QoR Estimation via µ-Benchmarking

Felix Jentzsch

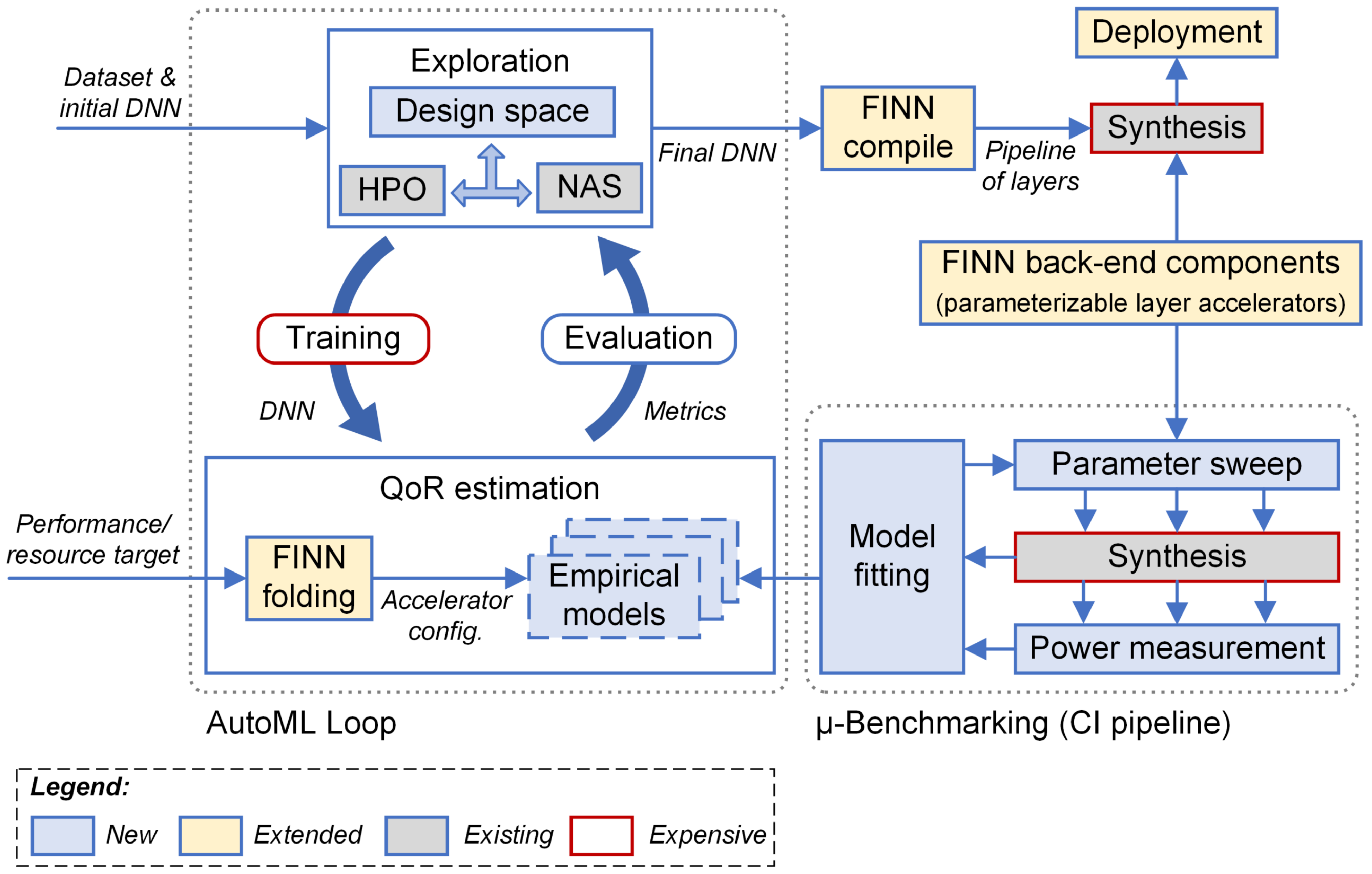
Computer Engineering Group, Paderborn University, Germany

Hardware-aware AutoML Concept

- **Problem:** Enormous design space + expensive exploration
- **Goal:** Develop tools for automatic exploration of optimized co-designs

