Towards Energy-Efficient Split Computing: A Hardware-Software Co-Design Perspective

Daniel May*, Alessandro Tundo*, Shashikant Ilager†, Ivona Brandic*
*Institute of Information Systems Engineering, TU Wien, Austria
†Informatics Institute, University of Amsterdam, The Netherlands

Abstract

Edge ML faces resource and energy constraints, requiring optimized split computing, which partitions inference between edge and cloud. We propose a two-phase framework combining offline optimization and dynamic scheduling. It jointly configures split points and hardware settings to balance energy and latency.

1 Introduction

Edge computing provides resources at the network edge, enabling latency-sensitive and privacy-aware applications that often rely on Machine Learning (ML) models for prediction and analytics [4]. This paradigm, known as Edge AI [1], supports smart applications, but faces significant challenges due to the resource and energy constraints of edge devices.

To address these challenges, edge applications can distribute tasks among edge nodes, user devices, and cloud resources [9]. This method keeps latency-sensitive processing at the edge, whilst utilizing the cloud's scalability. When employing neural networks (NNs) for inference tasks, their layers can be split into multiple sections and executed across edge and cloud based on current needs. This approach, known as split computing [5], can enhance performance by optimizing the use of both edge and cloud strengths.

Effectively implementing split computing is challenging. In fact, selecting split the best points is a complex task due to nonlinear dependencies in the resulting latency and energy usage [3]. Additionally, dynamic runtime conditions in edge environments further increase problem complexity [10]. Finally, both network conditions and hardware characteristics (e.g., CPU tuning via dynamic voltage frequency scaling (DVFS) and hardware accelerators) can significantly impact latency and energy consumption [2, 8].

These factors, combined with the need to satisfy dynamic Quality of Service (QoS) requirements, make jointly optimizing split points and hardware configurations particularly challenging. Addressing these complexities is crucial to enable energy-efficient inference in edge-cloud environments.

2 Proposed Approach

We propose a hardware-software co-design framework for energy-efficient, latency-aware inference in a single edge-cloud setup. It dynamically selects optimal split points and hardware configurations like CPU frequencies and whether to employ hardware accelerators. This method leverages edge and cloud resources to meet QoS requirements and reduce energy consumption.

Figure 1 illustrates the functionalities of the framework organized in two phases. In the *Offline Phase*, we set up a Multi-Objective Optimization Problem (MOOP) to reduce latency and energy consumption while ensuring inference accuracy. Then, a meta-heuristic optimization algorithm (i.e., NSGA-III) is used to identify a set of non-dominated configurations (i.e., Pareto front solutions). These configurations, precomputed from empirical testing, offer suitable trade-offs across a complex hardware-software parameter space.

During the *Online Phase*, the precomputed configurations are applied dynamically as per incoming requests and QoS needs. The system opts for the most energy-efficient configuration that meets latency demands or, if needed, the quickest one to minimize QoS breaches. This framework adeptly manages varying workloads and network conditions by adjusting edge and cloud parameters in real time.

This two-phase method effectively tackles the complexity of dynamic NN inference in distributed settings. It merges offline optimization with real-time adaptability, reducing energy use, meeting latency needs, and offering a practical solution for present-day edge-cloud inference.

3 Preliminary Results

We evaluated our framework using the VGG16 [7] neural network, a widely used model for computer vision, on a testbed employing a Raspberry Pi 4B, a Google Coral TPU, and a cloud node equipped with an NVIDIA Tesla V100 GPU[1]. Energy consumption was measured using digital power meters, ensuring more accurate results than estimation-based approaches. We evaluated energy consumption and QoS violations for 50 requests, each consisting of 1,000 inference tasks for reliable measurements. We modeled QoS latency limits as an exponential distribution and compared the results with *edge-only* and *cloud-only* baselines.

Figure 2a displays QoS violations for the baselines and our dynamic configuration approach, shown as violin plots with density and quartiles, with n indicating total violations. The edge-only baseline has high violations, with 25% of requests exceeding latency limits. The cloud-only baseline incurs one

[1]Experiments presented in this paper 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).

