New Problems in Distributed Inference for DNN Models on Robotic IoT

(Anonymous Authors)

***Abstract*—The abstract goes here.**

I. Introduction

The rapid progress in machine learning (ML) techniques has led to remarkable achievements in various fundamental robotic tasks, such as object detection[5], [10], [12], robotic control[7], [18], [21], and environmental perception[2], [8], [19]. To deploy these ML approaches on real-world robots, it is crucial to ensure fast and energy-efficient inference of their deep neural network (DNN) models, given the necessity for swift environmental responses and the constraints imposed by limited battery capacity. However, solely relying on additional computing accelerators (e.g., GPU[13], FPGA[14], SoC[4]) on robots is insufficient for achieving fast and energy-efficient inference due to the slow inference speed caused by the limited computing resources of these accelerators and the increased energy consumption (e.g., 62% for [12] in our experiments) resulting from the computationally-intensive nature of DNN models (e.g., large number of parameters, complex operations).

Inference across multiple devices appears to be a promising approach, which forms the paradigm of distributed inference[**?**]. By offloading a portion of the inference computation onto other GPU devices through Internet of Things for these robots (robotic IoT), distributed inference not only capitalizes on the high computing capabilities of these powerful GPU to expedite the inference process, but also alleviates the local computational burden and consequently lowers energy consumption.

However, all existing parallel methods for distributed inference, which are widely used in data center, are ill-suited in robotic IoT. In data center, there are mainly three kinds of parallel methods: Data parallelism (DP) replicates the model across devices, and lets each replica handle one micro-batch (i.e., splits of a batch which is a set of training inputs for each device); Tensor parallelism (TP) splits a single DNN layer (often too large to fit in one device) over devices; Pipeline parallelism (PP) places different layers of a DNN model over devices and pipelines the inference to reduce devices' idling time. In this paper, we show that several problems are hindering existing parallel methods applying on robotic IoT.

Problem 1 (DP). The small batch sizes inherent to robotic IoT applications (typically 1, 3, etc.) hinder the computation of micro-batches, rendering DP inapplicable for robotic IoT. In data centers, DP is feasible due to the large batch sizes employed (e.g., 16 or 32 images), and when divided into micro-batches, each one still contains several complete inputs (e.g., 2 or 4 images). However, in the context of robotic IoT, real-time performance is crucial for executing robotic tasks, necessitating immediate inference upon receiving inputs, which typically have smaller batch sizes (e.g., 1 or 3 images). Further splitting these inputs would result in micro-batches containing incomplete inputs (e.g., 1/4 or 1/2 of an image), which cannot be computed.

Problem 2 (TP). Robotic IoT is unable to provide the bandwidth required by TP. In TP, synchronizing and combining computation results from different devices after processing the split layer entails significant communication overhead, which necessitates dedicated high-speed interconnects (e.g., 400 Gbps for NVLink[6]) even within data centers. Conversely, robots must prioritize seamless mobility and primarily depend on wireless connections, which inherently possess limited bandwidth. For instance, Wi-Fi 6, the most advanced Wi-Fi technology, offers a maximum theoretical bandwidth of 1.2 Gbps for a single stream[9], which is much lower than the bandwidth required by TP.

Consequently, existing distributed inference approach[**?**] in robotic IoT primarily adopts the PP paradigm, which places different layers of a DNN model across robots and GPU devices. However, PP in data centers mainly enhances inference throughput via pipeline execution, rather than reducing the completion time of a single inference, which is the most critical requirement in robotic IoT. Therefore, existing methods for robotic IoT primarily focus on strategically placing model layers to achieve fast and energy-efficient inference, capitalizing on the fact that the number of parameters in DNN models typically decreases layer by layer. Allocating more layers to robots reduces transmission volume, while assigning more layers to GPU devices accelerates the computation phase of inference. These methods strike a balance on layer partition based on application-specific inference speed requirements and energy consumption demands.

Problem 3 (PP). Existing methods based on PP faces significant challenges due to transmission bottlenecks on robotic IoT. The bandwidth available of wireless networks is further exacerbated in the real world (e.g., only 400Mbps in our experiments) by factors (e.g., movement of the devices[11], [15], occlusion from by physical barriers[3], [17], and preemption of the wireless channel by other devices[1], [16]), and transmission remains a critical bottleneck (e.g., up to 70% of inference time and 50% of energy consumption in our

experiments). Meanwhile, reports from large-scale bandwidth testing services [20] reveal that the median bandwidth of robotic IoT is only 137 Mbps in the US and 153 Mbps in China when GPU devices are deployed in commodity clouds.

Moreover, such transmission bottleneck in PP on robotic IoT cannot be effectively mitigated through PP scheduling. Firstly, PP is unable to overlap the transmission and computation phases within the same inference[**?**]. PP can only overlap these phases across multiple inferences, which increases throughput but not the speed of a single inference. Secondly, reducing transmission volume is not an ideal solution for robotic IoT. Allocating more DNN model layers to robots does reduce the transmission volume and the associated transmission time. However, this approach increases energy consumption and under-utilizes the powerful GPU devices for accelerating inference, and may not necessarily reduce overall inference time.

