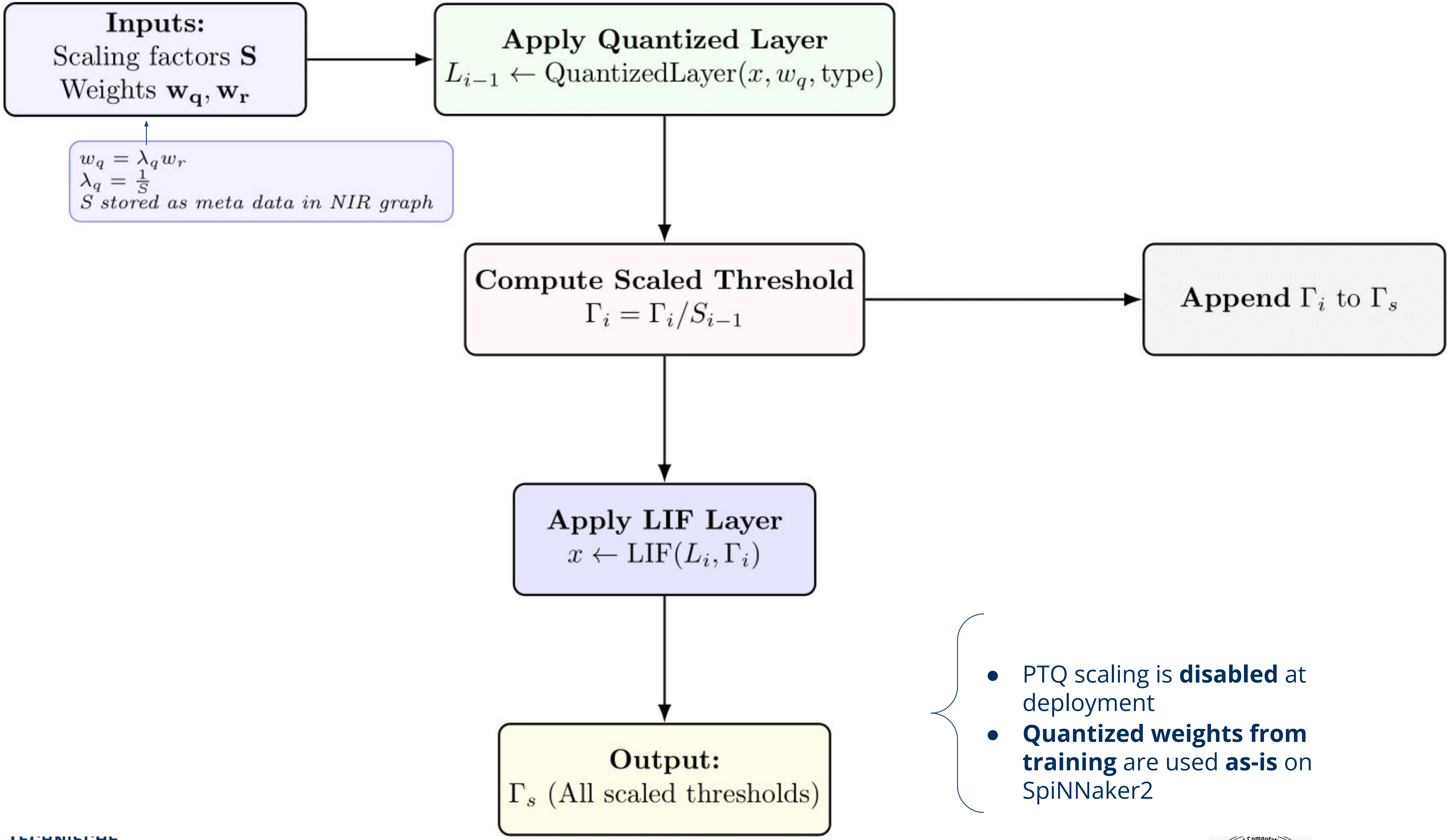
- ✅ More robust to noise → smoother quantization
- ❌ Compresses dynamic range → may limit expressiveness

From Training to Deployment: QAT / Q-SNN

- The **Q-SNN** model retains the structure of the **P-SNN** model (node types, quantities, and layer indices).
- Introduces **quantized layers** that store both full-precision and 8-bit weights, along with their **scaling factors**.