- **Phase I**: Improve Quality-of-Result (QoR) estimation in FINN to facilitate fast exploration
  - Current resource estimation tools based on bit-ops, FINN, or Vitis HLS are lacking in accuracy
  - Power estimation currently only possible in Vivado after synthesis
  - Only **empirical QoR estimation models** can provide *accurate* feedback as early as possible

- **Phase II**: Leverage recent advancements in automatic machine learning (AutoML) to automate search process
  - Construct a FINN-compatible design space for hyperparameter optimization (HPO) and neural architecture search (NAS)
  - Hardware-aware **AutoML loop** evaluates accelerator metrics (resources, performance, power) alongside accuracy to generate Pareto-optimal solutions



Dataset & initial DNN

Exploration

Design space

HPO

NAS

Final DNN

Training

Evaluation

DNN

Metrics

QoR estimation

Performance/ resource target

FINN folding

Accelerator config.

Empirical models

AutoML Loop

FINN compile

Pipeline of layers

Deployment

Synthesis

FINN back-end components (parameterizable layer accelerators)

Model fitting

Parameter sweep

Synthesis

Power measurement

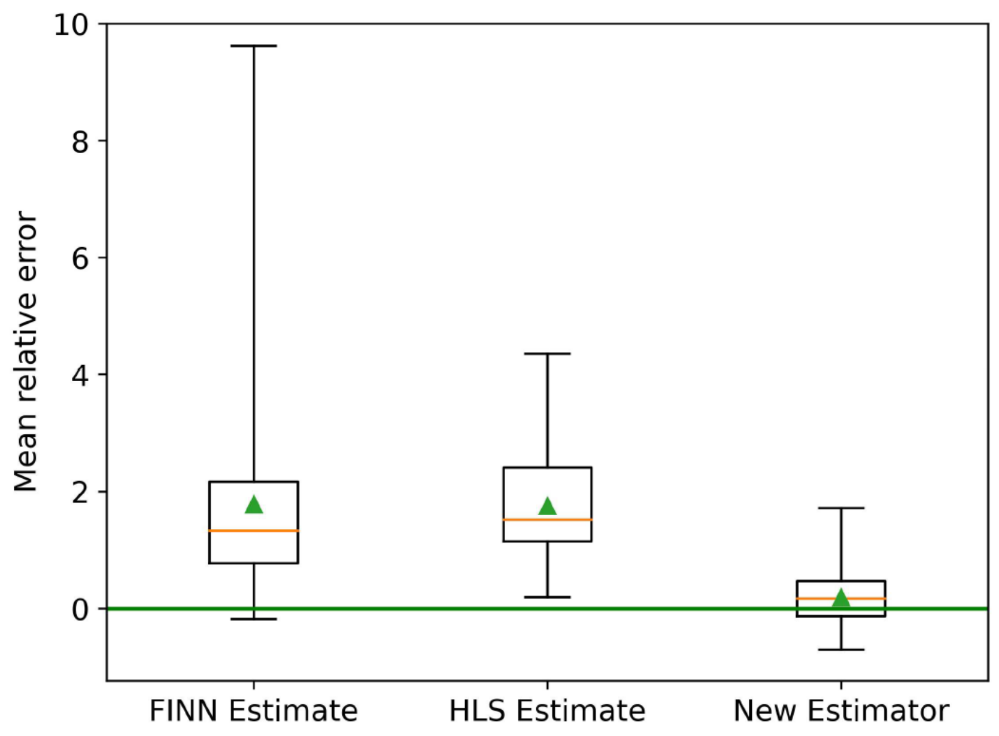
µ-Benchmarking (CI pipeline)

