AI-BASED REAL-TIME WAVEFORM PROCESSING AT THE EDGE

IHL KICKOFF MEETING 2026 – SIEGEN

17TH JULY 2026 | VESSELIN DIMITROV | ELGUJA ABULADZE

Member of the Helmholtz Association

Universität Siegen

CPPS Center for Particle Physics Siegen

color meets flavor

JÜLICH Forschungszentrum

Integrated Computing Architectures

ICA

MOTIVATION

Trigger for radio detection of air showers

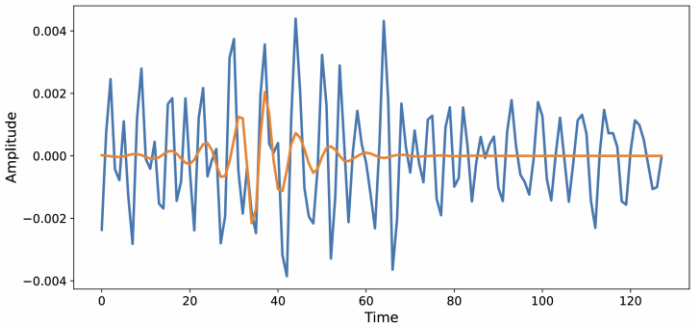
- **Low SNR** signals difficult to detect with classical amplitude-based trigger

  →High false positive rate or efficiency loss

AI-based detection strategy:

- Pipeline:
  - **Denoiser → Classifier**

→**FPGA** implementation



source: Q.Dorosti 2025 *JINST* **20** P10010

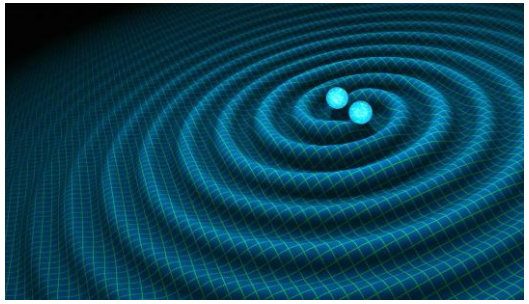
Noise mitigation for Einstein Telescope

- Detection of **gravitational waves** using laser interferometry
- Enables **multimessenger observations** of the universe
- Limited in part by environmental noise sources (e.g., **seismic / Newtonian noise**)
- Online Newtonian-noise mitigation needed

→**FPGA** implementation

→**Requirements**:

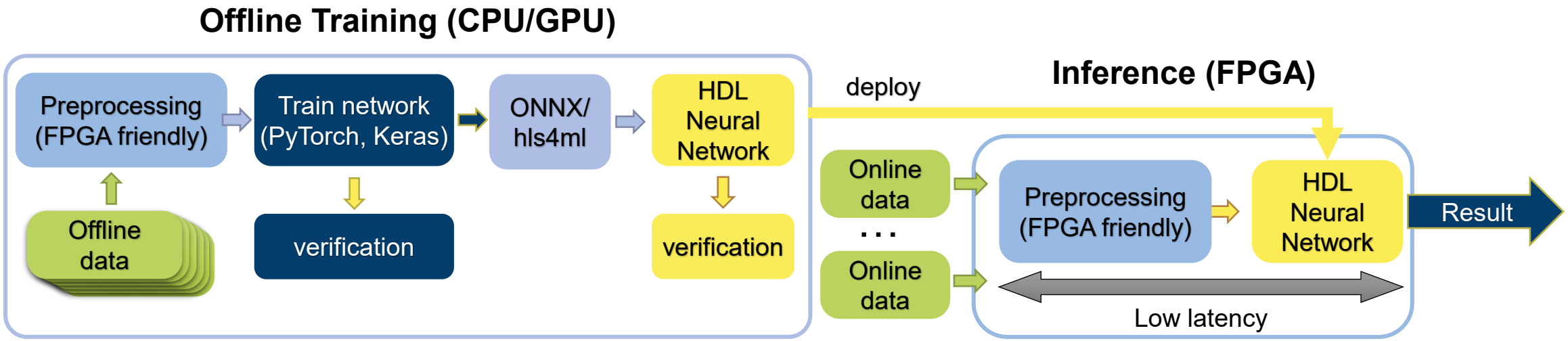
- Non-linear functions evaluation
- Low latency, resource efficient
- High accuracy



