Poster: Towards a Performance-Driven Device-Edge-Cloud Relationship

Pragya Sharma[1], Brian Wang[1], Xiaomin Ouyang[1], Rahal Nanayakkara[1], Bharathan Balaji[2], Paulo Tabuada[1], Mani B. Srivastava[†1,2]

[1]Electrical and Computer Engineering, University of California Los Angeles, [2]Amazon USA

EXTENDED ABSTRACT

Real-time cyber-physical systems (CPS) rely on Perception-Cognition-Actuation (PCA) pipelines to enable autonomous observation, decision-making, and action execution. Closed-loop PCA systems utilize feedback-driven control to iteratively adapt actions in response to real-time environmental changes whereas open-loop PCA systems execute single actions without iterative feedback. The overall performance of these systems is inherently tied to the models selected for each pipeline component. Recent advancements in neural networks, particularly for perception tasks, have substantially enhanced CPS capabilities but have introduced significant complexity into the PCA pipeline. While traditional research [1] often evaluates perception models in static, controlled settings, it fails to account for the cascading latency and accuracy trade-offs that manifest across interconnected PCA modules in dynamic, real-time applications. Additionally, the proliferation of distributed device-edge-cloud architectures [2] has expanded computational possibilities but introduced new challenges in balancing latency and accuracy with resource constraints. The holistic impact of model selection, deployment platforms, and network conditions on application performance in real-time scenarios remains under-explored.

To address this gap, we analyze the impact of these factors on application performance across two representative CPS scenarios: leader-follower navigation and perception-driven target interception. Using a custom testbed (Fig. 1(left)) with realistic network latencies and heterogeneous computational platforms, we evaluated key performance indicators for each application. In the closed-loop PCA scenario, we analyzed performance across four model and platform configurations, integrating perception and control models deployed either on-device (co-located with sensors and actuators) or in the cloud. On-device setups utilized smaller, lower-accuracy models, including the YOLOv8s object detector, SORT tracker, and PID controller, prioritizing low latency. In contrast, cloud-based deployments hosted larger, high-accuracy models such as YOLOv8x, DeepSORT tracker, and MPC controller. We used two key metrics to assess performance: following distance error (FDE) and jerk profile. The on-device perception and control configuration demonstrated

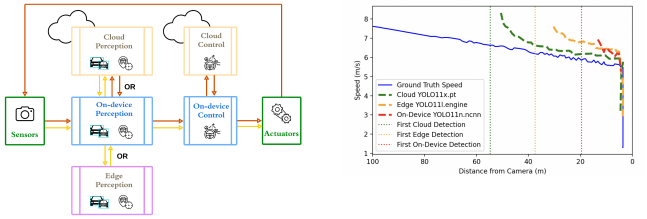

Figure 1: Illustration of Distributed Closed-Loop and Open-Loop PCA pipeline (left); Open-Loop PCA experimental results (right)

the best performance, achieving the lowest FDE and a stable jerk profile despite the reduced accuracy of smaller models. The rapid response time of this setup effectively compensates for any perception inaccuracies, enabling timely corrective actions that minimize overall FDE. In contrast, the cloud-based perception and control configuration exhibited unstable behavior, with larger FDE and occasional oscillations. Although cloud-based perception models provided more accurate results, the inherent network latency caused the data to become stale by the time it reached the actuator, undermining its utility. These findings challenge the prevailing assumption that deploying high-accuracy models on computationally powerful platforms inherently improves application performance. Instead, we find that in closed-loop settings where iterative actions reshape the environment, on-device deployment of lower-accuracy models delivers superior performance by minimizing response times.

In the open-loop PCA framework, we evaluated three model configurations — cloud-based YOLO11x, edge-based YOLO11l, and on-device YOLO11n object detectors — using initial detection distance (IDD) and detection accuracy as key metrics. The cloud-based YOLO11x, despite higher network latency, achieved the largest IDD (>50 m) and the highest detection accuracy, leveraging extended observational windows to enhance performance (Fig. 1(right)). In contrast, the on-device YOLO11n demonstrated the shortest IDD (20 m) and the lowest accuracy, providing insufficient lead time for reliable interception. These findings demonstrate that strategically leveraging cloud resources can substantially enhance application performance by enabling earlier and more accurate detections, challenging the belief that high network latencies inherently render cloud-based models unsuitable for real-time tasks. **Summary:** Our findings show that while distributed device-edge-cloud architectures boost overall performance, finding optimal deployment strategies requires an application-specific approach to effectively balance trade-offs between latency, accuracy, and resource utilization.

References

[1] Mingxing Tan et al. 2020. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF CVPR*. 10781–10790.

[2] Yehan Ma and et al. 2020. Exploring edge computing for multitier industrial control. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 11 (2020), 3506–3518.

†The author holds concurrent appointments as Amazon Scholar and Professor at UCLA, but the work in this paper is not associated with Amazon.

Acknowledgment: This research was sponsored in part by DEVCOM ARL under cooperative agreement W911NF1720196, and by the NSF under award CNS-2211301.

HOTMOBILE '25, La Quinta, CA, USA

ACM ISBN 979-8-4007-1403-0/25/02
https://doi.org/10.1145/3708468.3715681