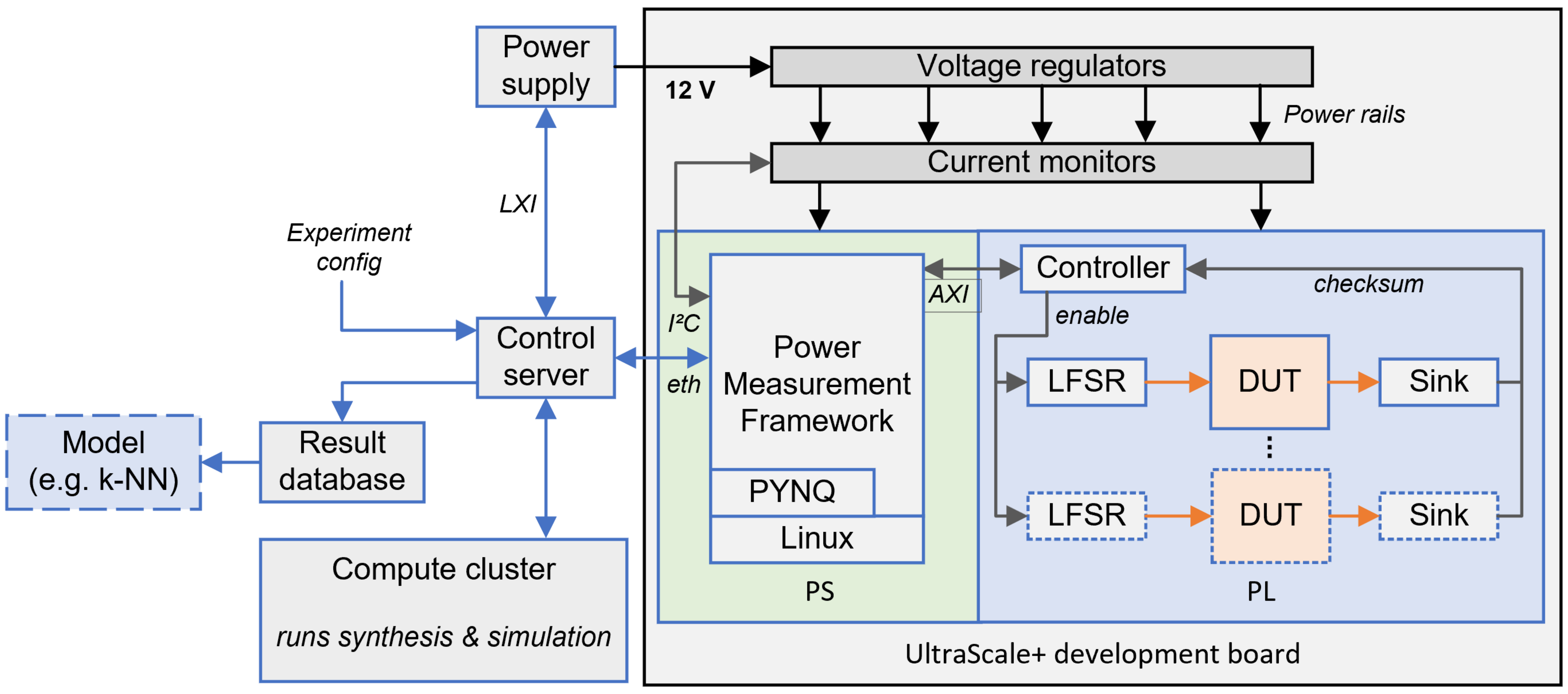
Legend:

New *Extended* *Existing* *Expensive*

Benchmarking and Estimation Infrastructure

- **Micro-benchmarking infrastructure** for large-scale synthesis, simulation, and power measurement
  - Parallel synthesis/simulation on our compute cluster
  - Accurate power measurement via monitoring circuitry of development boards
  - Database of results can be updated automatically as tools progress (CI pipeline)
- By relying on measurements for model creation, we gain insight into all low-level optimizations





Power supply

LXI

Experiment config

Control server

Model (e.g. k-NN)