source: https://www.einsteintelescope-emr.eu/de/gravitationswellen/

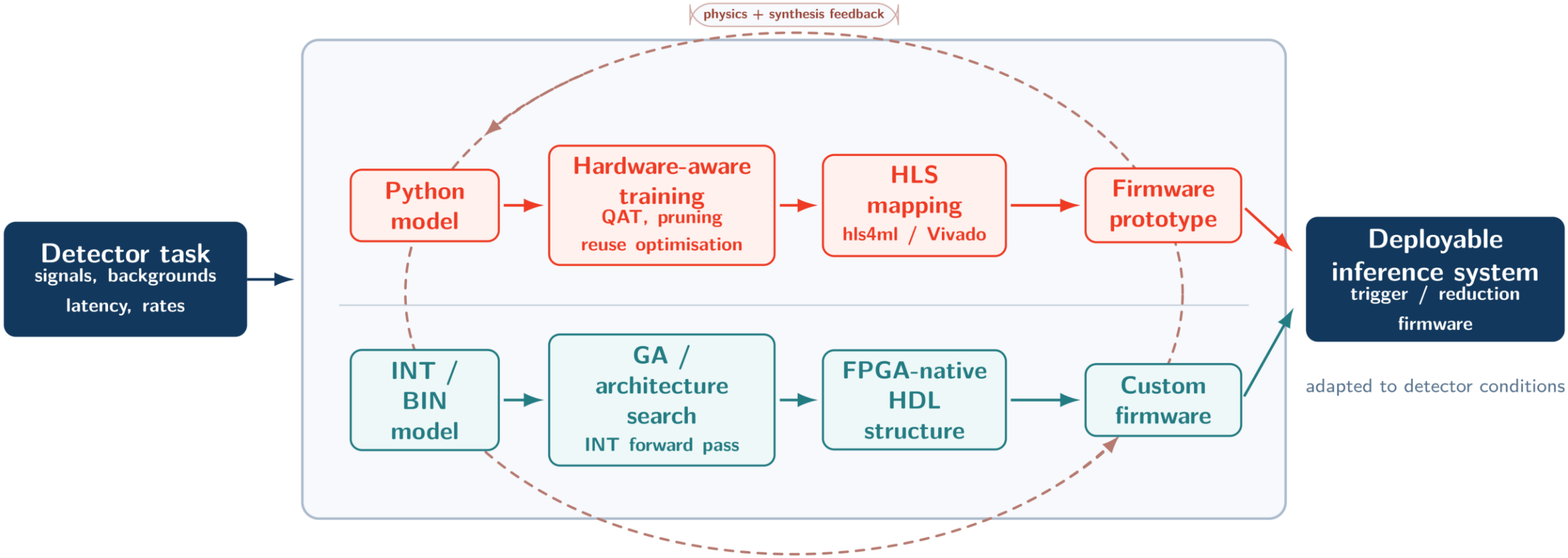
WORKFLOW

Current state



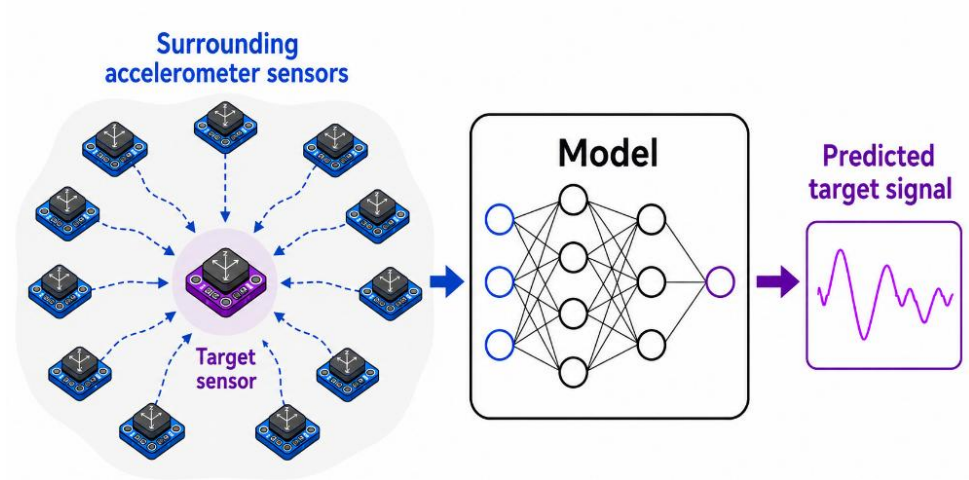
WORKFLOW

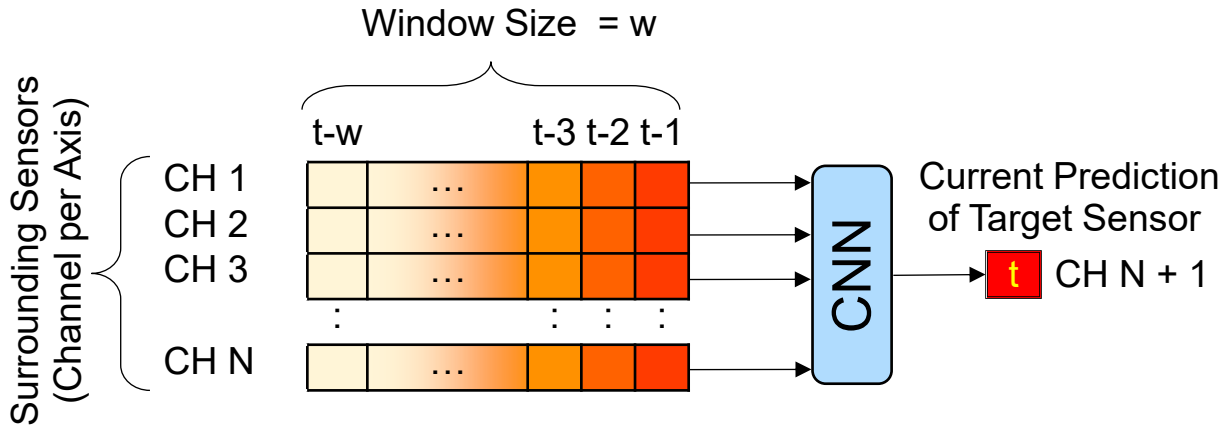
IHL importance: Develop alternatives for deployment of AI firmware



PREDICTION FROM ACCELEROMETER ARRAYS

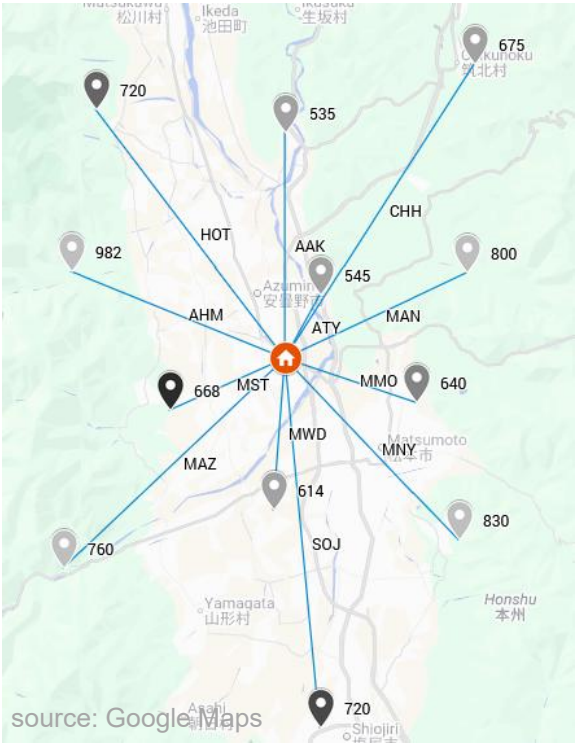
- Accelerometer arrays can be used to predict seismic disturbances at the mirror
- Current approaches are done offline on recorded data [1].
- Our goal is to predict it in real-time (online)
- In preparation for ET our proxy is
  - **Prediction** of the target sensor **seismic signal** from the others