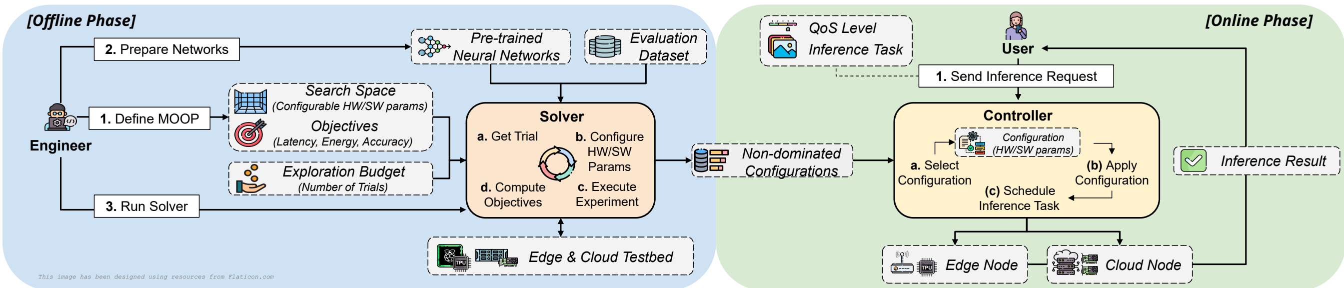
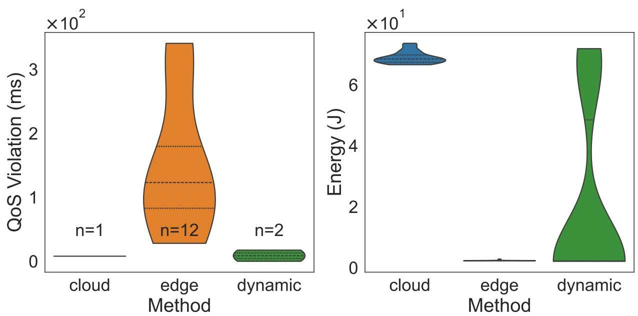


Figure 1. An overview of the proposed framework.



(a) QoS Violations **(b)** Energy Consumption

Figure 2. Preliminary results for VGG16.

violation. Our dynamic method reduces the violation rate to 4%, with a median exceedance of 10 ms, sustaining QoS.

Figure 2b shows the results of the combined energy consumption of the edge and the cloud. The cloud-only method uses the most energy (median 68 J), while the edge-only method is much more efficient (median <3 J). Our dynamic approach also has a median energy use of <3 J and adapts to QoS, sometimes using up to 72 J.

Summary and Conclusion: Our preliminary results show the potential of dynamic split computing to balance latency and energy efficiency while adapting to variations in workload. Compared to edge-only execution, our approach significantly reduces QoS violations while consuming less energy than cloud-only computation. These findings highlight the need for more experiments to study the full impact of edge-cloud inference, which we already have started including in a pre-print available on arXiv [6].

Acknowledgments

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.

References

[1] Aaron Yi Ding, Ella Peltonen, Tobias Meuser, Atakan Aral, Christian Becker, Schahram Dustdar, Thomas Hiessl, Dieter Kranzlmüller, Madhusanka Liyanage, Setareh Maghsudi, Nitinder Mohan, Jörg Ott, Jan S. Rellermeyer, Stefan Schulte, Henning Schulzrinne, Gürkan Solmaz, Sasu Tarkoma, Blesson Varghese, and Lars C. Wolf. 2022. Roadmap for edge AI: a Dagstuhl perspective. *Comput. Commun. Rev.* 52, 1 (2022), 28–33.

[2] Walid A. Hanafy, Tergel Molom-Ochir, and Rohan Shenoy. 2021. Design Considerations for Energy-efficient Inference on Edge Devices. In *e-Energy: ACM International Conference on Future Energy Systems.* 302–308.

[3] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor N. Mudge, Jason Mars, and Lingjia Tang. 2017. Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge. In *ASPLOS: ACM International Conference on Architectural Support for Programming Languages and Operating Systems.* 615–629.

[4] Ivan Lujic, Vincenzo De Maio, Klaus Pollhammer, Ivan Bodrozic, Josip Lasic, and Ivona Brandic. 2021. Increasing Traffic Safety with Real-Time Edge Analytics and 5G. In *EdgeSys: ACM International Workshop on Edge Systems, Analytics and Networking.* 19–24.

[5] Yoshitomo Matsubara, Marco Levorato, and Francesco Restuccia. 2023. Split Computing and Early Exiting for Deep Learning Applications: Survey and Research Challenges. *ACM Comput. Surv.* 55, 5 (2023), 90:1–90:30.

[6] Daniel May, Alessandro Tundo, Shashikant Ilager, and Ivona Brandic. 2024. DynaSplit: A Hardware-Software Co-Design Framework for Energy-Aware Inference on Edge. (2024). arXiv:2410.23881

[7] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR: International Conference on Learning Representations.*

[8] Zhenheng Tang, Yuxin Wang, Qiang Wang, and Xiaowen Chu. 2019. The Impact of GPU DVFS on the Energy and Performance of Deep Learning: an Empirical Study. In *e-Energy: ACM International Conference on Future Energy Systems.* 315–325.

[9] Junhua Wang, Kai Liu, Bin Li, Tingting Liu, Ruoguang Li, and Zhu Han. 2020. Delay-Sensitive Multi-Period Computation Offloading with Reliability Guarantees in Fog Networks. *IEEE Trans. Mob. Comput.* 19, 9 (2020), 2062–2075.

[10] Shigeng Zhang, Yinggang Li, Xuan Liu, Song Guo, Weiping Wang, Jianxin Wang, Bo Ding, and Di Wu. 2020. Towards Real-time Cooperative Deep Inference over the Cloud and Edge End Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2 (2020), 69:1–69:24.