Real-Time Systems (2025) 61:259–267
https://doi.org/10.1007/s11241-025-09445-9

Timing guarantees for inference of AI models in embedded systems

Seunghoon Lee[1] · Woosung Kang[2] · Marko Bertogna[3] · Hoon Sung Chwa[2] · Jinkyu Lee[1]

Accepted: 9 May 2025 / Published online: 18 June 2025

Abstract

Machine learning (ML) is increasingly being integrated into real-time embedded systems, enabling intelligent decision-making in applications such as autonomous driving and industrial automation. However, ensuring predictable execution of deep neural network (DNN) inference remains a major challenge, as real-time systems must meet strict timing constraints to guarantee safety and reliability. This paper identifies key challenges in achieving real-time AI inference in embedded systems, including limited memory capacity, high energy consumption, efficient multi-DNN scheduling, and heterogeneous resource management. To address these challenges, we emphasize the need for advanced scheduling algorithms to efficiently allocate heterogeneous computing resources across multiple DNNs, hierarchical memory management to reduce memory bottlenecks, and real-time neural architecture search and optimization techniques to enhance AI model performance under strict timing constraints. Furthermore, we discuss future research directions aimed at improving real-time AI execution, including time-predictable scheduling frameworks to ensure consistent inference latency, cross-device AI workload management to optimize resource utilization across heterogeneous processors, and benchmarking methodologies to systematically evaluate performance, timing guarantees, and energy efficiency in real-time AI systems. Advancing these research areas will enhance the reliability, efficiency, and scalability of AI-driven embedded systems, bridging the gap between ML advancements and real-time system requirements.

Keywords Timing guarantees · Embedded systems · Machine learning · Inference

Extended author information available on the last page of the article

1 Introduction

The rapid advancement of machine learning (ML) in embedded systems has revolutionized industries ranging from autonomous driving to industrial automation. However, while ML-driven decision-making offers enhanced functionality and intelligence, ensuring predictable execution times remains a formidable challenge. Deep neural networks (DNNs) require significant computational resources, often exceeding the constraints of embedded platforms with limited processing power and memory.

In real-time systems, ensuring deterministic behavior is not optional; instead, it is a fundamental requirement. Timing unpredictability in ML inference can lead to failures in safety-critical applications, such as autonomous vehicles missing hazard detection deadlines or robotic control systems failing to respond promptly to environmental changes. The challenge extends beyond raw computational power; optimizing execution order, managing memory efficiently, and coordinating heterogeneous processing units are critical to achieving real-time inference.

Recently, real-time systems research has been actively integrated into machine learning, with studies such as Kang et al. (2022a), Bateni and Liu (2018), Zhou et al. (2018), Yang et al. (2019), Xiang and Kim (2019), and Kang et al. (2021) proposing real-time scheduling frameworks for DNNs across various environments, aiming to meet stringent timing constraints and ensure reliable execution guarantees. Additionally, studies like Kang et al. (2022b) and Kang et al. (2024a) have demonstrated that employing real-time scheduling for specific tasks, such as object detection, can enhance performance while significantly improving safety. Beyond scheduling, effective memory management is essential for real-time DNN inference. To ensure predictable execution, real-time scheduling frameworks should effectively mitigate GPU memory bottlenecks. In this context, studies such as Kang et al. (2024b) and Ji et al. (2022) have proposed memory management techniques that optimize data movement and allocation, reducing memory contention while ensuring timely execution.

This paper promotes the creation of innovative real-time scheduling frameworks that ensure execution deadlines for ML inference in embedded systems. This step is crucial for maintaining the intelligence and predictability of AI-driven embedded systems. We also discuss recent research initiatives that have addressed these challenges and propose future research directions necessary to reconcile AI with stringent real-time requirements.

2 Challenges in real-time AI model inference

2.1 Memory constraints in AI execution

Novel scheduling and memory management techniques must be developed to enable AI models to run efficiently on embedded systems with extremely limited

internal memory. The use of small embedded devices with diverse processor units, such as TPUs and NPUs, has increased rapidly. However, there has been insufficient research addressing the utilization of these processor units, particularly in ensuring real-time guarantees. Currently, DNNs face compatibility and efficiency challenges when executed on processor units other than GPUs. Developing scheduling strategies for DNNs on these alternative units can significantly enhance embedded system performance. In addition, new compression techniques that are targeted at specific processor units, external memory optimization, and advanced memory-swapping mechanisms should be further explored.