In this paper, we take the first step to analysis existing parallel methods for distributed inference on robotic IoT. These findings aim to raise research effort to find a new parallel method to speedup distributed inference on robotic IoT, so that the DNN models deployed on real-world robots can achieve fast and energy-efficient inference and it will nurture diverse ML applications deployed on mobile robots in the field.

The rest of the paper is organized as follows: xxxxxx

II. Background

A. DNN models on mobile devices

B. Characteristics of device-cloud collaboration

C. Existing parallel inference strategies in data center

Data parallelism.

Tensor parallelism.

Pipeline parallelism.

D. Related Work

Asynchronous update.

Compressed communication.

III. Problems in Parallel Inference Strategies for DNN Models on Robotic IoT

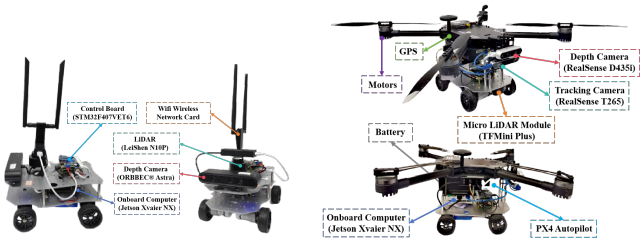
A. Existing Collaborative Inference Strategies

B. Dilemma on Inference Time

C. Dilemma on Energy Consumption

IV. Evaluation

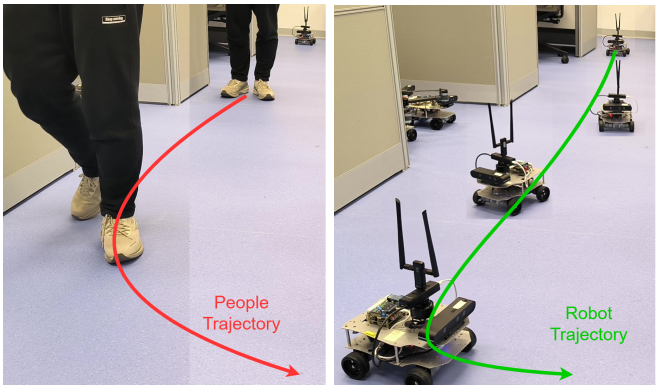
Testbed. The evaluation was conducted on a custom four-wheeled robot (Fig 1a), and a custom air-ground robot(Fig 1b). They are equipped with a Jetson Xavier NX[13] 8G onboard computer serving as the ROS master. The system runs Ubuntu 18.04 and utilizes a SanDisk 256G memory card, with ROS2 Galactic installed for application development and a dual-band USB network card (MediaTek MT76x2U) for wireless connectivity. The Jetson Xavier NX interfaces with a Leishen N10P LiDAR, ORBBEC Astra depth camera, and an STM32F407VET6 controller via USB serial ports. Both LiDAR and depth cameras facilitate environmental perception, enabling autonomous navigation, obstacle avoidance, and SLAM mapping. The host computer processes environmental information in ROS2 Galactic, performing path planning, navigation, and obstacle avoidance before transmitting velocity and control data to corresponding ROS topics. The controller then subscribes to these topics, executing robot control tasks.



(a) Four-wheeled robot (b) Air-ground robot

Fig. 1: The detailed composition of the robot platforms

Real-world Robotic Applications. We evaluated two kinds of typical real-world applications on robots: a real-time people-tracking robotic application on our four-wheeled robot as depicted in Fig 2 and a autonomous navigation on our air-ground robot as depicted in Fig 3.



(a) Targeted people (b) Robot moving trajectory

Fig. 2: A real-time people-tracking robotic application on our robot based on a well-known human pose estimation ML model, KAPAO[12].

V. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] Adame, T., Carrascosa-Zamacois, M., and Bellalta, B. Time-sensitive networking in ieee 802.11 be: On the way to low-latency wifi 7. *Sensors 21*, 15 (2021), 4954.

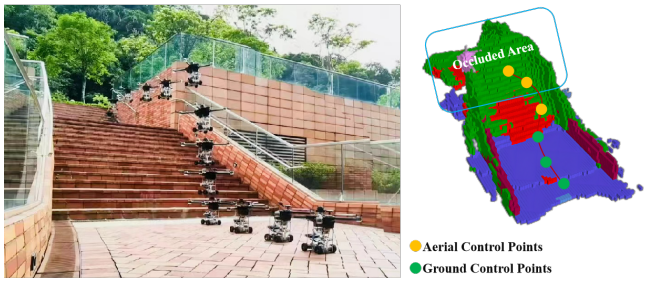


Fig. 3: By predicting occlusions in advance, AGRNav[18] gains an accurate perception of the environment and avoids collisions, resulting in efficient and energy-saving paths.