From Training to Deployment: QAT / Q-SNN



- PTQ scaling is **disabled** at deployment
- **Quantized weights from training** are used **as-is** on SpiNNaker2

TECHNISCHE UNIVERSITÄT DRESDEN

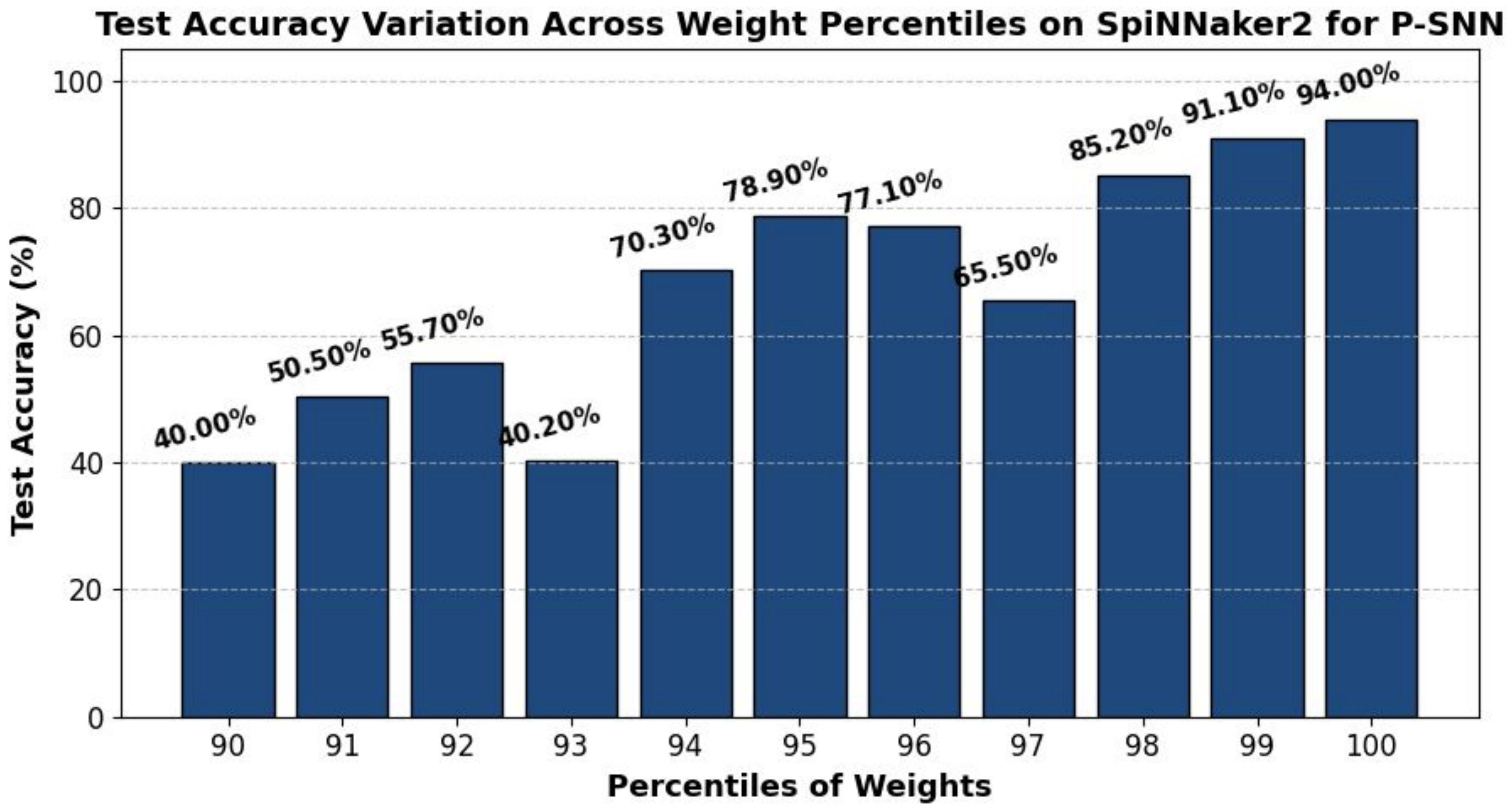
DRESDEN concept

Results: Precision and Performance

- **P-SNN Baseline Accuracy: 95.07%**

- **On-Chip Accuracy after PTQ: 94.0%** (100th percentile)