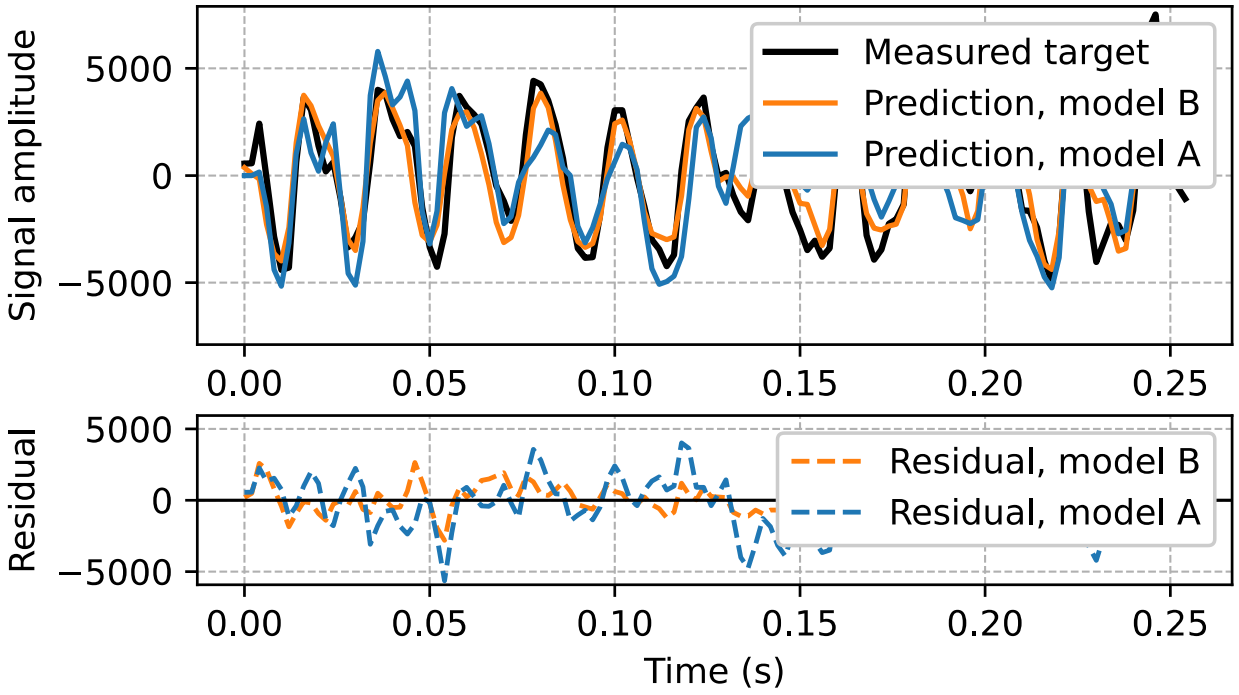
[1] doi.org/10.1088/1361-6382/acf3c8

DATA SOURCES

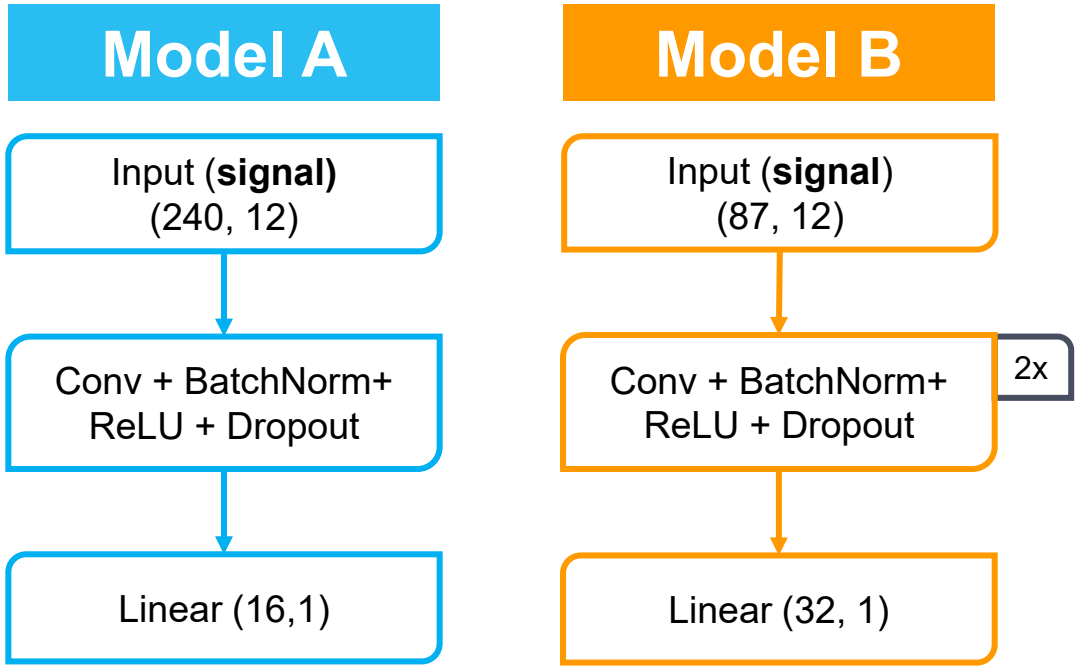


	ETpathfinder (ETPF) https://www.etpathfinder.eu	Hi-Net https://www.hinet.bosai.go.jp/about_data/	Coridonia https://data.mendeley.com/datasets/zsrk4cngtr/2
Dataset size	30 hours	12 hours	~ 2 hours
Sampling rate	500 Hz	100 Hz	250 Hz
ADC resolution	24 bits	27 bits	20 bits
Number of channels	13x vertical	13x vertical	5x 3-axial

ETPF RESULTS



FPGA	Latency [µs]	BRAM 18kb	DSP	FF k	LUT k
ZCU216	464	426	71	108	78
		20 %	2 %	13 %	18 %



Reach

SER = 6.7 dB

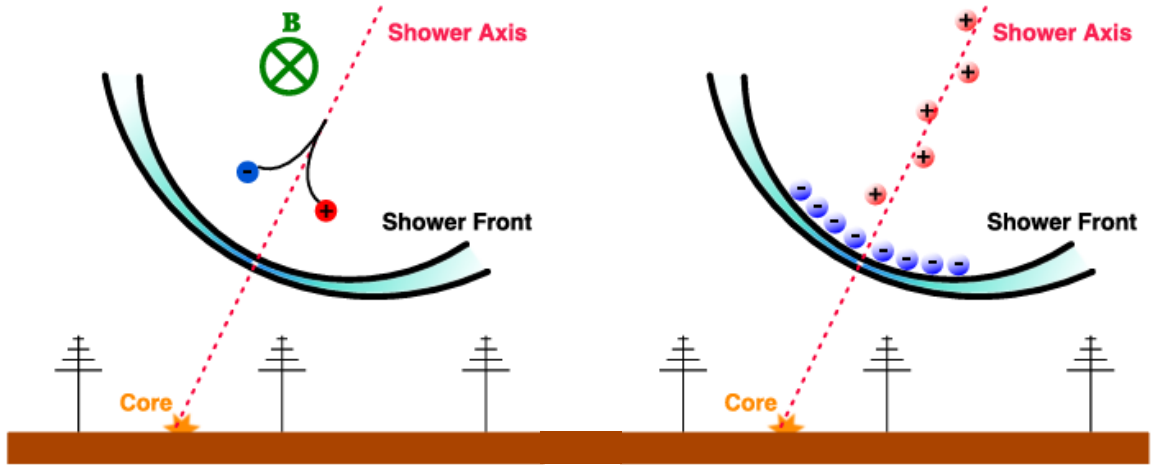
SER = 6.2 dB

improvement

DATASETS FOR TRIGGER

Extended Air Showers from Ultra High Energy Cosmic Rays

- **Trace format:**
  - 180 MHz – 250 MHz sampling rate
  - **128 samples** per trace



Main mechanisms for radio production. (left) Geomagnetic emission, (right) Charge excess emission

T.Huege: https://arxiv.org/pdf/1601.07426v1

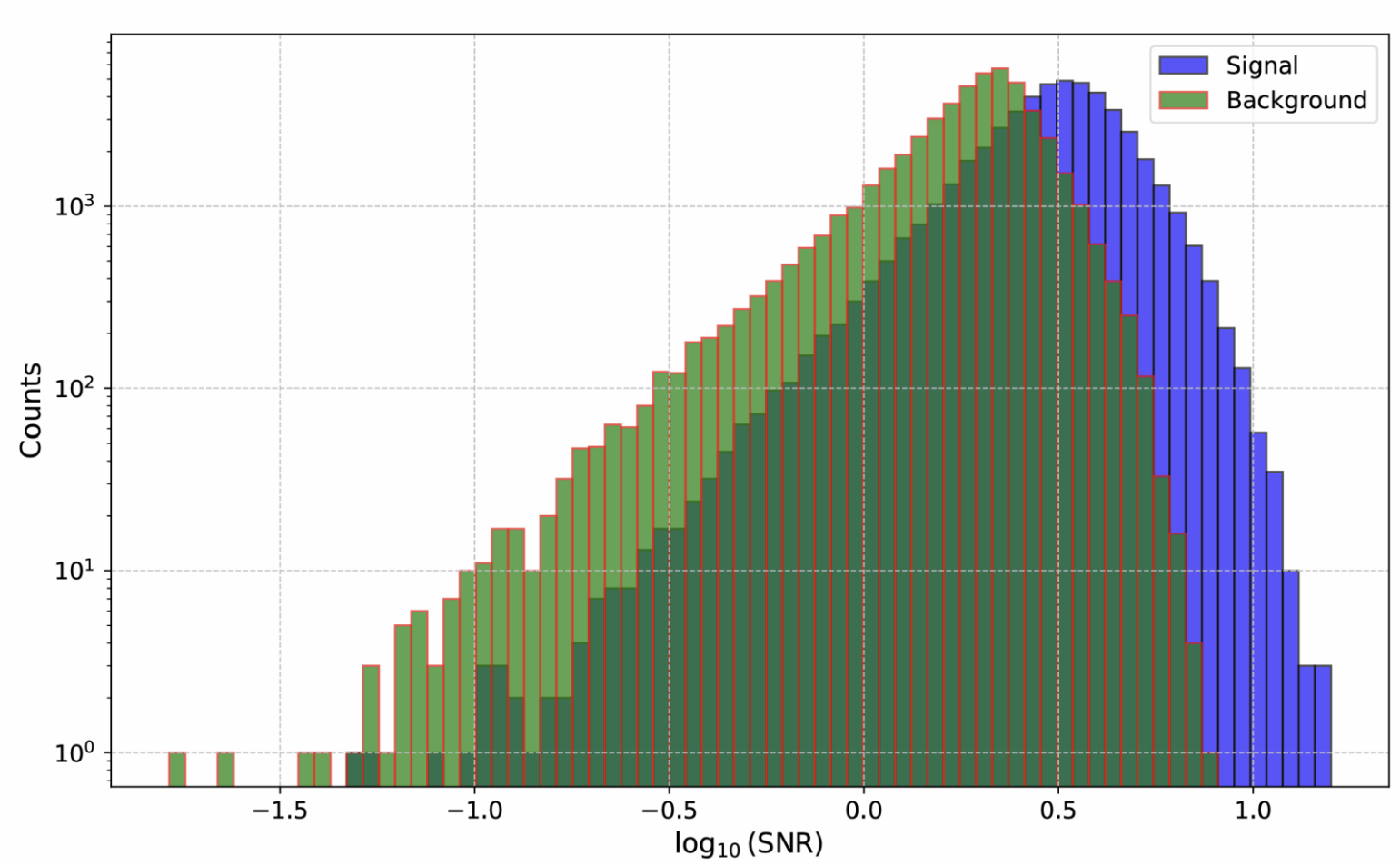
- **Trace categories:**
  - ***Background***:
    Measured background traces
  - **Pure pulse**:
    Simulated clean pulses[1]
  - **Signal**:
    Pulses injected into measured background