Result database

Compute cluster

runs synthesis & simulation

12 V

Voltage regulators

Power rails

Current monitors

I²C

eth

Power Measurement Framework

PYNQ

Linux

PS

AXI

Controller

enable

checksum

LFSR

DUT

Sink

LFSR

DUT

Sink

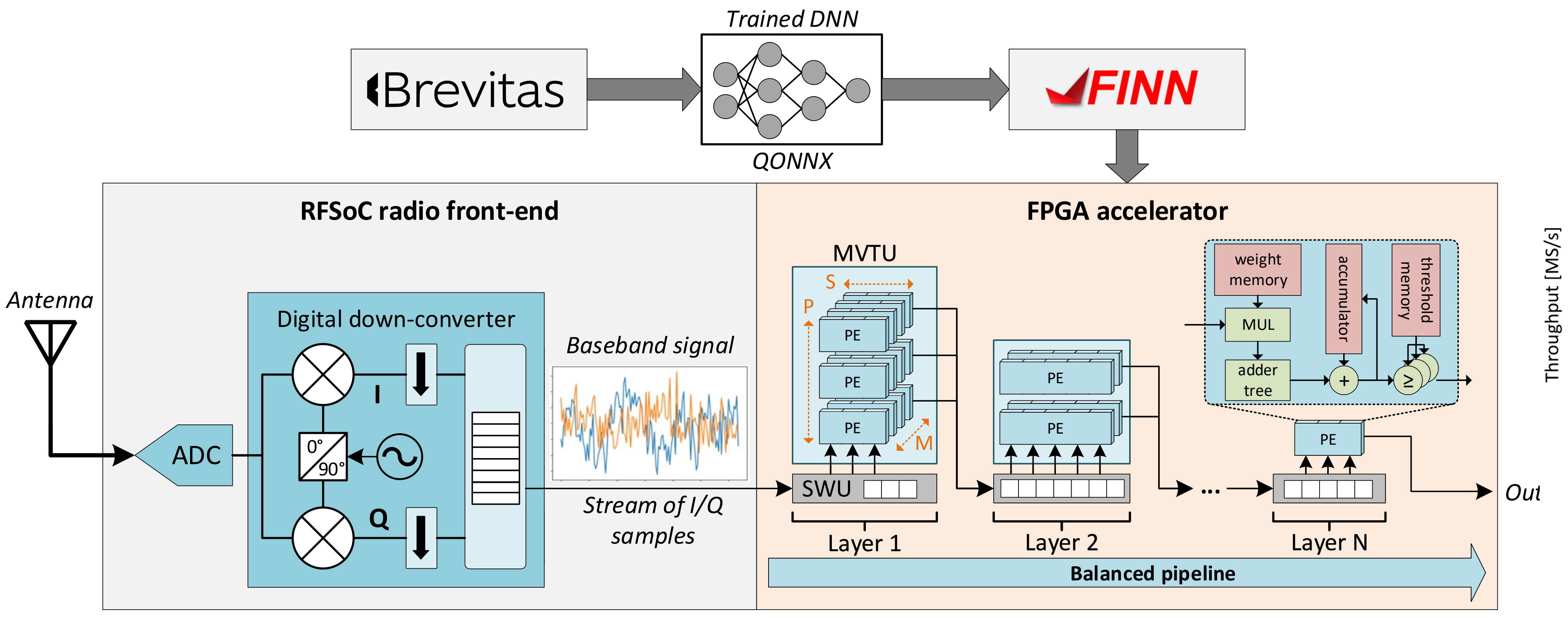
PL

UltraScale+ development board

- **FINN integration**
  - GitLab-based CI including artifact storage for benchmark results
  - Benchmark suite operates on common individual FINN layers ("custom ops"), like the MVAU
  - Includes regression testing on full DNNs
  - Adds new analysis passes for best-effort resource & power estimation based on latest available benchmark results
  - Adds new back-end transformations to generate FINN accelerator with test harness and driver for different platforms (Zynq, Alveo, Versal)

RadioML Use Case

- FINN's unique optimization potential is especially relevant for the **RadioML** field: traditional DSP algorithms are replaced by DNNs
  - Extreme throughput and latency demands, often on power-constrained embedded devices
- We **extend FINN** to support small CNNs with extreme parallelism
  - Optimized RTL implementation of the sliding window unit (SWU)
  - Introduction of an additional degree of parallelism (*M*) beyond input (*S*) and output (*P*) folding
- Exploration of a modulation classification use case using a 1D CNN reaches **state-of-the-art performance**
  - Batch-independent architecture can reach peak throughput and latency at the same time, in contrast to GPUs
  - Prototype integration into a transceiver design on the RFSoC 2x2/4x2 platform



