

# FPGA-Deployed VAE for real-time SXR electron temperature reconstruction in RFX-mod2

How to reconstruct in real time the electron temperature of a fusion experiment in case of diagnostic failure

Luca Orlandi, A. Rigoni Garola, L. Saccaro, P. Franz, M. Gobbin, L. Piron, R. Cavazzana · Consorzio RFX – Università di Padova

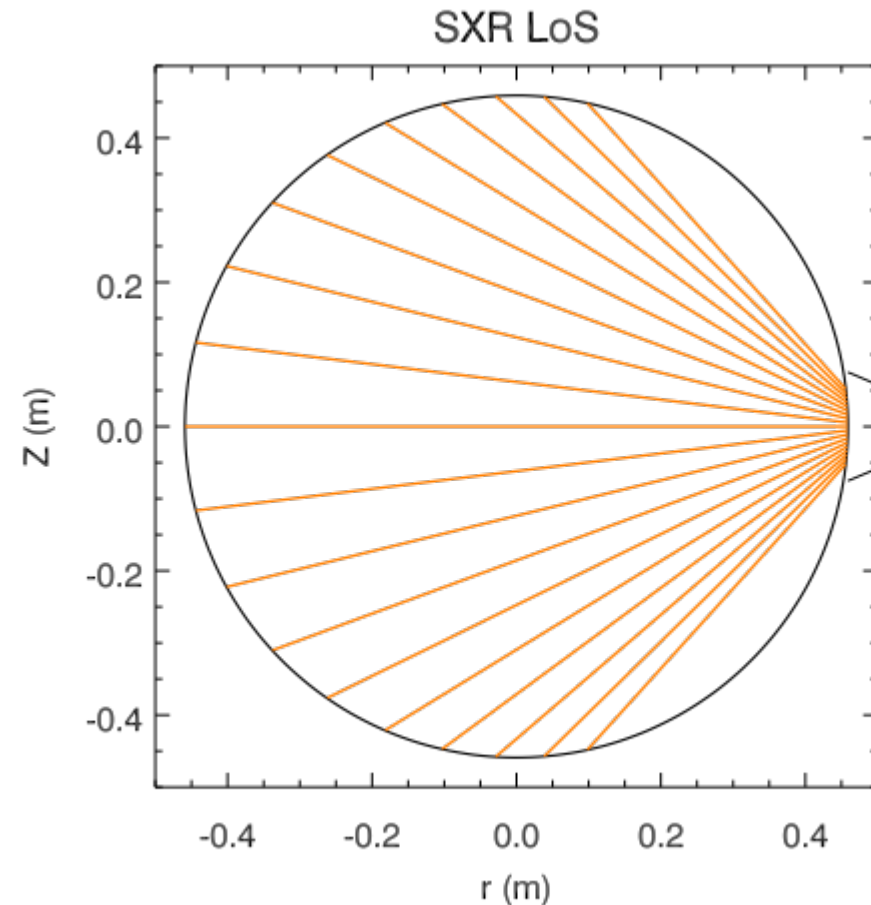
25<sup>th</sup> IEEE Real Time Conference · La Biodola · 27.05.2026

---

# Electron Temperature measurements

Electron temperature ( $T_e$ ) is a key parameter in fusion experiments.

- **Diagnostic system:** data is acquired through the Soft X-ray diagnostic
- **Problem:** the acquired data is noisy and sometimes the electron temperature computation explodes leading to missing data
- **Solution:** a **Neural Network** deployed on **FPGA** can reconstruct the temperature profile, also supplying the missing data points