Butterfly antenna used for capturing the noise on the campus in Siegen

[1] Pierre Auger collaboration, T. Huege et al.: https://pos.sissa.it/501/292

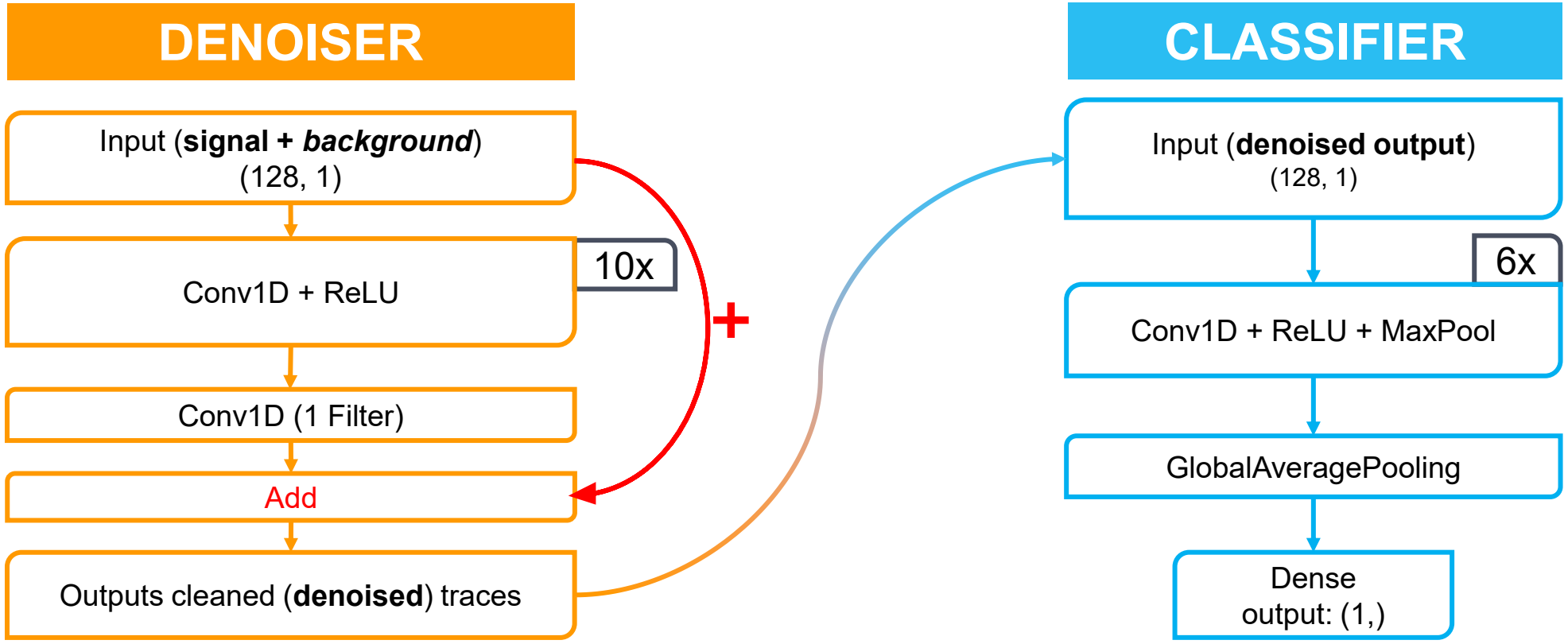
CPPS color meets flavor JÜLICH Forschungszentrum Integrated Computing Architectures ICA

SNR DISTRIBUTION



CPPS color meets flavor JÜLICH Forschungszentrum Integrated Computing Architectures ICA

HYBRID MODEL ARCHITECTURE

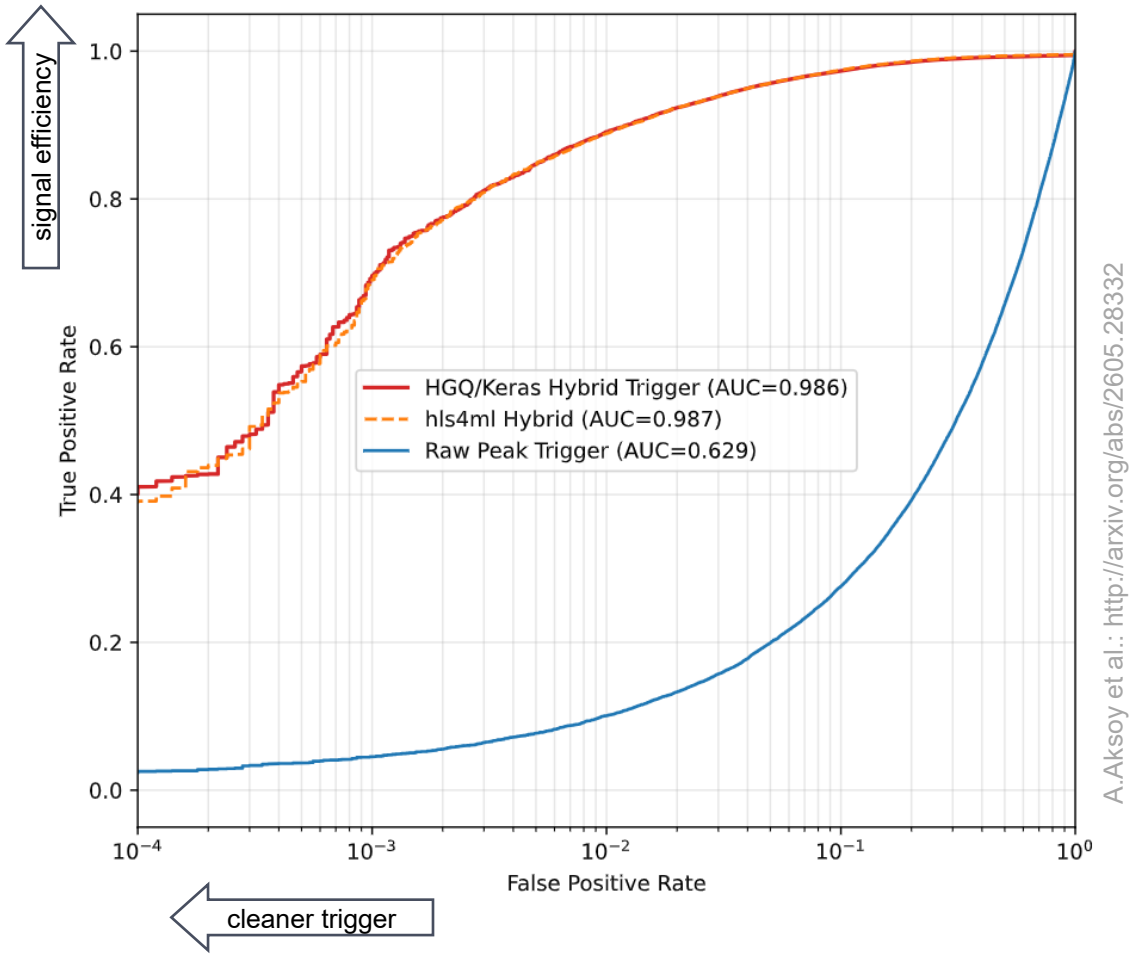


→ Suppress noise while preserving signal structure

→ Binary classification: signal vs. noise

HYBRID (DENOISER + CLASSIFIER)

Hybrid performance vs. simple threshold trigger



FPGA	Power [W]	Latency [µs]	BRAM 18kb	DSP	FF k	LUT k
Zynq-7020	0.56	13.6	100 36%	42 19%	39 37%	23 43%
Kintex U040	0.36	5.7	63 5%	68 4%	25 5%	23 9%
Alveo U250	0.27	3.6	64 1%	70 ~0%	21 ~0%	22 1%

→Improved signal efficiency for very low FPR

→High Area-Under-Curve (AUC)

SUMMARY AND OUTLOOK

Trigger for radio detection

- **Two-stage architecture:**
  - **Denoiser** for noise suppression
  - **Classifier** for signal identification
  - Models trained independently of each other
- **High classification** performance demonstrated ($\mathrm{AUC} = 0.986$, $TPR \approx 0.4$ @ $\mathrm{FPR} = 10^{-4}$)
- **Low resource usage** achieved
- Estimated **power consumption**: 0.562 W

