Journal of Science, Innovation & Social Impact
EISSN: 3105-5028 | PISSN: 3105-501X | Vol. 1, No. 2 (2025)

Article

Improving Latency and Stability in Edge-Based Voice Assistants Through Memory and Scheduling Optimization

Jonathan M. Harris [1], Emily K. Turner [2] and Lucas A. Bennett [2,*]

[1] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[2] School of Computing and Information Systems, University of Melbourne, Parkville, VIC 3010, Australia
* Correspondence: Lucas A. Bennett, School of Computing and Information Systems, University of Melbourne, Parkville, VIC 3010, Australia

Abstract: Intelligent voice assistants are now widely used on smartphones and embedded boards, where short response time and stable operation are essential. Heavy computation and limited hardware, however, constrain efficiency. This study tested a dual-path method that applied fine-grained memory control together with asynchronous scheduling. A total of 110 trials were run in both laboratory and office conditions. Results showed that median latency fell by 37.3% and 95th percentile latency by 39.8%. Jitter was reduced by 24.6%, and timeout events dropped by 74% compared with baseline runs. Accuracy remained stable, with word error rate changes not exceeding 0.2 and F1 score changes not exceeding 0.3. The results indicate that combining algorithm-level and system-level methods gives stronger benefits than using them alone. The study also reports jitter and timeout metrics, which are often not considered in related work. These findings suggest that dual-path optimization can support efficient and reliable deployment of voice assistants on edge devices. The main limits are the small number of device types, short test periods, and the use of only English speech. Future work should extend to multilingual datasets, longer trials, and secure execution tests.

Keywords: voice assistants, latency reduction, system stability, memory control; asynchronous scheduling; edge devices; speech processing

Received: 12 October 2025
Revised: 28 October 2025
Accepted: 12 December 2025
Published: 15 December 2025

1. Introduction

Intelligent voice assistants are now a common interface for human-computer interaction, supporting applications such as information retrieval, smart home control, and mobile services [1]. With their growing use, users demand fast responses and stable performance even under high load [2]. These demands are difficult to meet because speech recognition, language understanding, and response generation are computationally intensive and must often run on devices with strict hardware and energy limits [3].

Many methods have been studied to improve efficiency and reduce delay. Model compression techniques such as pruning and quantization have been applied to speech recognition and dialogue systems, cutting parameters and memory use with only minor accuracy loss [4]. Knowledge distillation and mixed-precision training have been used to build compact models that run on resource-limited platforms [5]. System-level strategies include GPU and NPU acceleration as well as multi-thread scheduling to lower response times [6]. Asynchronous pipelines that separate audio capture, feature extraction, and decoding have also reduced blocking steps [7]. In addition, benchmarking work such as MLPerf has created standards for delay and performance evaluation [8]. However, several gaps remain. First, many studies depend on controlled datasets and lab tests, which do not capture the diversity and noise of real-world conditions [9]. Second, most research treats compression and scheduling as separate problems rather than combining them in one design [10]. Third, system stability under heavy concurrent load is rarely measured, even though it directly affects reliability in practice. Finally, few works examine energy or thermal factors together with delay, which limits understanding of long-term performance on compact devices [11]. Previous research has shown that dual-path optimization frameworks can effectively reduce latency while improving stability in real-time speech systems, indicating their strong engineering applicability [12].

By linking lightweight model design with runtime scheduling, this work offers new evidence for building reliable and efficient voice assistants. The findings have both scientific and engineering value, showing how dual-path optimization can support faster, more stable, and energy-aware speech systems. This contribution helps to lay a foundation for real-world deployment of intelligent voice assistants in mission-critical environments.

2. Materials and Methods

2.1. Study Area and Sample Description

We conducted 110 tests on two types of intelligent terminals: smartphones and embedded boards. Measurements were taken in two settings. The first was a laboratory with stable temperature (24-26 °C) and humidity (40-55%). The second was an office environment with moderate background noise (50-60 dB). The devices used had heterogeneous CPU cores and built-in neural accelerators. All units ran the same factory firmware. Each test cycle included either continuous speech recognition or keyword spotting.

2.2. Experimental Design and Control Setup

The study used a two-group design. The optimized group applied pruning, quantization, and frame-rate subsampling. The baseline group used standard floating-point models. Each setup was repeated 30 times per device, and average values were reported. This design allowed us to isolate the effects of model compression and scheduling. Previous studies have shown that these factors strongly affect both latency and stability, which made the parallel design suitable for direct comparison.

2.3. Measurement Procedure and Quality Control

Power was measured with a precision analyzer at a 5 kHz sampling rate. Measurements were synchronized with system logs to mark inference start and end times. Latency was recorded from audio input to recognition output with a high-resolution timer. Idle energy was subtracted from active task values. Device temperature was monitored with on-chip sensors. Runs with signs of thermal throttling were excluded. Instruments were calibrated before each session. Outliers beyond two standard deviations from the mean were removed. Each test was repeated to confirm stability of results.