2.2 Energy-efficient real-time AI execution

Many embedded systems, such as drones, UAVs, and battery-powered IoT devices, operate under stringent power constraints, necessitating efficient energy management strategies. While dynamic voltage and frequency scaling (DVFS) has been widely studied for general power optimization, its integration with task-specific scheduling to minimize energy consumption while ensuring real-time guarantees for ML remains largely unexplored since DVFS assumes a relatively predictable environment, whereas AI models are very data-dependent. Furthermore, hardware-software co-design approaches that leverage architectural optimizations, adaptive scheduling, and workload-aware power management strategies are essential for maximizing energy efficiency without compromising critical timing constraints.

2.3 Multi-DNN real-time scheduling

Emerging machine learning applications increasingly require the concurrent execution of multiple DNN models on a single embedded system, each with varying priorities and deadlines. Inefficient scheduling can lead to processor failures, degraded performance, and missed real-time constraints. To address this, advanced scheduling techniques must be developed to effectively manage resource allocation while leveraging intermediate layer outputs to maximize computational efficiency and minimize redundancy. Moreover, further advancements in scheduling algorithms are necessary to efficiently handle inter-model dependencies and optimize execution order, thereby preventing resource contention and latency bottlenecks.

2.4 Heterogeneous computing optimization

Heterogeneous computing has become essential for embedded AI workloads, which increasingly rely on diverse processing units such as GPUs, TPUs, NPUs, and custom accelerators to balance performance, power efficiency, and real-time constraints. However, research on effectively leveraging these accelerators remains in its early stages, requiring significant advancements in workload scheduling and resource management. One of the key challenges is the limited on-chip memory available in many of these accelerators, making it infeasible to store entire DNN models. To overcome this, models must be segmented into smaller portions that fit

within the available memory, requiring efficient memory allocation strategies and dynamic execution order management. While some segmentation techniques have been explored, they often fail to account for real-time constraints, necessitating further refinement.

Addressing these challenges is crucial for ensuring real-time guarantees in AI model inference on resource-constrained embedded systems. By advancing research in memory optimization, energy-efficient execution, multi-DNN scheduling, and heterogeneous computing, the real-time AI community can pave the way for more predictable, efficient, and scalable AI deployments in critical applications.

3 Future research directions

Implementing real-time AI inference in embedded systems presents multiple challenges related to execution determinism, memory efficiency, scheduling, and heterogeneous computing. In this section, while addressing the challenges above in detail, we illustrate our future research areas and highlight the methods that focus on time-predictable execution, hierarchical memory management, neural architecture search (NAS) with real-time constraints, cross-device AI scheduling, and the development of benchmarking frameworks.

3.1 Time-predictable system software

Ensuring time-predictable execution in embedded systems requires the development of real-time scheduling algorithms that minimize execution variability [e.g., a few millisecond inference time variance (Kang et al. 2022b)]. Current AI models lack deterministic execution behavior, leading to unpredictable inference latencies. We propose the design of deadline-aware scheduling algorithms for multi-DNN workloads, allowing concurrent execution without violating real-time constraints. Adaptive scheduling mechanisms that dynamically adjust AI model configurations based on real-time constraints are necessary to optimize inference latency. To further refine execution predictability, a feedback-driven scheduling framework will be implemented to iteratively refine AI model selection and execution order based on empirical performance data.

3.2 Hierarchical memory optimization for real-time AI

Hierarchical memory optimization is another critical aspect of real-time AI inference. Many embedded systems utilize multi-tiered memory architectures, which include on-chip SRAM, DRAM, and external storage such as SSD or zRAM. Traditional AI inference pipelines suffer from excessive memory access overhead when DNN models exceed available memory. To address this, we aim to develop real-time memory swapping mechanisms that allow AI inference on resource-limited devices while minimizing data transfer overhead. Furthermore, DNN partitioning strategies will be implemented to divide large models into smaller segments that fit within

constrained memory spaces. Efficient scheduling of data movement will be incorporated to ensure that inference execution overlaps with memory transfers, thereby reducing latency.