→Implementation with **Binary Neural Networks**

→Full Hardware validation

- Validation with **real, live antenna data**

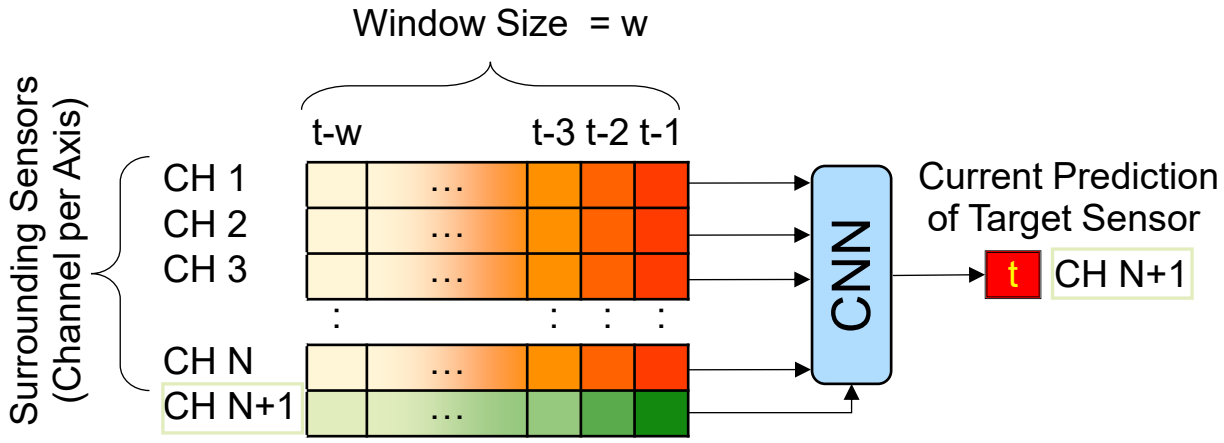
Signal prediction of seismic events

- Model that **predicts** seismic noise
  - **Input** N sensors → **predict** sensor N + 1
- Target-sensor prediction demonstrated with approximately **6–7 dB SER** on the tested dataset
- Synthesized and implemented for the ZCU216, with a latency of 464 µs and a resource utilization of <20% for each resource block

→Try other architecture (**Temporal Convolutional Network**)

→Perform **real experiment** at FZJ

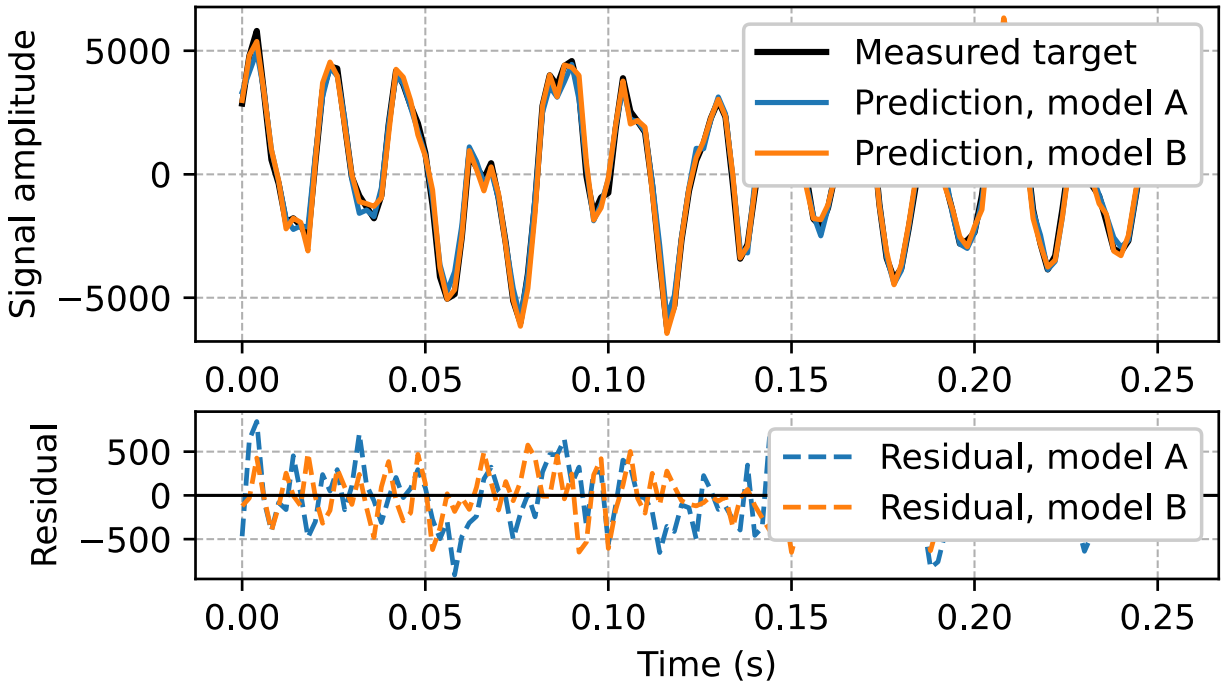
BACKUP

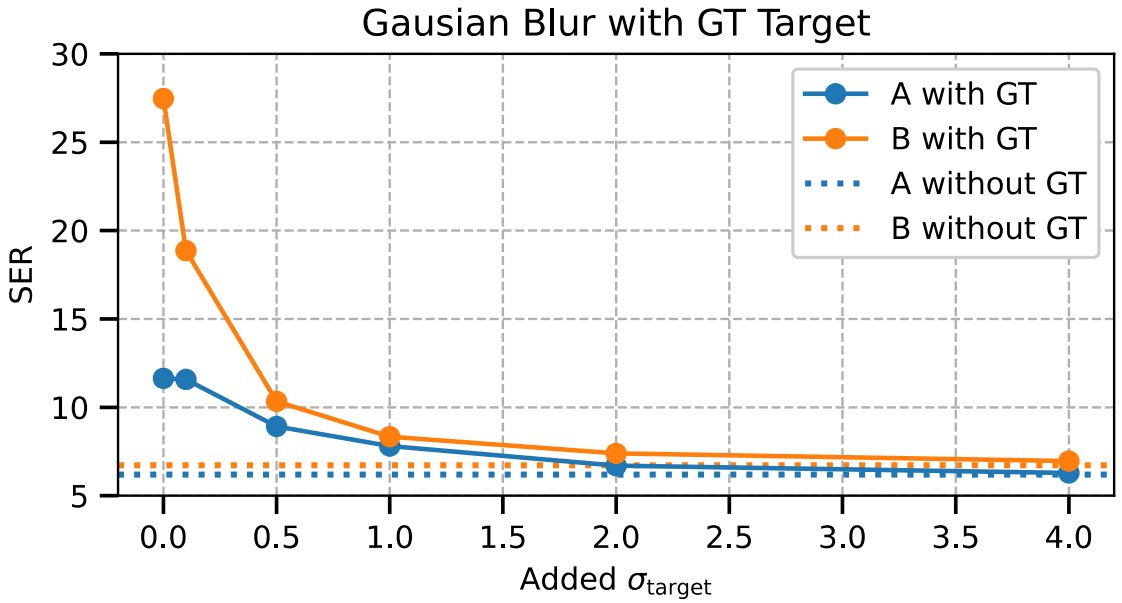


Signal to Error Ratio (higher is better)

$$\mathrm{SER} = 10 \times \log_{10} \frac{\sum_{i=1}^{N} y_i^2}{\sum_{j=1}^{N} (y_i - \tilde{y}_i)^2}$$

SER is the ratio between target-signal power and residual power, expressed in dB



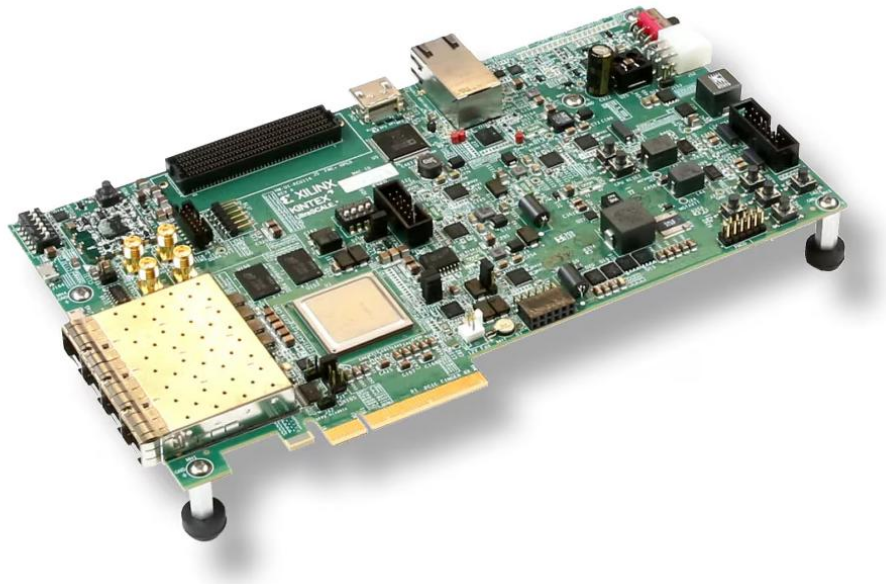


FIELD-PROGRAMMABLE GATE ARRAYS (FPGA)