[2] CAO, A.-Q., AND DE CHARETTE, R. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 3991–4001.

[3] DING, M., WANG, P., LÓPEZ-PÉREZ, D., MAO, G., AND LIN, Z. Performance impact of los and nlos transmissions in dense cellular networks. *IEEE Transactions on Wireless Communications 15*, 3 (2015), 2365–2380.

[4] HONKOTE, V., KURIAN, D., MUTHUKUMAR, S., GHOSH, D., YADA, S., JAIN, K., JACKSON, B., KLOTCHKOV, I., NIMMAGADDA, M. R., DATTAWADKAR, S., ET AL. 2.4 a distributed autonomous and collaborative multi-robot system featuring a low-power robot soc in 22nm cmos for integrated battery-powered minibots. In *2019 IEEE International Solid-State Circuits Conference-(ISSCC)* (2019), IEEE, pp. 48–50.

[5] JOSEPH, K. J., KHAN, S., KHAN, F. S., AND BALASUBRAMANIAN, V. N. Towards open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021), pp. 5830–5840.

[6] LI, A., SONG, S. L., CHEN, J., LI, J., LIU, X., TALLENT, N. R., AND BARKER, K. J. Evaluating modern gpu interconnect: Pcie, nvlink, nv-sli, nvswitch and gpudirect. *IEEE Transactions on Parallel and Distributed Systems 31*, 1 (2019), 94–110.

[7] LI, Q., GAMA, F., RIBEIRO, A., AND PROROK, A. Graph neural networks for decentralized multi-robot path planning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020), IEEE, pp. 11785–11792.

[8] LI, Y., YU, Z., CHOY, C., XIAO, C., ALVAREZ, J. M., FIDLER, S., FENG, C., AND ANANDKUMAR, A. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 9087–9098.

[9] LIU, R., AND CHOI, N. A first look at wi-fi 6 in action: Throughput, latency, energy efficiency, and security. *Proceedings of the ACM on Measurement and Analysis of Computing Systems 7*, 1 (2023), 1–25.

[10] LIU, S., LI, X., LU, H., AND HE, Y. Multi-object tracking meets moving uav. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), pp. 8876–8885.

[11] MASIUKIEWICZ, A. Throughput comparison between the new hew 802.11 ax standard and 802.11 n/ac standards in selected distance windows. *International Journal of Electronics and Telecommunications 65*, 1 (2019), 79–84.

[12] MCNALLY, W., VATS, K., WONG, A., AND MCPHEE, J. Rethinking keypoint representations: Modeling keypoints and poses as objects for multi-person human pose estimation. In *European Conference on Computer Vision* (2022), Springer, pp. 37–54.

[13] NVIDIA. The world's smallest ai supercomputer. https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-series/, 2024.

[14] OHKAWA, T., YAMASHINA, K., KIMURA, H., OOTSU, K., AND YOKOTA, T. Fpga components for integrating fpgas into robot systems. *IEICE TRANSACTIONS on Information and Systems 101*, 2 (2018), 363–375.

[15] PEI, Y., MUTKA, M. W., AND XI, N. Connectivity and bandwidth-aware real-time exploration in mobile robot networks. *Wireless Communications and Mobile Computing 13*, 9 (2013), 847–863.

[16] REN, Y., TUNG, C.-W., CHEN, J.-C., AND LI, F. Y. Proportional and preemption-enabled traffic offloading for ip flow mobility: Algorithms and performance evaluation. *IEEE Transactions on Vehicular Technology 67*, 12 (2018), 12095–12108.

[17] SARKAR, N. I., AND MUSSA, O. The effect of people movement on wi-fi link throughput in indoor propagation environments. In *IEEE 2013 Tencon-Spring* (2013), IEEE, pp. 562–566.

[18] WANG, J., SUN, Z., GUAN, X., SHEN, T., ZHANG, Z., DUAN, T., HUANG, D., ZHAO, S., AND CUI, H. Agrnav: Efficient and energy-saving autonomous navigation for air-ground robots in occlusion-prone environments. In *IEEE International Conference on Robotics and Automation (ICRA)* (2024).

[19] XIA, Z., LIU, Y., LI, X., ZHU, X., MA, Y., LI, Y., HOU, Y., AND QIAO, Y. Scpnet: Semantic scene completion on point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 17642–17651.

[20] YANG, X., LIN, H., LI, Z., QIAN, F., LI, X., HE, Z., WU, X., WANG, X., LIU, Y., LIAO, Z., ET AL. Mobile access bandwidth in practice: Measurement, analysis, and implications. In *Proceedings of the ACM SIGCOMM 2022 Conference* (2022), pp. 114–128.

[21] YANG, Y., JUNTAO, L., AND LINGLING, P. Multi-robot path planning based on a deep reinforcement learning dqn algorithm. *CAAI Transactions on Intelligence Technology 5*, 3 (2020), 177–183.