Towards Energy-Efficient Split Computing

A Hardware-Software Co-Design Perspective

Daniel May*, Alessandro Tundo*, Shashikant Ilager†, Ivona Brandic*

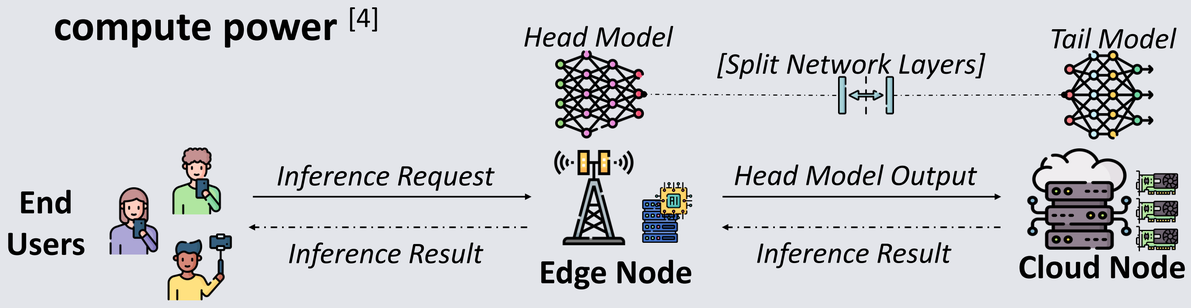
*TU Wien, Austria

†University of Amsterdam, The Netherlands

Motivation & Challenges

Motivation

- Edge ML enables **privacy-aware, low-latency** apps [3]
- Split computing enables joint use of **edge proximity** and **cloud compute power** [4]



Challenges

- Selecting **optimal split points** is non-trivial [2]
- Latency–energy trade-offs are **hardware-dependent** [1, 6]
- **Dynamic network conditions** complicate runtime decisions [7]
- Varying **QoS targets** must be met reliably

Contribution

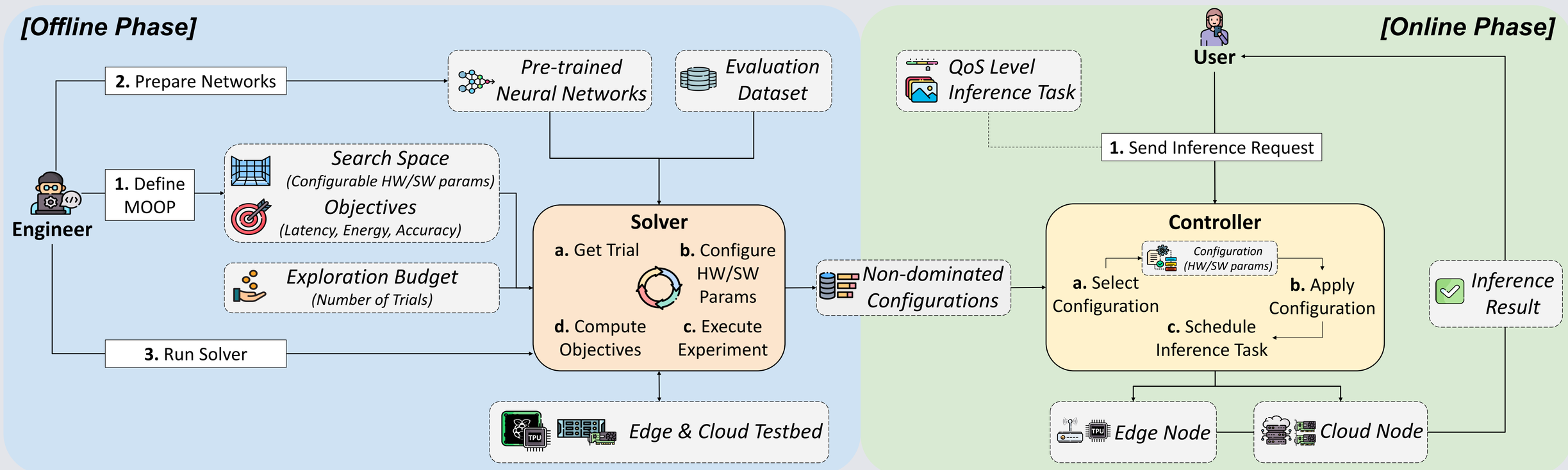
Framework Overview

- A **hardware-software co-design** framework for splitting ML inference across edge and cloud
- **Offline phase**: Multi-objective optimization to explore latency–energy trade-offs
- **Online phase**: Runtime selection of optimal configuration based on QoS requirements

Jointly tunes parameters:

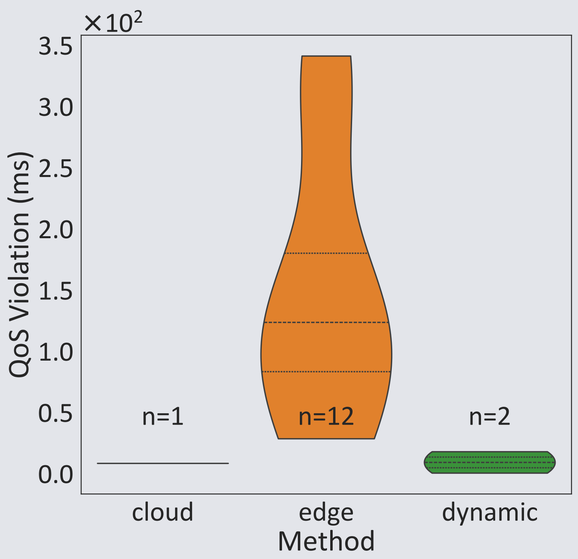
- **Split layer**: Placement of neural network boundary
- **CPU frequency**: Dynamic voltage and frequency scaling (DVFS)
- **Accelerator usage:** Enable/disable TPU or GPU
- **Accelerator frequency:** Select optimal clock speeds for TPU

Proposed Framework

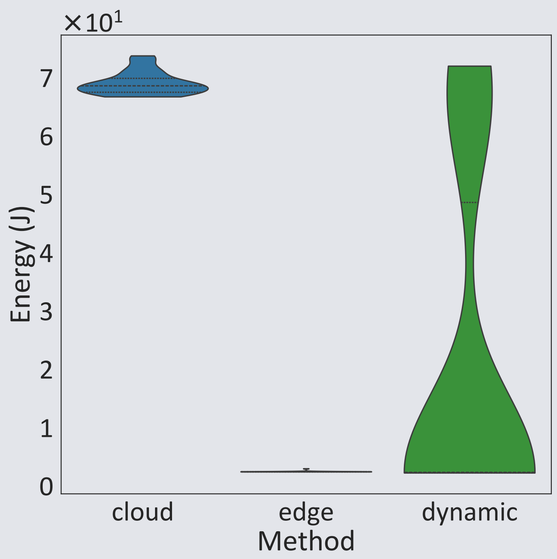


Preliminary Results

- **Model:** VGG16 (image classification)
- **Devices:** Raspberry Pi 4B + Coral TPU (edge), Tesla V100 (cloud)
- **Workload:** 50×1,000 inferences with QoS latency targets
- **Energy:** Measured via power meters
- **Baselines**: Cloud-only and edge-only vs. our dynamic method



QoS Violations



Energy Consumption

Discussion

- Our dynamic approach **reduces QoS violations** compared to edge-only execution
- It also consumes **less energy** than inference fully on the cloud
- The system adapts to **varying workloads** and **network dynamics**
- Results show the promise of **hardware-software co-design** for edge-cloud ML

Extended results available in our preprint [5]

SCAN ME

References

[1] Hanafy, W. et al. Design Considerations for Energy-efficient Inference on Edge Devices. ACM e-Energy, 2021.
[2] Kang, Y. et al. Neurosurgeon: Collaborative Intelligence Between Cloud and Edge. ASPLOS, 2017.
[3] Lujic, I. et al. Increasing Traffic Safety with Real-Time Edge Analytics and 5G. EdgeSys, 2021.
[4] Matsubara, Y. et al. Split Computing and Early Exiting for DL Apps: Survey and Challenges. ACM Comput. Surv., 2023.
[5] May, D. et al. DynaSplit: A Hardware-Software Co-Design Framework. arXiv:2410.23881, 2024.
[6] Tang, Z. et al. GPU DVFS Impact on Energy and Performance of Deep Learning. ACM e-Energy, 2019.
[7] Zhang, S. et al. Real-time Cooperative Deep Inference over Cloud and Edge. IMWUT, 2020.

This poster has been designed using images from Flaticon.com. Experiments presented in this poster were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr) This work was supported by a netidee scholarship. This research has been partially funded through the projects: Transprecise Edge Computing (Triton), Austrian Science Fund (FWF), DOI: 10.55776/ P36870; Trustworthy and Sustainable Code Offloading (Themis), FWF, DOI: 10.55776/ PAT1668223; Satellite-based Monitoring of Livestock in the Alpine Region (Virtual Shepherd), Austrian Research Promotion Agency (FFG), Austrian Space Applications Programme (ASAP) 2022 # 5307925; Digital Twin for LoRaWAN Agriculture Systems, Steirische Wirtschaftsförderung (SFG), Ideen!Reich XS 1.000.073.260.

Supported by:

danielmay.me
daniel.may@tuwien.ac.at