3.3 Neural architecture search with real-time constraints

Optimizing neural architecture search (NAS) for real-time constraints is also a key research challenge. Existing NAS frameworks focus on optimizing accuracy and model size but fail to consider execution-time constraints. We plan to extend the NAS methodologies to optimize multiple DNN architectures simultaneously while incorporating computational deadlines and energy constraints into the model selection process. The development of constraint-aware search algorithms will ensure that selected models comply with execution deadlines before deployment. A probabilistic real-time guarantee mechanism will also be introduced to predict whether an architecture will meet its timing constraints.

3.4 Cross-device AI scheduling for heterogeneous computing

As embedded systems increasingly incorporate heterogeneous computing architectures, effective scheduling of AI workloads across different processors is essential. The diversity of processing units, including MCUs, GPUs, TPUs, NPUs, and FPGAs, introduces challenges in workload distribution and resource allocation. Developing workload-aware task scheduling algorithms that dynamically assign AI inference tasks to the most appropriate processing unit based on real-time constraints and hardware capabilities is needed. In addition, latency-sensitive resource management strategies will be introduced to optimize data movement across these devices and improve inference efficiency. To enhance execution scalability, a cross-device AI execution pipeline will be designed to dynamically adapt to variations in processing power and available memory.

3.5 Real-time AI benchmarking and testbed development

The development of real-time AI benchmarking and testbeds is necessary to validate the proposed scheduling and optimization strategies. The absence of standardized benchmarking frameworks makes it difficult to compare different real-time AI inference techniques. To address this, we plan on establishing a benchmarking framework that models realistic workloads derived from autonomous systems, robotics, and edge computing applications. This will include the development of end-to-end evaluation pipelines that assess inference latency, execution predictability, and energy efficiency under real-world conditions.

Continuing in the research directions discussed above, our research team aims to bridge the gap between the design of the AI model and real-time execution. The proposed approaches are expected to enable the next generation of high-performance, low-latency embedded AI systems, ensuring reliable and deterministic inference under stringent computational constraints.

4 Conclusion

The intersection of real-time systems and ML presents both challenges and opportunities for embedded computing. Our future research focuses on addressing memory limitations, optimizing execution predictability, and developing novel scheduling methodologies to ensure AI models can operate reliably within real-time constraints. By implementing these strategies, we believe that it will improve the feasibility of deploying intelligent applications on resource-constrained embedded platforms.

Acknowledgements This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea government (MSIT) (RS-2024-00438248, RS-2024-00398157, RS-2023-00213309).

Author contributions ALL: contributing to the project proposal, which is associated with this paper. Seunghoon Lee and Jinkyu Lee: mainly writing and supervising the first draft. Woosung Kang, Marko Bertogna, and Hoon Sung Chwa: participating in writing the first draft, and refining/revising/proofreading the draft.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

Bateni S, Liu C (2018) Apnet: approximation-aware real-time neural network. In: 2018 IEEE real-time systems symposium (RTSS), IEEE, pp 67–79

Ji M, Yi S, Koo C, Ahn S, Seo D, Dutt N, Kim, J-C (2022) Demand layering for real-time dnn inference with minimized memory usage. In: 2022 IEEE 43rd real-time systems symposium (RTSS), IEEE, pp 291–304

Kang W, Lee K, Lee J, Shin I, Chwa HS (2021) LaLaRAND: flexible layer-by-layer CPU/GPU scheduling for real-time DNN tasks. In: 2021 IEEE real-time systems symposium (RTSS), IEEE, pp 329–341

Kang W, Chung S, Kim JY, Lee Y, Lee K, Lee J, Shin KG, Chwa HS (2022a) DNN-SAM: split-and-merge dnn execution for real-time object detection. In: 2022 IEEE 28th real-time and embedded technology and applications symposium (RTAS), IEEE, pp 160–172

Kang D, Lee S, Chwa HS, Bae S-H, Kang CM, Lee J, Baek H (2022b) RT-MOT: confidence-aware real-time scheduling framework for multi-object tracking tasks. In: 2022 IEEE real-time systems symposium (RTSS), IEEE, pp 318–330

Kang D, Lee S, Hong C-H, Lee J, Baek H (2024a) Batch-MOT: batch-enabled real-time scheduling for multi-object tracking tasks. IEEE Trans Comput-Aided Des Integr Circuits Syst. https://doi.org/10.1109/TCAD.2024.3443002