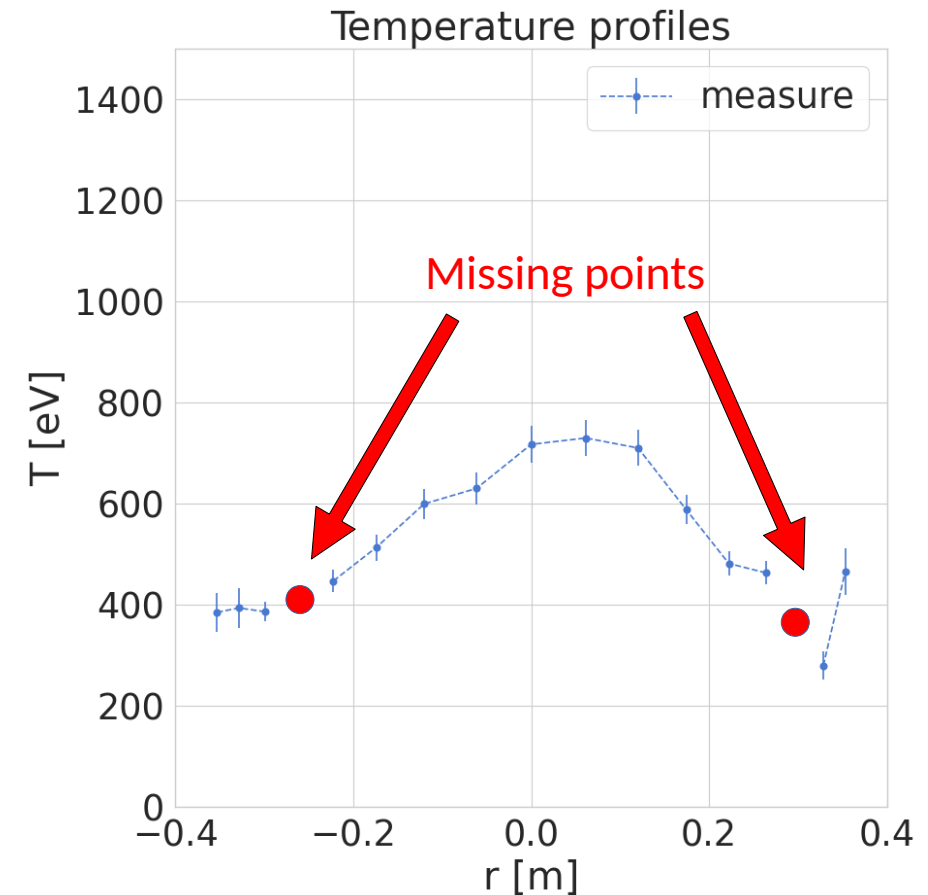


Lines of sight of the Soft X-Ray diagnostic in the RFX-mod experiment.

# Problems with the Diagnostic

Electron temperature ( $T_e$ ) is a key parameter in fusion experiments.

- **Diagnostic system:** data is acquired through the Soft X-ray diagnostic
- **Problem:** the acquired data is noisy and sometimes the electron temperature computation explodes leading to missing data
- **Solution:** a **Neural Network** deployed on **FPGA** can reconstruct the temperature profile, also supplying the missing data points