Brevitas

Trained DNN

QONNX

FINN

RFSoC radio front-end

Antenna

ADC

Digital down-converter

I

Q

0°

90°

Baseband signal

Stream of I/Q samples

FPGA accelerator

MVTU

S

P

M

PE

SWU

weight memory

MUL

adder tree

accumulator

threshold memory

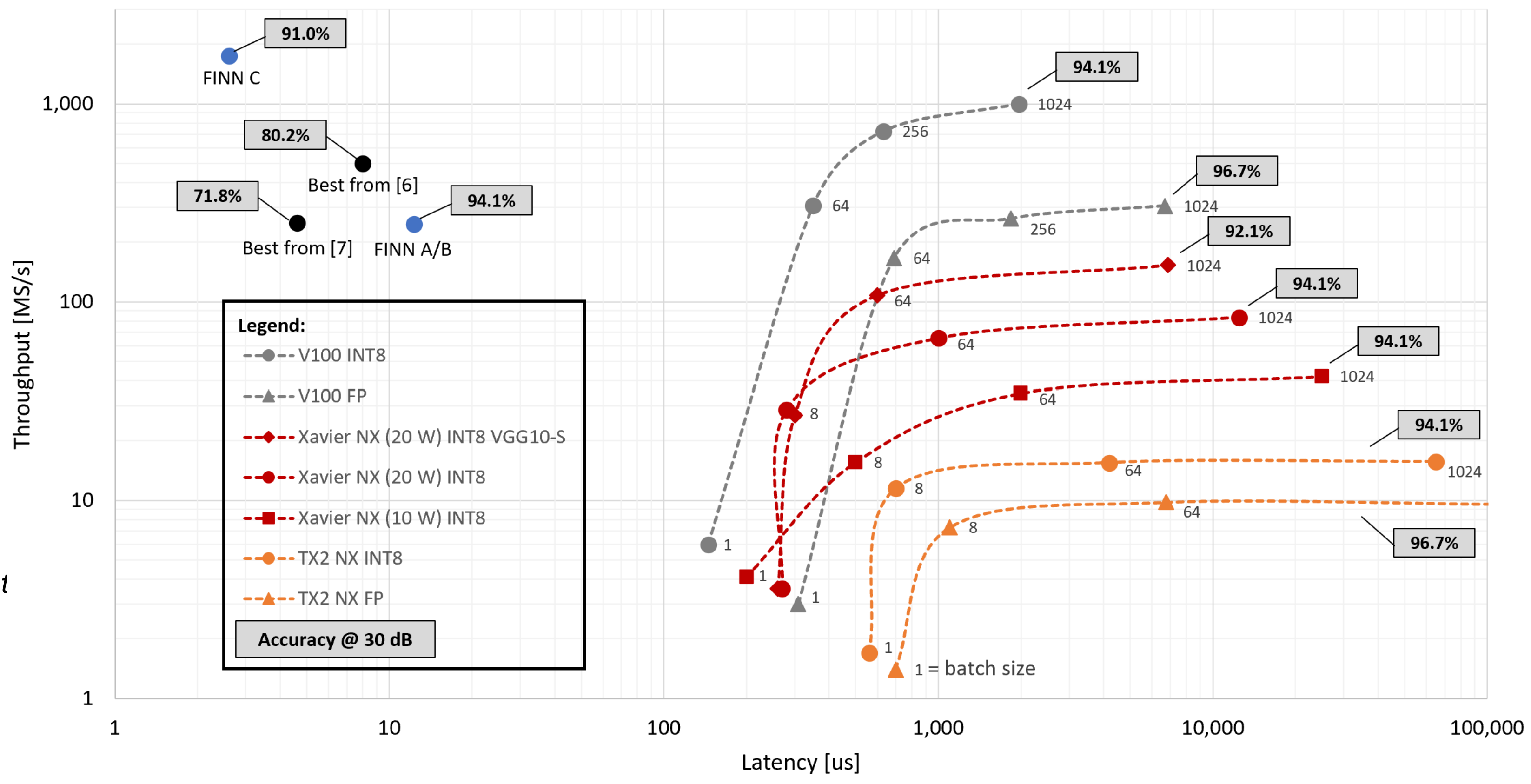
Layer 1

Layer 2

Layer N

Out

Balanced pipeline



Contact: ✉ felix.jentzsch@upb.de fpjentzsch eki-project.tech cs.uni-paderborn.de/ceg