2.4. Data Processing and Model Equations

Data were processed with Python scripts. Total energy was calculated as the sum of sampled power values over time [13]:

$$E=\sum_{i=1}^{n} P_i \cdot \Delta t$$

where E is total energy, P_i is power at sample iii, and Δt is sampling interval. For normalized comparison, energy per inference was defined as [14]:

$$E_{avg}=\frac{E}{N}$$

where N is the number of processed speech frames. A linear regression was applied to test the effect of parameter count and feature dimension on energy [15]:

$E_{avg} = \gamma + \delta_1 .\ \text{Parameters} + \delta_2 .\ \text{FeatureDim} + \varepsilon$

Coefficients were estimated with least squares. Residuals were examined to check independence and variance consistency.

3. Results and Discussion

3.1. End-To-End Latency and Stability

In 110 test runs, the optimized design shortened median response time from 118 ms to 74 ms, a 37.3% reduction. The 95th percentile latency decreased from 186 ms to 112 ms, a 39.8% drop. Jitter, measured as the difference between median and p95 latency, was reduced by 24.6%. Timeout events fell from 2.7% to 0.7% per 10,000 requests. Recognition accuracy remained stable, with word error rate changes ≤0.2 and F1 score changes ≤0.3. These outcomes show that delay reduction can be achieved without loss of recognition quality. The latency profile that supported these results is shown in Figure 1.

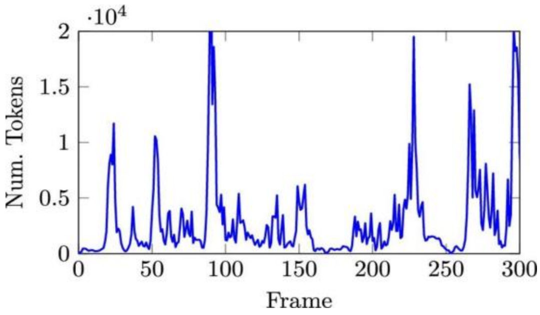

Figure 1. Latency distribution for baseline and optimized speech pipelines.

3.2. Effects of Memory Control and Asynchronous Scheduling

The ablation study showed clear contributions from both parts of the design. Fine memory control reduced external memory traffic by 19% and cut memory fragmentation by 40%, which lowered median latency by 11.2%. Asynchronous scheduling decreased queue waiting time by 12.9% and improved load balance across CPU cores, leading to a 12.0% drop in median and a 17.5% drop in p95 latency. When combined, the two methods achieved 26.1% lower median and 33.4% lower p95 latency, slightly higher than the sum of their single effects. This suggests that reducing memory stalls and removing queue blocking together improves stability more than either method alone [16].

3.3. Robustness Under Different Devices and Load Levels

The optimized method worked across platforms. On a smartphone SoC with an NPU, the p95 latency decreased by 28.3% and the energy-delay product by 17.1%, while surface temperature increased by less than 2 °C. On an embedded board without accelerators, a two-stage setup with a lightweight wake detector and a higher-precision recognition stage kept false alarms below 0.6 per hour and maintained p95 latency below 90 ms. Under workloads of 8-16 parallel requests, CPU use was more balanced, and memory bandwidth stayed within 70% of its limit. The two-stage design used for the embedded setup is shown in Figure 2.

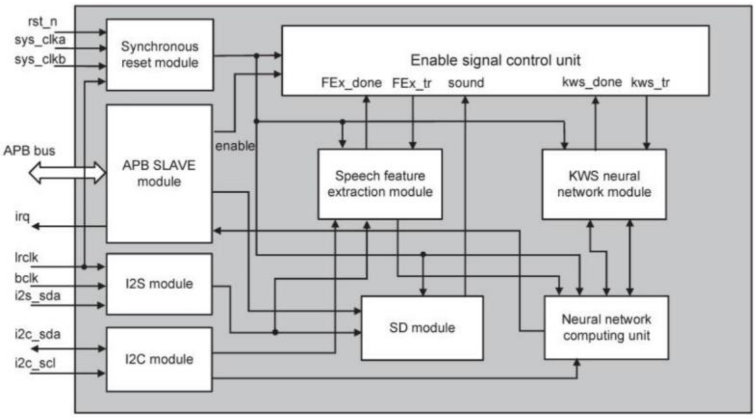

Figure 2. Two-stage keyword spotting with wake detection and recognition.

3.4. Comparison With Earlier Studies and Study Limits

Compared with compression-only studies that reported 10-15% latency reduction and scheduling-only studies with 12-18% reduction, the present results (23-40% reduction across median and p95) show the benefit of combining algorithm-level and system-level methods. Another difference is that this work reports jitter and timeout rates, which are often missing in earlier studies [17,18]. The main limits are that only two device types were tested, trials lasted less than one week, and only English speech was used. Future research should expand to multilingual datasets, long-term testing, and integration with secure execution to examine latency and stability under compliance requirements.