- **Reconfigurable logic**
- Consists of dedicated resources
  - **Block RAM** (BRAM) → on-chip memory
  - **Digital Signal Processors** (DSPs) → arithmetic units
  - **Flip-Flops** (FFs) → sequential storage
  - **Look-Up Tables** (LUTs) → truth tables (NAND, XOR, …)
- True hardware-level **parallelism** (unlike CPUs/GPUs)

→Deterministic **latency**

→**High throughput** at **low power** consumption



source: AMD Kintex™ UltraScale+™ FPGA KCU116 Evaluierungskit

FRAMEWORKS

Model development (Keras)

- High-level Python framework for defining and training neural networks
- Quantization-Aware Training
  - Incorporates fixed-point constraints during training
  - Preserves accuracy and reduces model size

Firmware Generation (hls4ml)

- Create firmware implementations of AI models for FPGA deployment

MODEL ARCHITECTURE

DENOISER

Input (**signal + pure pulse**)
(128, 1)

10x

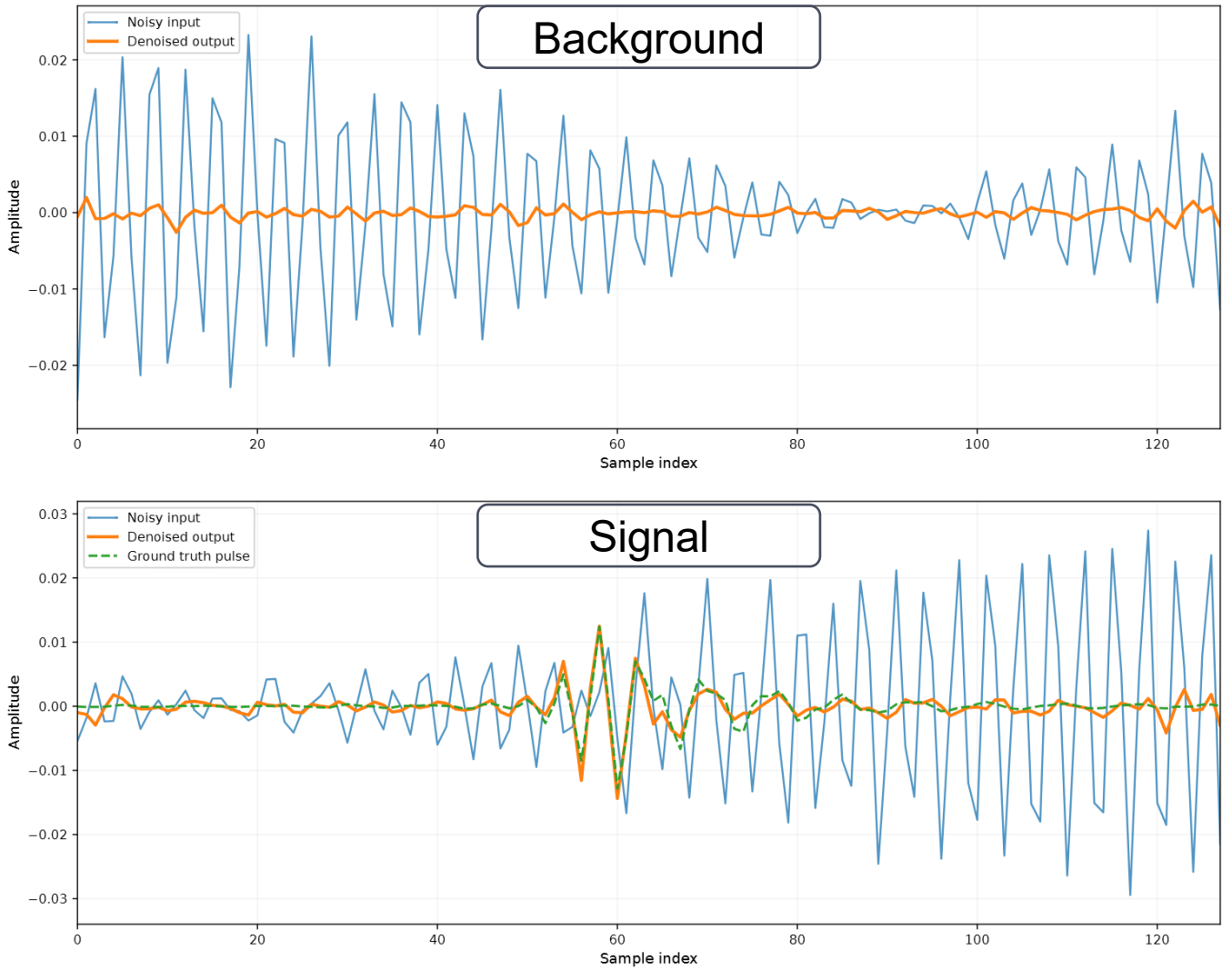
Conv1D + ReLu

Conv1D (1 Filter)

Add

Reshape
(128,)

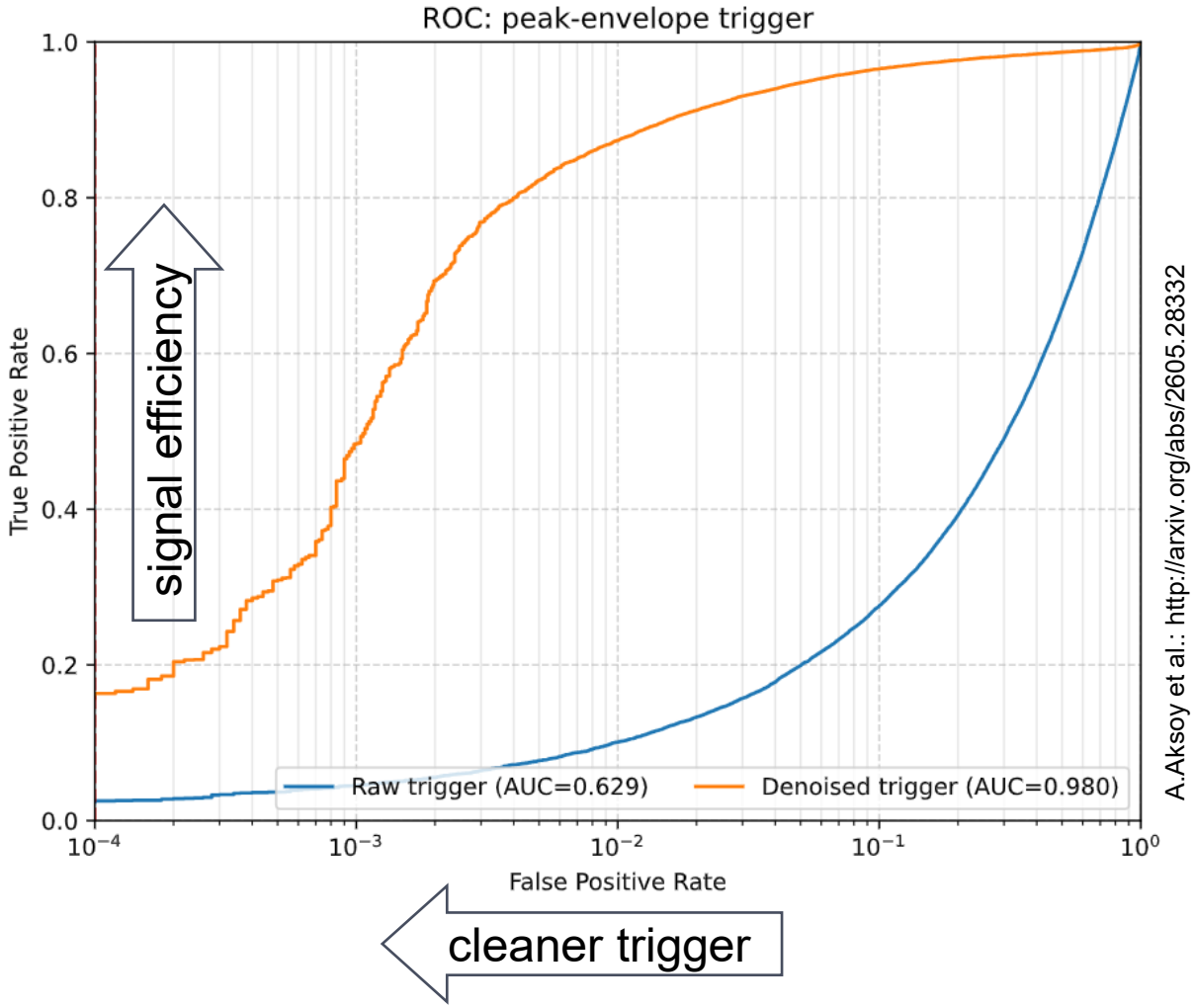
→ Suppress noise while preserving signal structure