- **Impact of PTQ: ~1.07%** accuracy drop on-chip

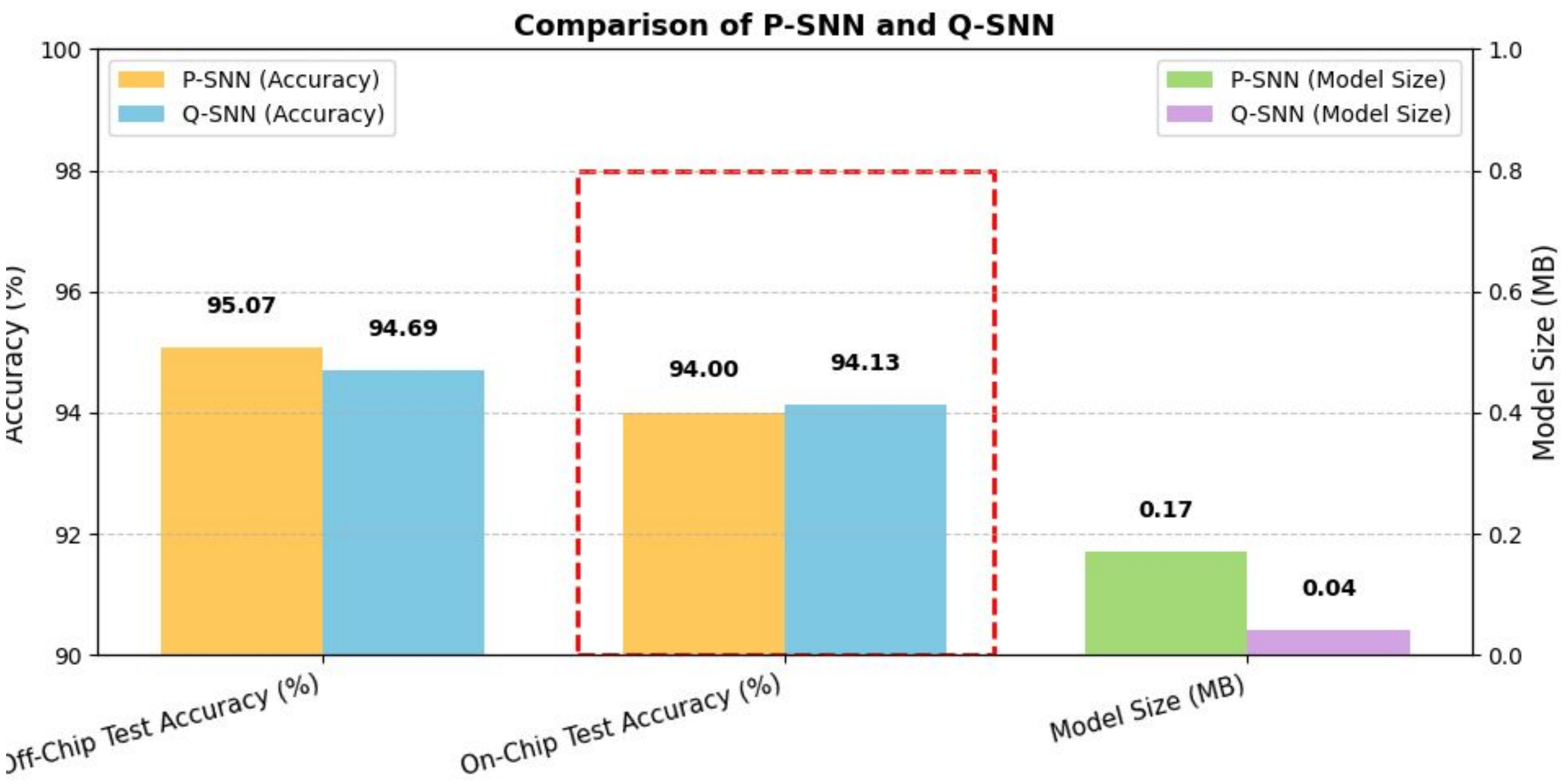


Results: Precision and Performance

Key Takeaways

1. Accuracy vs. Quantization Trade-off

- **P-SNN (Full Precision)** → 95.07% → 94.0% **(-1.07%)** (PTQ)
- **Q-SNN (Quantized)** → 94.69% → 94.13% **(-0.56%)** (QAT)
- **QAT** achieves **94.13%** on-chip accuracy, outperforming PTQ.
- Memory footprint reduced by **75%**, maintaining accuracy.