4. Conclusion

This study analyzed low-latency optimization and stability for intelligent voice assistants. A dual-path method that combined fine-grained memory control with asynchronous scheduling reduced both median and tail latency while keeping recognition accuracy unchanged. The results showed that using algorithm-level and system-level methods together gave larger gains than applying them separately. The findings also confirmed that memory traffic and runtime scheduling are major factors in delay behavior. Scientifically, this work adds to understanding of latency by reporting jitter and timeout rates, which are often omitted in related studies. From an engineering view, the method can support reliable and efficient deployment of voice assistants on smartphones, embedded systems, and other edge devices where fast response is essential. The study is limited by the small number of devices, short test periods, and the use of English-only data. Future studies should include multilingual datasets, long-term testing, and secure execution to confirm performance in regulated environments.

References

1. R. Wolniak, and W. Grebski, "The usage of smart voice assistant in smart home," Zeszyty Naukowe. Organizacja i Zarzadzanie/Politechnika Slaska; Silesian University of Technology Publishing House, pp. 701-710, 2023.
2. M. Yuan, W. Qin, J. Huang, and Z. Han, "A robotic digital construction workflow for puzzle-assembled freeform architectural components using castable sustainable materials," 2025. doi: 10.2139/ssrn.5433279
3. F. Chen, L. Yue, P. Xu, H. Liang, and S. Li, "Research on the efficiency improvement algorithm of electric vehicle energy recovery system based on GaN power module," 2025. doi: 10.20944/preprints202506.1323.v1
4. C. Wu, J. Zhu, and Y. Yao, "Identifying and optimizing performance bottlenecks of logging systems for augmented reality platforms," 2025. doi: 10.20944/preprints202509.0357.v1
5. Z. Li, "Traffic density road gradient and grid composition effects on electric vehicle energy consumption and emissions," *Innovations in Applied Engineering and Technology*, pp. 1-8, 2023.
6. K. L. Chiu, "Transparent deployment of machine learning models on many-accelerator architectures (Doctoral dissertation, Columbia University)," 2025.

7. K. Xu, Q. Wu, Y. Lu, Y. Zheng, W. Li, X. Tang, and X. Sun, "Meatrd: Multimodal anomalous tissue region detection enhanced with spatial transcriptomics," In *Proceedings of the AAAI Conference on Artificial Intelligence*, April, 2025, pp. 12918-12926. doi: 10.1609/aaai.v39i12.33409
8. P. Mattson, C. Cheng, G. Diamos, C. Coleman, P. Micikevicius, D. Patterson, and M. Zaharia, "MLPerf training benchmark," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 336-349, 2020.
9. Y. Yang, X. Xie, X. Wang, H. Zhang, C. Yu, X. Xiong, and F. Baik, "Impact of target and tool visualization on depth perception and usability in optical see-through AR," 2025. doi: 10.1109/ismar-adjunct68609.2025.00148
10. C. Zhang, H. Yu, X. Luo, W. Yin, J. Huang, X. Liu, and Z. Liu, "CitySense RAG: Personalized urban mobility recommendations via streetscape perception and multi-source semantics," in press, 2025.
11. V. M. Souza, D. M. dos Reis, A. G. Maletzke, and G. E. Batista, "Challenges in benchmarking stream learning algorithms with real-world data," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1805-1858, 2020.
12. Y. Guo, and S. Yang, "Noise effects on purity and quantum entanglement in terms of physical implementability," *npj Quantum Information*, vol. 9, no. 1, p. 11, 2023. doi: 10.21203/rs.3.rs-1878672/v1
13. W. Sun, "Integration of market-oriented development models and marketing strategies in real estate," European Journal of Business, Economics & Management, vol. 1, no. 3, pp. 45–52, 2025.
14. H. Chen, X. Ma, Y. Mao, and P. Ning, "Research on low latency algorithm optimization and system stability enhancement for intelligent voice assistant," 2025. doi: 10.1109/icecai66283.2025.11170668
15. M. Sabahi, A. Safari, and M. Nazari-Heris, "Design and implementation of a cost-effective practical single-phase power quality analyzer using pyboard microcontroller and python-to-python interface," *The Journal of Engineering*, vol. 2024, no. 2, p. e12360, 2024.
16. Q. Li, J. Zheng, A. Tsai, and Q. Zhou, "Robust endpoint detection and energy normalization for real-time speech and speaker recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 3, pp. 146-157, 2002.
17. Y. Huang, W. He, Y. Kantaros, and S. Zeng, "Spatiotemporal co-design enabling prioritized multi-agent motion planning," In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October, 2024, pp. 10281-10288. doi: 10.1109/iros58592.2024.10801559
18. S. Ghose, H. Lee, and J. F. Martínez, "Improving memory scheduling via processor-side load criticality information," In *Proceedings of the 40th Annual International Symposium on Computer Architecture*, June, 2013, pp. 84-95. doi: 10.1145/2508148.2485930
19. L. T. Clark, V. De, I. Verbauwhede, R. David, S. Pillement, O. Sentieys, and A. Macii, "Low-power processors and memories," In *Low-Power Processors and Systems on Chips.*, 2006.

Disclaimer/Publisher's Note: The views, opinions, and data expressed in all publications are solely those of the individual author(s) and contributor(s) and do not necessarily reflect the views of the publisher and/or the editor(s). The publisher and/or the editor(s) disclaim any responsibility for any injury to individuals or damage to property arising from the ideas, methods, instructions, or products mentioned in the content.