A.Aksoy et al.: http://arxiv.org/abs/2605.28332

DENOISER

Performance



Peak-envelope trigger

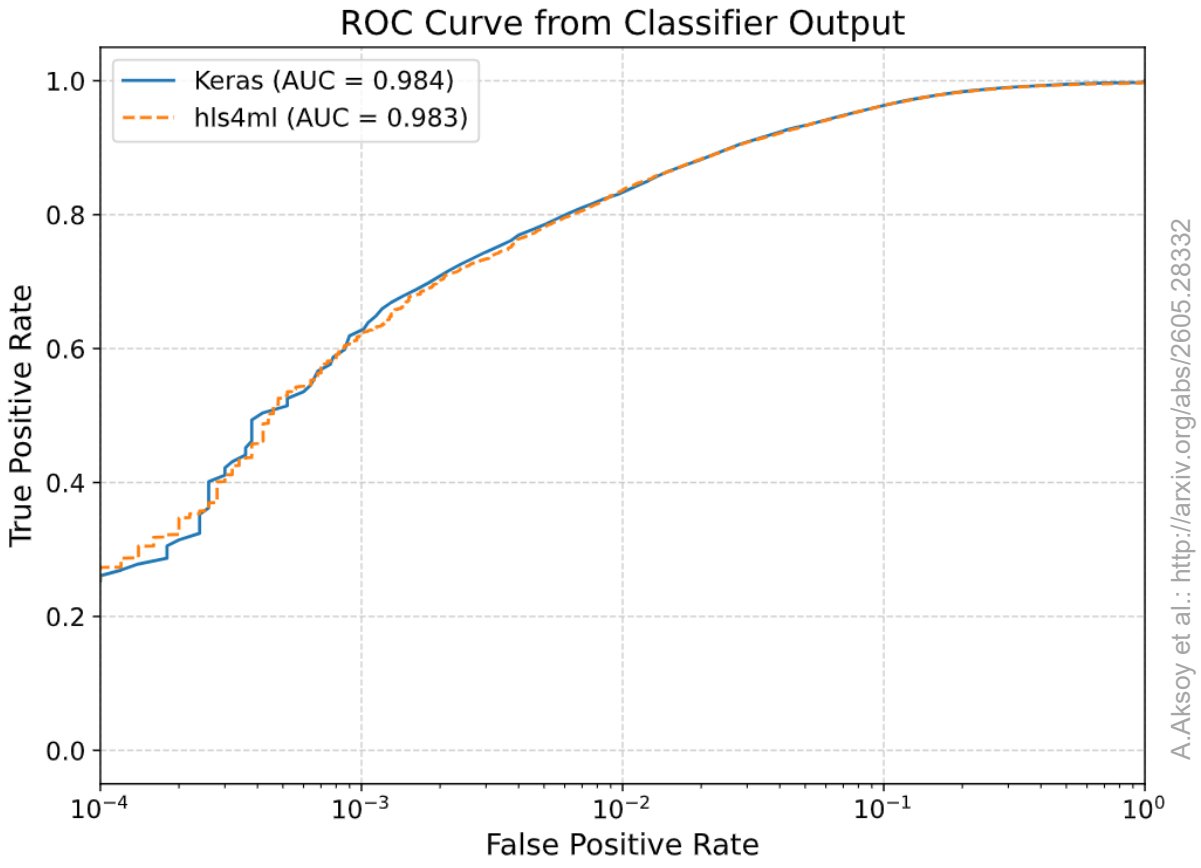
- Determine peak envelope amplitude A_{peak}
- Select threshold θ based on target false positive rate
- Trigger if $A_{peak} > \theta$

FPGA	Latency	BRAM	DSP	FF	LUT
Zynq-7020	7.68 µs	168 60%	45 20%	38,710 36%	34,733 65%
Kintex U040	3.03 µs	168 14%	45 2%	17,042 3%	30,520 12%
Alveo U250	2.30 µs	121 2%	45 ~0%	16,564 ~0%	33,826 1%

→ Denoising improves trigger discrimination

CLASSIFIER

Performance



A.Aksoy et al.: http://arxiv.org/abs/2605.28332

FPGA	Latency	BRAM	DSP	FF	LUT
Zynq-7020	5.87 µs	29 10%	1 ~0%	17,592 16%	21,721 40%
Kintex U040	2.63 µs	29 2%	1 ~0%	10,553 2%	20,188 8%
Alveo U250	1.34 µs	21 ~0%	2 ~0%	8,848 ~0%	20,154 1%

Input for training and evaluation:

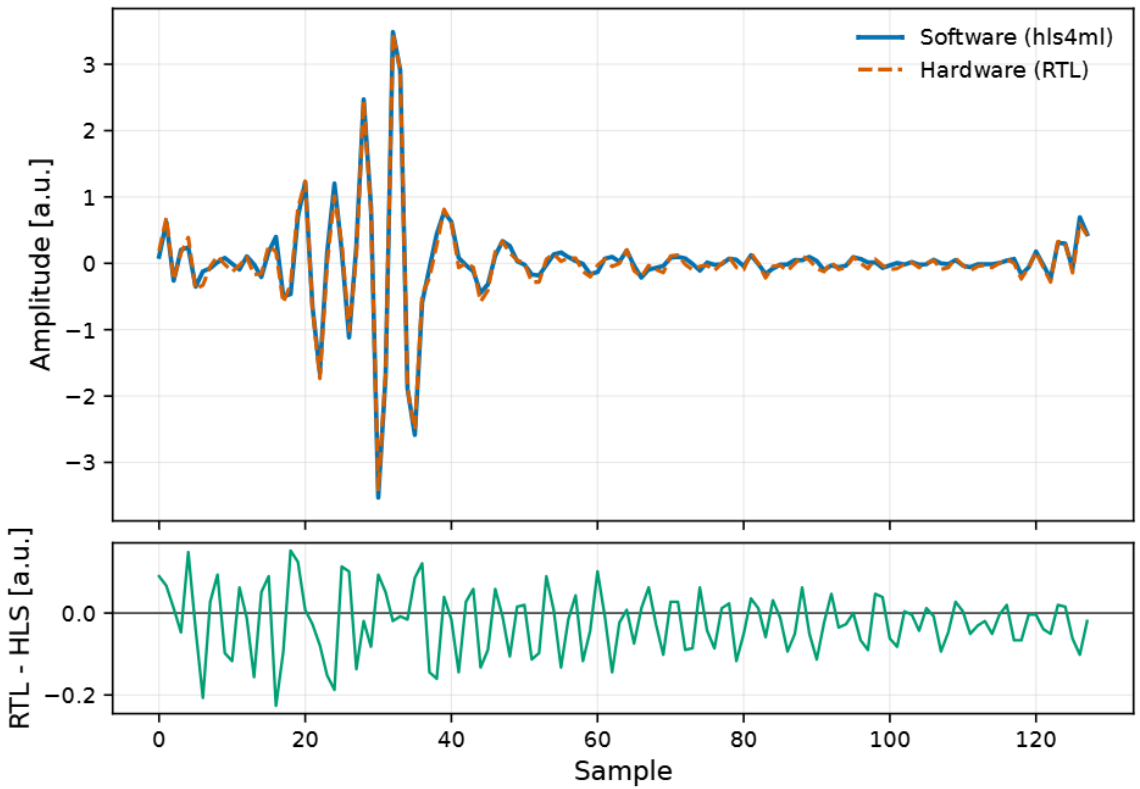
- Raw traces (no denoising yet)

