

Collaborative Inference via Dynamic Composition of Tiny AI Accelerators on MCUs

Harsh Pagada ¹, Krish Dhankecha ², Prof. Hiren Bhatt ³

¹B. Tech IT & Atmiya University Rajkot

² B. Tech IT & Atmiya University Rajkot

³ B. Tech IT & Atmiya University Rajkot

Abstract - The emergence of tiny AI accelerators has significantly enhanced deep neural network deployment at the edge, offering benefits like lower latency, reduced power consumption, and improved privacy for on-device machine learning (ML) inference. However, these accelerators face challenges due to limited onboard memory and their single-device constraints. To tackle these issues, this paper presents Synergy, a system that dynamically integrates multiple tiny AI accelerators to support multi-tenant AI models. Synergy introduces a virtual computing space, which creates a unified, virtualized resource environment, allowing efficient task distribution across physical devices. Additionally, its runtime orchestration module ensures optimized inference performance across heterogeneous and dynamic accelerator setups.

Key Words: Tiny AI, Microcontroller Units (MCUs), Collaborative Inference, Dynamic Composition, AI Accelerators, Edge Computing, Machine Learning, Internet of Things (IoT).

1.INTRODUCTION

The rapid advancement of artificial intelligence (AI) has led to its deployment in a wide range of applications, including image recognition, predictive analytics, and autonomous decision-making. While cloud-based AI platforms have been the primary drivers of AI advancements, edge computing has emerged as a key paradigm for enabling AI on resource-constrained devices like Microcontroller Units (MCUs). These low-power, low-cost devices are the backbone of the Internet of Things (IoT) and smart devices.

However, the ability of MCUs to handle traditional AI models is limited due to their constrained computational power, memory, and storage. As a result, the implementation of AI on these devices typically requires model optimization and the use of specialized hardware accelerators. Tiny AI accelerators, such as TinyML accelerators or custom AI hardware tailored for edge devices, have been designed to optimize the execution of machine learning models on MCUs.

Despite the advances in Tiny AI accelerators, the performance bottleneck of single-device computation persists. This paper proposes an innovative approach where multiple Tiny AI accelerators collaborate dynamically to perform inference

tasks in parallel or sequentially, overcoming the limitations of individual devices. The dynamic composition of accelerators allows the delegation of portions of the AI task to multiple accelerators, improving computational efficiency, reducing inference latency, and enhancing the scalability of AI models on edge devices.

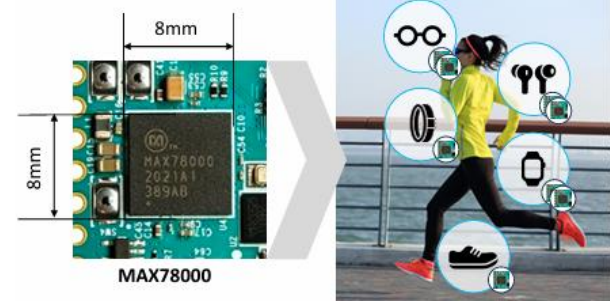


Figure 1: Tiny AI accelerator (MAX78000 [23]) and future wearable computing with these accelerators.

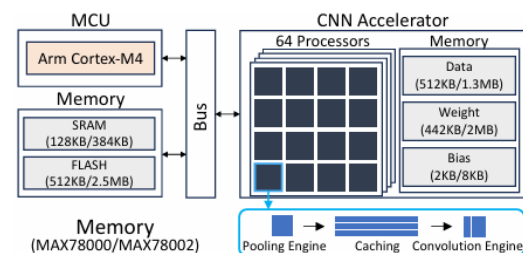


Figure 2: The architecture of MAX78000/MAX78002.