Kang W, Lee J, Lee Y, Oh S, Lee K, Chwa HS (2024b) RT-swap: addressing gpu memory bottlenecks for real-time multi-dnn inference. In: 2024 IEEE 30th real-time and embedded technology and applications symposium (RTAS), IEEE, pp 373–385

Xiang Y, Kim H (2019) Pipelined data-parallel CPU/GPU scheduling for multi-DNN real-time inference. In: 2019 IEEE real-time systems symposium (RTSS), IEEE, pp 392–405

Yang M, Wang S, Bakita J, Vu T, Smith FD, Anderson JH, Frahm J-M (2019) Re-thinking CNN frameworks for time-sensitive autonomous-driving applications: addressing an industrial challenge. In: 2019 IEEE real-time and embedded technology and applications symposium (RTAS), IEEE, pp 305–317

Zhou H, Bateni S, Liu C (2018) S∧ DNN: supervised streaming and scheduling for GPU-accelerated real-time dnn workloads. In: 2018 IEEE real-time and embedded technology and applications symposium (RTAS), IEEE, pp 190–201

Seunghoon Lee is a Ph.D. candidate in the Department of Computer Science and Engineering at Sungkyunkwan University (SKKU), South Korea, and a member of the Real-Time Computing Laboratory (RTCL). His research focuses on real-time embedded systems, particularly in scheduling algorithms, system-level optimization, and machine learning applications for timing-critical tasks.

Woosung Kang is a Ph.D. candidate in Computer Science at Daegu Gyeongbuk Institute of Science and Technology (DGIST) in South Korea and a member of the Real-Time Computing Laboratory. His research focuses on real-time embedded AI systems, with an emphasis on DNN inference scheduling, system-level optimization, and GPU memory management for resource-constrained environments.

Marko Bertogna is Full Professor at UNIMORE and leader of the HiPeRT Lab. His main research interests are in High-Performance Real-Time systems for multi- and many-core devices, Autonomous Driving and Industrial Automation systems. In 2008, he received a PhD in Computer Sciences from the Scuola Superiore Sant'Anna of Pisa. He has authored more than 100 papers, receiving multiple Best Paper Awards in first level international conferences. He coordinated multiple EU and industrial projects. He is CEO and founder of the academic spinoff HiPeRT Srl.

Hoon Sung Chwa received the B.S., M.S., and Ph.D. degrees in Computer Science from KAIST, Daejeon, South Korea, in 2009, 2011, and 2016, respectively. He is currently an Associate Professor in the Department of Electrical Engineering and Computer Science at DGIST, Daegu, South Korea. He was a Research Fellow in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, until 2018. His research interests include system design and analysis with timing guarantees, and resource management in real-time embedded and cyber-physical systems. He received Best Paper Awards from the 33rd IEEE Real-Time Systems Symposium (RTSS) in 2012, the IEEE International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA) in 2014, and the 31st IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) in 2025.

Jinkyu Lee is a professor in the Department of Computer Science and Engineering at Sungkyunkwan University (SKKU), South Korea, where he joined in 2014. He received the BS, MS, and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2004, 2006, and 2011, respectively. He has been a research fellow/visiting scholar in the Department of Electrical Engineering and Computer Science, University of Michigan until 2014. His research interests include system design and analysis with timing guarantees, QoS support, and resource management in real-time embedded systems and cyber–physical systems. He won the best student paper award from the 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) in 2011 and the Best Paper Award from the 33rd IEEE Real-Time Systems Symposium (RTSS) in 2012.

Authors and Affiliations

Seunghoon Lee[1] · Woosung Kang[2] · Marko Bertogna[3] · Hoon Sung Chwa[2] · Jinkyu Lee[1]

✉ Jinkyu Lee
jinkyu.lee@skku.edu

Seunghoon Lee
seunghoon.l@skku.edu

Woosung Kang
woosungkang@dgist.ac.kr

Marko Bertogna
marko.bertogna@unimore.it

Hoon Sung Chwa
chwahs@dgist.ac.kr

[1] Sungkyunkwan University, Suwon, Republic of Korea

[2] Daegu Gyeongbuk Institute of Science & Technology (DGIST), Daegu, Republic of Korea

[3] Universita di Modena e Reggio Emilia, Modena, Italy