→ Achieves high accuracy (high AUC – Area Under Curve) with low resource usage

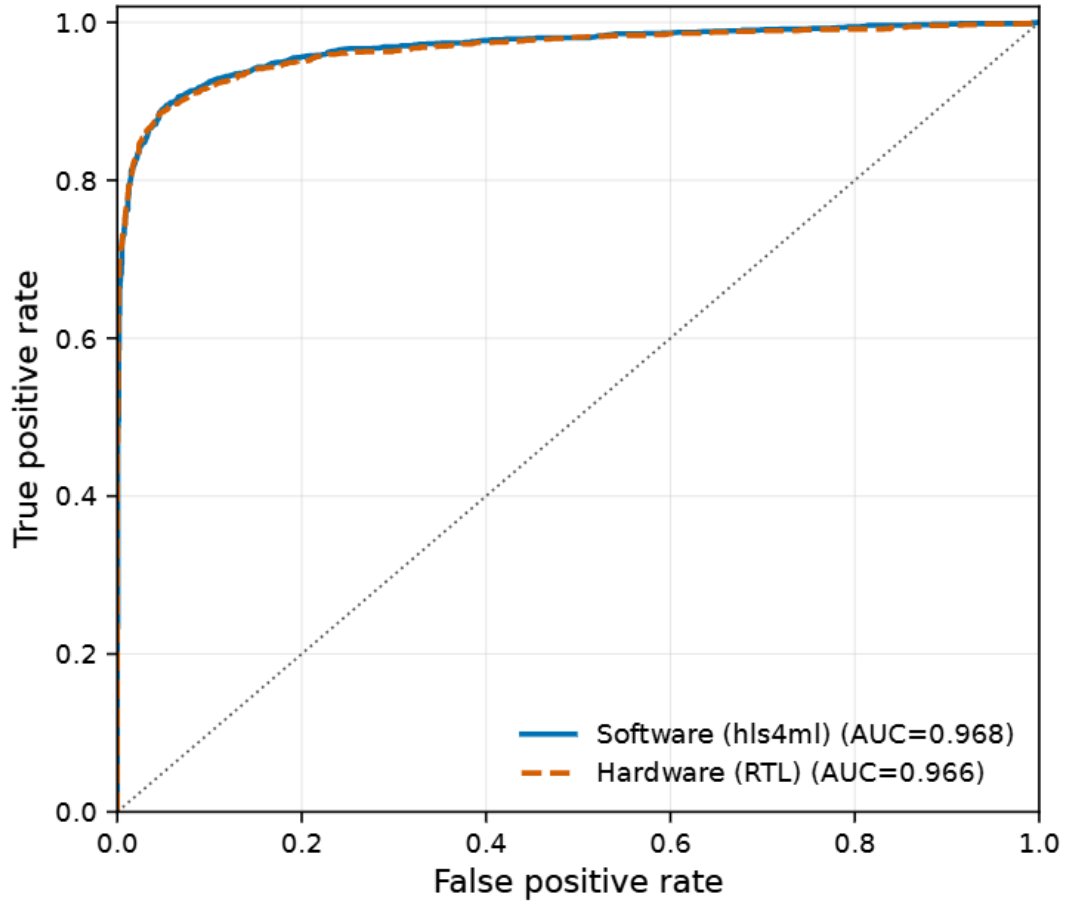
CPPS color meets flavor JÜLICH Forschungszentrum Integrated Computing Architectures ICA

RTL VALIDATION

cocotb testbench



A.Aksoy et al.: http://arxiv.org/abs/2605.28332



→ Excellent agreement between RTL simulation and software performance

HYPERPARAMETER OPTIMIZATION - HYPERBAND

- Training a model fully is expensive

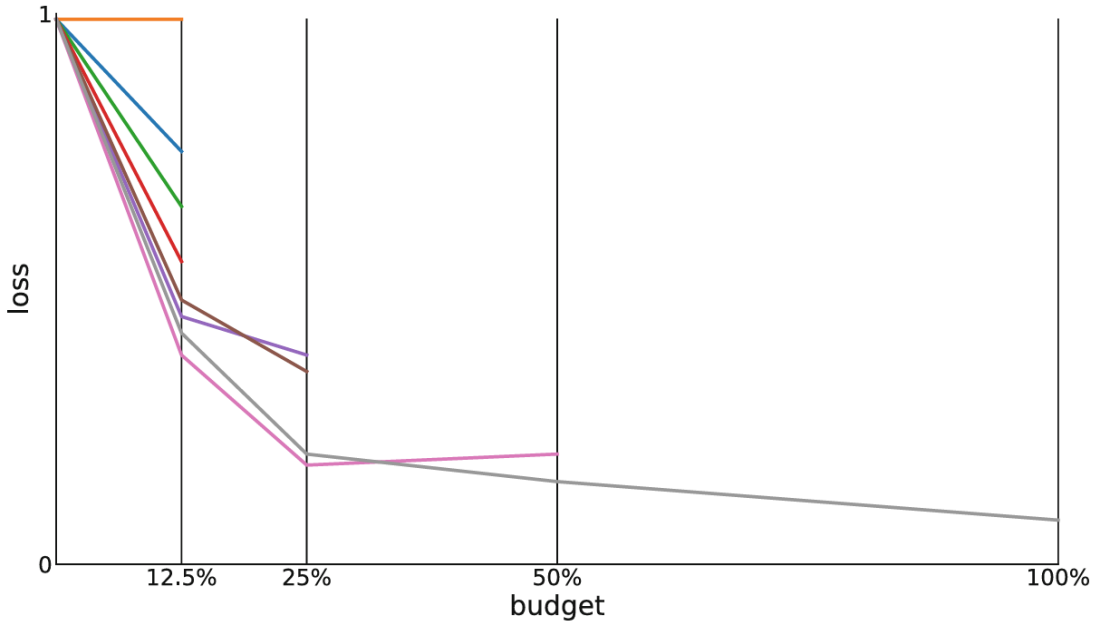
→Train **many** configurations a **little bit**

Successive Halving

- Start with **many hyperparameter configurations**
- Give each a **small budget** (e.g. few epochs)
- **Evaluate** them
- Keep the **best fraction**
- **Increase budget** for survivors
- **Repeat** until one candidate left

  → *Choose* **many configs/tiny budget** or **few configs/large budget**

HYPERBAND (HB)



HB runs **many schedules** with **different budgets**

Successive Halving for 8 different configurations

- Start with low budget ($^1/_8$)
- Drop half configs after evaluating
- Increase budget for the remaining by same factor

→ Automatically optimize the model, without manual tuning

ENVELOPE THRESHOLD TRIGGER

Purpose and decision rule

- Classical amplitude-based trigger used as benchmark
- Compute peak envelope **amplitude** A_{peak}
- Select decision **threshold** θ based on target **FPR**
- Trigger if $\mathrm{A_{peak}} > \theta$

Evaluation

- ROC curve (TPR vs. FPR)

HARDWARE IMPLEMENTATION

- At sampling rate of ~180 – 250 MHz
  → trace length of ~711 – 512 ns
- Inference of trigger ~10 µs

Possible solutions:

1. Implement multiple models on an FPGA

   Run in parallel

   → Continuous operation at higher resource usage

2. Keep one model

   Start new inference with same model with randomly sampled trace

   → Dead time affected operation

BINARY NEURAL NETWORKS

- Neural networks with **binary weights and activations** (±1)
- **Typical methods:**
  - Binarization of weights and activations
  - Straight-Through Estimators (STE) for backpropagation
  - Specialized training strategies to recover accuracy

→**Much lower memory usage** at the cost of accuracy

→**Faster inference** compared to full-precision networks