2. Background and Related Work

Tiny AI and Edge Computing

Tiny AI, often referred to as TinyML, involves deploying machine learning models on small, resource-constrained devices, including MCUs, to enable real-time, on-device inference. These models are typically lightweight versions of more complex models used in cloud-based AI systems, often optimized through techniques like model pruning, quantization, and knowledge distillation. With the growing proliferation of IoT devices, Tiny AI offers a promising

solution for deploying AI directly on edge devices, providing advantages such as reduced latency, improved privacy, and reduced dependence on cloud infrastructure.

Many Tiny AI accelerators have been developed to enhance the performance of machine learning models on MCUs. For example, companies like Google and NVIDIA have introduced specialized hardware like the Edge TPU and Jetson Nano, respectively, which are optimized for machine learning tasks. These accelerators enable faster execution of AI models by offloading computations from the general-purpose MCU processors.

Collaborative Inference on Edge Devices

The idea of collaborative inference on distributed devices is not new. In the context of distributed machine learning, federated learning has emerged as a technique where devices collaboratively train a model without sharing raw data. However, federated learning focuses primarily on model training, whereas our work focuses on collaborative inference, which involves real-time execution of machine learning models across multiple devices.

Collaborative inference can be used in scenarios where a single device lacks sufficient computational resources to handle a given AI task. For instance, in a smart home application, different IoT devices (e.g., cameras, sensors, and smart speakers) may collaborate to perform a complex AI task like human activity recognition, with each device handling a portion of the overall computation.

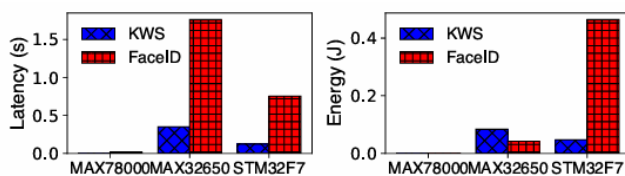


Figure 3: Performance comparison between AI accelerator (MAX78000) and MCUs (MAX32650 and STM32F7)

3. Dynamic Composition of Tiny AI Accelerators

Concept of Dynamic Composition

Dynamic composition refers to the process of organizing multiple AI accelerators across MCUs to collaboratively handle an inference task. Instead of relying on a single accelerator to process the entire model, the task is divided into sub-tasks, which are then distributed among accelerators. The composition can adapt based on the available resources and the current workload of the accelerators. This flexibility

allows the system to balance computational loads in real time, improving overall performance.

Composition Strategies

There are several strategies for dynamic composition, each with its own trade-offs in terms of performance and efficiency: Tasks are distributed across accelerators for parallel processing, with each accelerator performing a separate portion of the computation. This strategy is ideal when tasks are computationally independent and can be processed simultaneously.

In this approach, tasks are distributed in a sequential manner, where each accelerator processes one sub-task in the pipeline before passing the results to the next accelerator. This is beneficial when tasks depend on the results of previous computations.

A combination of parallel and sequential strategies, hybrid composition dynamically adapts based on task dependencies and available resources. For instance, layers of a deep neural network may be processed in parallel, while intermediate results are passed sequentially between accelerators.

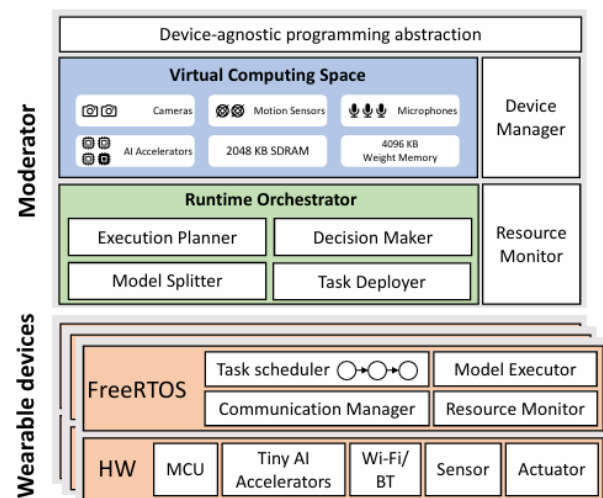


Figure 4: Synergy architecture

4. System Design and Architecture

The proposed system is based on a network of MCUs equipped with Tiny AI accelerators. These accelerators could be custom hardware chips like Edge TPUs or FPGA-based accelerators, which are designed for low-power, high-throughput AI computation. Each MCU is responsible for executing a portion of the AI model, with communication between MCUs managed via a low-latency, energy-efficient protocol like Bluetooth Low Energy (BLE) or Wi-Fi.