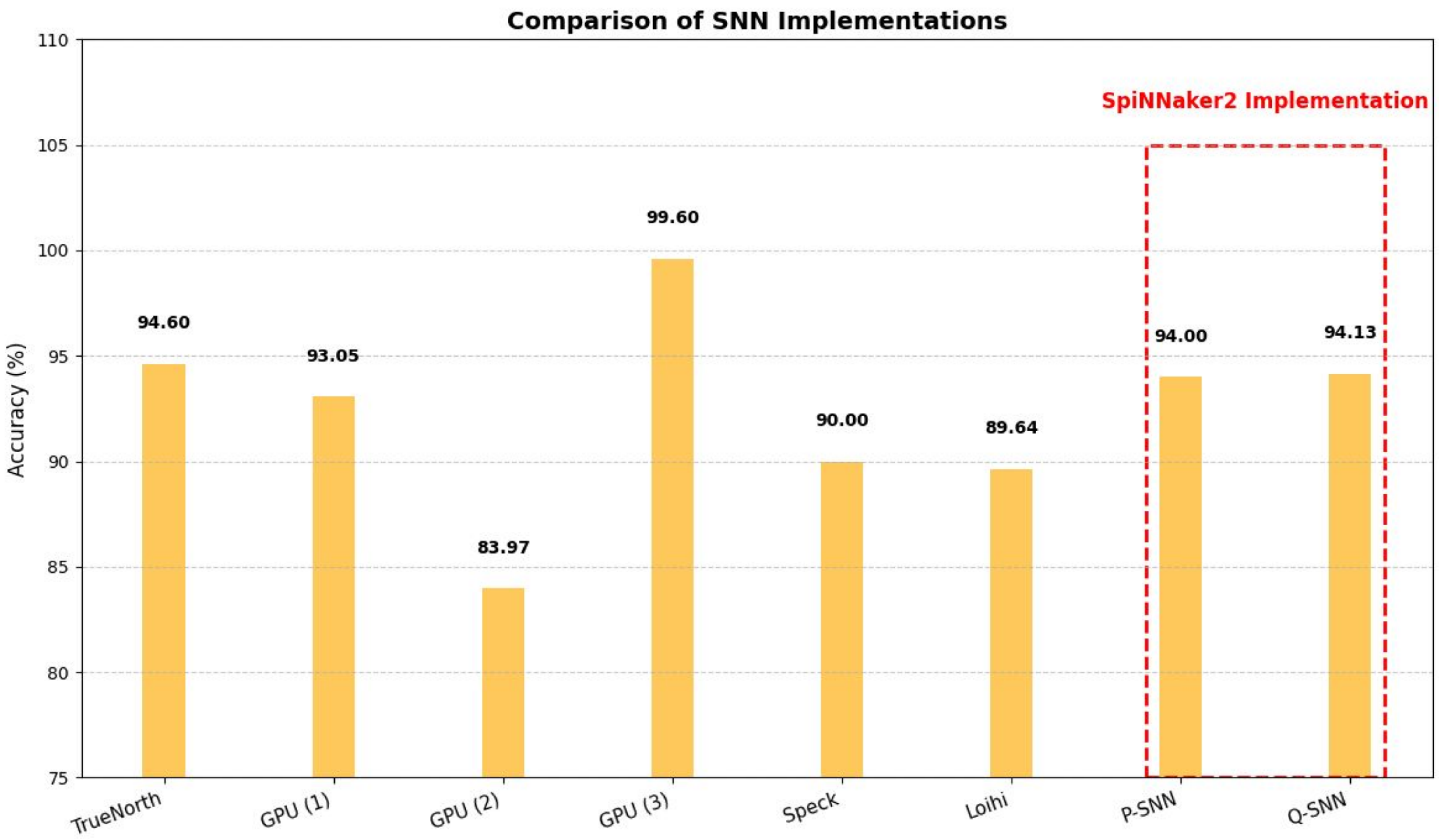


→ QAT minimizes accuracy degradation compared to PTQ, making it **better suited for SpiNNaker2 deployment**.

Results: Precision and Performance

2. Performance Comparison

- Our models achieve **state-of-the-art accuracy** on **DVS Gesture** in both full-precision and quantized settings.
- SpiNNaker2 deployment shows **higher accuracy** than some other neuromorphic platforms.



Lessons from Pushing SNNs to the Edge

- Neuron threshold scaling plays a key role in preserving accuracy.
- QAT with adaptive threshold scaling achieves the best trade-off
- **What surprised us?** Memory optimization had a bigger impact than expected.
- Where can we go next? Multi-chip scaling and real-time gesture recognition.

Q&A / link to gitlab and LinkedIn repo