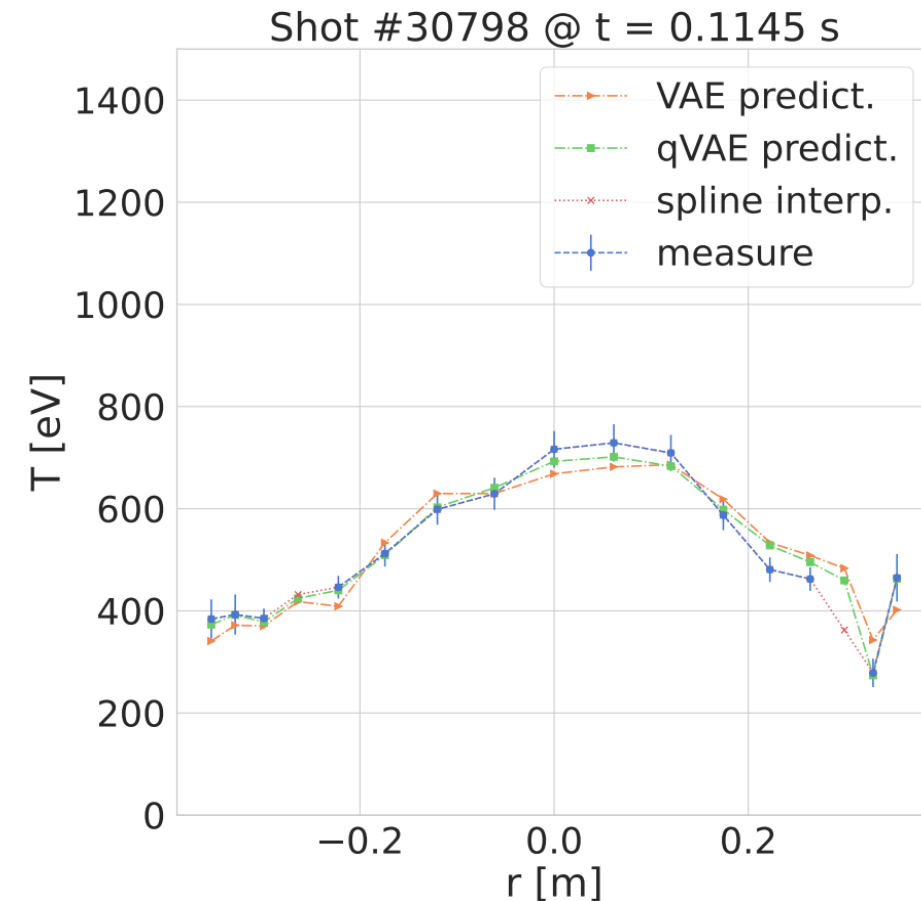


# Reconstructing Electron Temperature Profiles

Electron temperature ( $T_e$ ) is a key parameter in fusion experiments.

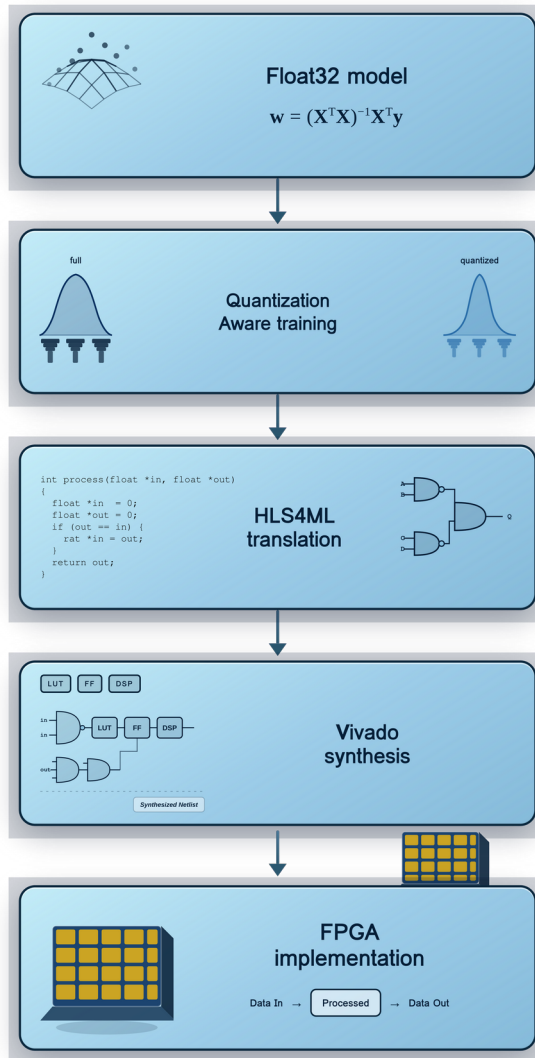
- **Diagnostic system:** data is acquired through the Soft X-ray diagnostic
- **Problem:** the acquired data is noisy and sometimes the electron temperature computation explodes leading to missing data
- **Solution:** a **Neural Network** deployed on **FPGA** can reconstruct the temperature profile, also supplying the missing data points

Metti immagine dell'architettura



Reconstruction of a temperature profile with float32 model, quantized model, B-spline and raw measure.