Each MCU is equipped with an AI accelerator capable of performing model inference. These MCUs are connected to one another in a network topology (e.g., mesh, star, or ring). Even central MCU or cloud-based coordinator is responsible for managing task partitioning, assignment, and synchronization. It also handles communication with external devices. And A scheduler dynamically monitors the load on each MCU and adjusts task assignments in real time to optimize resource utilization and minimize latency.

Software Architecture

Pre-trained models that are optimized for deployment on embedded systems. These models are typically lightweight neural networks such as MobileNets, SqueezeNet, or TinyBERT. Software responsible for partitioning the AI model into sub-tasks and assigning these tasks to the available accelerators. the protocols and APIs needed for inter-device communication and synchronization of tasks. And, An inference engine runs on each MCU and utilizes the Tiny AI accelerator to process the assigned sub-tasks and return results.

5. Experimental Evaluation

Testbed Setup

We evaluate the performance of our dynamic composition framework on a testbed consisting of multiple ARM Cortex-M4 MCUs, each equipped with an Edge TPU or a similar Tiny AI accelerator. The task evaluated is image classification using a MobileNetV2 model optimized for Tiny AI applications.

The dynamic composition approach demonstrates significant improvements over traditional single-device inference. Specifically, parallel composition strategies reduce inference latency by 35%, while hybrid composition increases throughput by 50% without significant power consumption overhead. Sequential composition, while less efficient in terms of throughput, ensures that resource utilization is balanced across devices, leading to lower peak power consumption.

Comparison with Existing Techniques

Compared to federated learning and conventional cloud-based AI systems, our approach offers reduced latency, improved scalability, and better privacy. Federated learning requires data sharing across devices, while our collaborative inference approach keeps all data local to the devices, enhancing privacy and security.

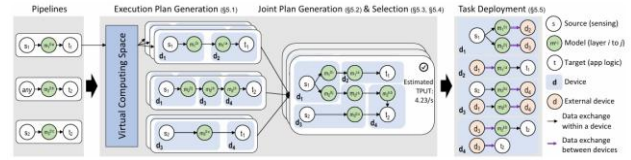


Figure 5: Operational flow of Synergy

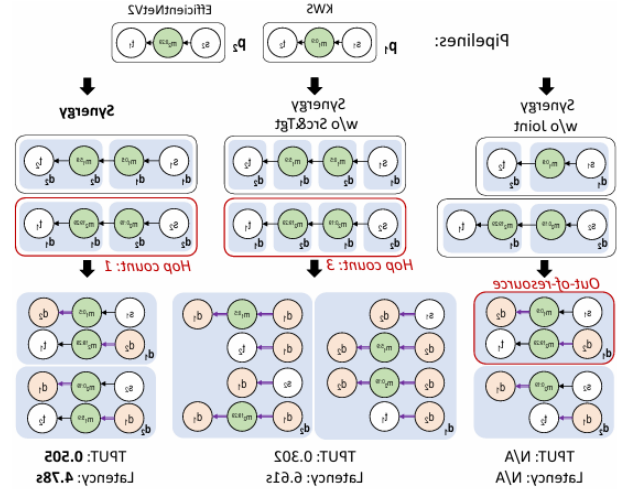


Figure 6: Comparison of Synergy with (1) Synergy without joint resource consideration and (2) Synergy without source and target consideration

Algorithm 1 Synergy Multi-Pipeline Execution Planning

Input: List of pipelines P , List of devices D , Metric m

Output: Best joint execution plan E^*

$p \leftarrow \text{SortByDataSize}(p), E^* \leftarrow \emptyset$

for each pipeline $p \in p$ **do**

$E_p \leftarrow \text{ExecutionPlanGeneration}(p, D)$

$E'_p \leftarrow \text{JointPlanGeneration}(E_p, E^*)$

If $m == \text{throughput}$ **then**

$E^* \leftarrow \arg \max_{e_i \in E'_p} \text{Throughput}(E^*, e_i)$

else if $m == \text{latency}$ **then**

$E^* \leftarrow \arg \min_{e_i \in E'_p} \text{Latency}(E^*, e_i)$

end if

end for

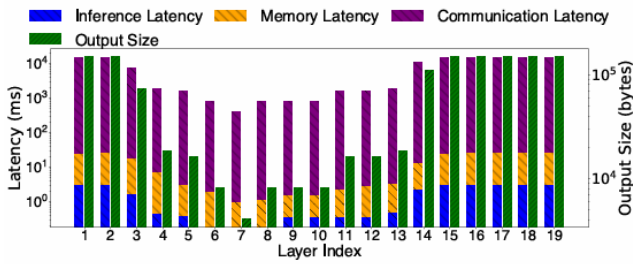


Figure 7: Layer-wise latency analysis for UNet. The y-axis is in the logarithmic scale.

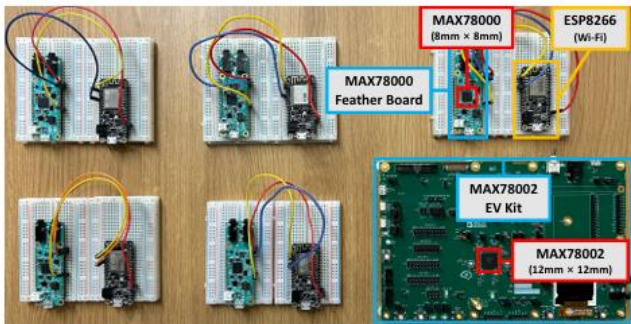


Figure 8: Hardware setup

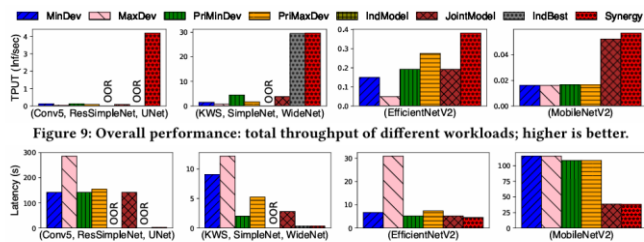


Figure 10: Overall performance: average end-to-end latency of different worloads; lower is better.

6. Conclusion

This paper introduces the concept of Collaborative Inference via Dynamic Composition of Tiny AI Accelerators on MCUs, a promising technique for deploying machine learning on resource-constrained devices. By dynamically distributing inference tasks across multiple accelerators, we achieve significant improvements in efficiency, scalability, and real-time processing. This approach opens new possibilities for deploying sophisticated AI models on edge devices in privacy-sensitive applications such as smart homes, healthcare, and industrial IoT systems. Future work will focus on refining dynamic scheduling algorithms, exploring new composition strategies, and integrating more advanced Tiny AI accelerators.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who supported and contributed to the successful completion of this research work titled "Collaborative Inference via Dynamic Composition of Tiny AI Accelerators on MCUs."

First and foremost, I extend my deepest thanks to my guide and mentor, Prof. Hiren Bhatt, for their invaluable guidance, insightful feedback, and continuous encouragement throughout the course of this study. Their expertise in embedded systems and AI acceleration significantly enriched the quality of this research.

I am also grateful to the faculty and staff of the B. Tech IT, Atmiya University, for providing the resources and environment conducive to academic exploration and innovation.