# Work Flow



1) Train a float32 model

2) Apply quantization aware training with **HGQII**

3) Translate to HLS with **HLS4ML**

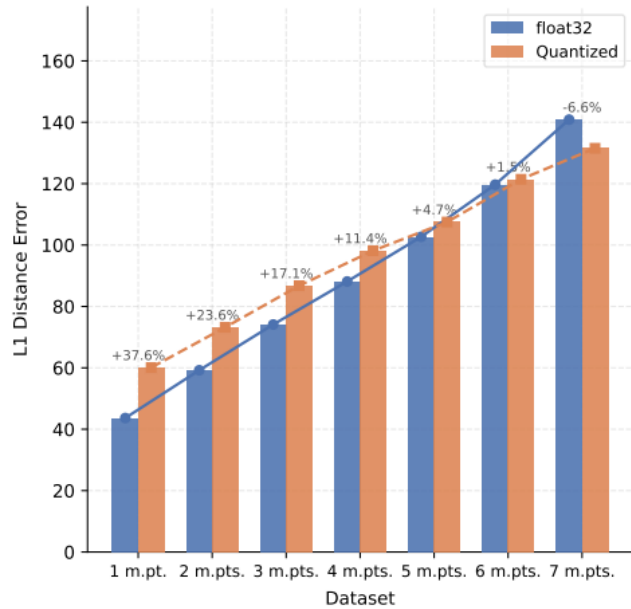
4) Synthesize using Vivado

5) Deploy on FPGA

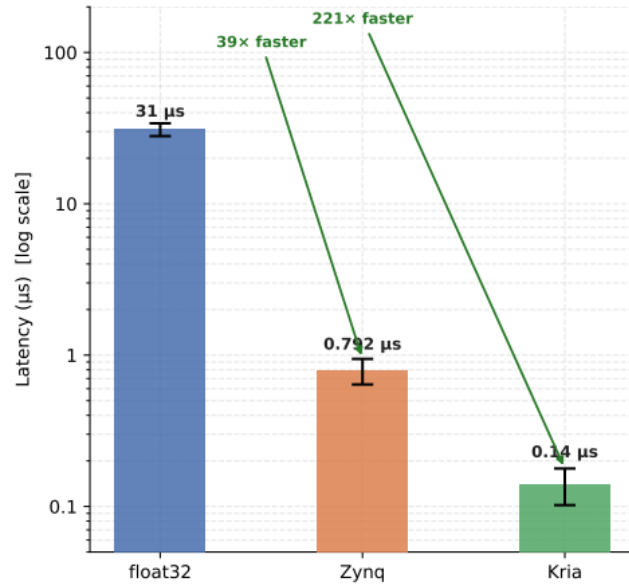
# Accuracy and Resource utilization on FPGA

Model Performance: float32 vs. INT8 Quantized

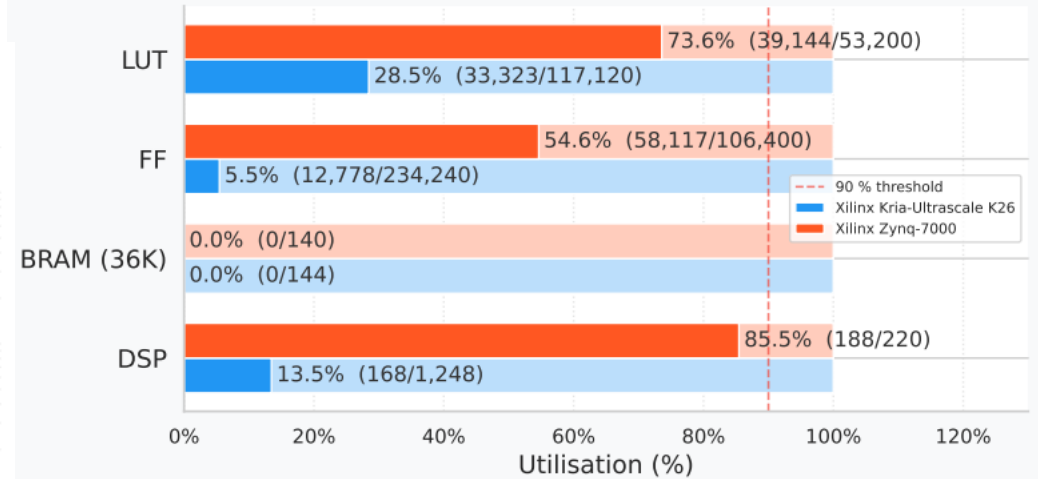
L1 Distance Comparison (float32 vs. Quantized)



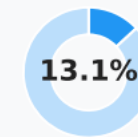
Inference Latency (float32 vs. Zynq vs. Kria)



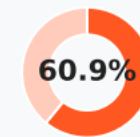
FPGA Resource Utilisation  
Xilinx Kria-Ultrascale K26 vs Xilinx Zynq-7000



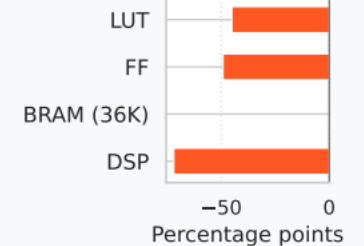
Xilinx Kria-Ultrascale Overall Utilisation

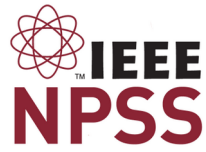


Xilinx Zynq-7000 Overall Utilisation



Δ Utilisation (Kria-Ultrascale - Zynq-7000)





UNIVERSITÀ  
DI PADOVA



# Thank you!

Also many thanks to the co-authors:

A. Rigoni Garola, L. Saccaro, P. Franz, M. Gobbin, L. Piron, R. Cavazzana

[Luca.orlandi@igi.cnr.it](mailto:Luca.orlandi@igi.cnr.it)