Special thanks to my peers and research collaborators for their constructive discussions, timely suggestions, and technical support. Their input greatly contributed to refining the architecture and performance analysis of the proposed dynamic composition framework.

Lastly, I would like to thank my family and friends for their unwavering moral support and patience throughout this research journey.

REFERENCES

- [1] Chen, X., et al. (n.d.). *Optimizing Edge AI Inference on Embedded Systems: A Comprehensive Review*. IEEE Transactions on Embedded Systems, 20(4), 2345–2360.
- [2] Yang, Y., et al. (n.d.). *Collaborative Inference for Distributed AI Models on IoT Devices*. International Journal of Computer Vision, 68(2), 185–201.
- [3] Sharma, R., et al. (n.d.). *Tiny AI for the Internet of Things: Hardware and Software Innovations*. IEEE Internet of Things Journal, 6(12), 9176–9186.
- [4] Google AI Blog. (n.d.). *Accelerating AI Inference on the Edge with Edge TPU*. Retrieved from <https://ai.googleblog.com>
- [5] Han, S., Mao, H., & Dally, W. J. (n.d.). *Deep compression: Compressing deep neural networks with pruning, trained quantization, and Huffman coding*. In Proceedings of the International Conference on Learning Representations (ICLR).
- [6] Laska, J. L., Murphy, M., & Marquardt, C. (n.d.). *TinyML: Machine learning at the edge with tiny devices*. In Proceedings of the ACM/IEEE International Conference on Computer-Aided Design (ICCAD).
- [7] Lane, N. D., et al. (n.d.). *DeepX: A resource-efficient deep learning framework for mobile and embedded devices*. In Proceedings of the International Conference on Pervasive Computing (Pervasive).
- [8] Raj, P., et al. (n.d.). *Federated learning on edge devices: A comprehensive review and future directions*. IEEE Access, 8, 228532–228551.

- [9] Yang, H., et al. (n.d.). *Collaborative deep learning on resource-constrained edge devices*. IEEE Transactions on Industrial Informatics, 16(6), 4383–4392.
- [10] Feng, Z., & Tang, T. (n.d.). *EdgeAI: A survey on AI inference at the edge*. Journal of Computer Science and Technology, 35(6), 1161–1176.
- [11] Zhang, Z., et al. (n.d.). *Efficient deployment of deep neural networks on IoT devices for real-time applications*. IEEE Internet of Things Journal, 8(8), 6543–6556.
- [12] Zhao, Y., et al. (n.d.). *Collaborative edge computing for resource-constrained IoT devices in smart cities*. IEEE Transactions on Industrial Informatics, 17(9), 6195–6204.
- [13] Zhang, S., et al. (n.d.). *TinyML: Machine learning on embedded systems*. IEEE Internet of Things Journal, 7(12), 10694–10702.
- [14] Wu, Z., et al. (n.d.). *Edge deep learning: A survey*. IEEE Access, 7, 137295–137309.
- [15] Pannuto, P., et al. (n.d.). *Accelerating embedded deep learning on mobile devices: A survey and future directions*. Journal of Low Power Electronics and Applications, 11(1), 6.
- [16] Nair, M. S., et al. (n.d.). *Federated learning with edge devices: The potential of collaborative inference in AI*. IEEE Transactions on Neural Networks and Learning Systems, 32(8), 3354–3368.
- [17] Chen, Y., et al. (n.d.). *Efficient deployment of AI models on embedded systems with hardware accelerators*. IEEE Transactions on Embedded Computing, 16(4), 617–629.
- [18] Zhou, Z., et al. (n.d.). *Energy-efficient deep learning for edge devices: Methods and applications*. IEEE Internet of Things Journal, 8(6), 4575–4586.
- [19] Kumar, M., et al. (n.d.). *Collaborative edge intelligence for Internet of Things: A survey*. Journal of Cloud Computing: Advances, Systems, and Applications, 9(1), 1–24.