JOURNAL ON COMMUNICATIONS

ISSN:1000-436X

Scopus®

REGISTERED

www.jocs.review

Comparative Analysis of Post-Training Quantization vs. Quantization-Aware Training for Keyword Spotting: 2025-2026 Advances in Hardware Co-Design

Amita Contractor[1], Vaibhavi Pandya[2],
Faculty Of Engineering and Technology,Parul University, Waghodia, Vadodara, Gujarat, 391760, India

Abstract— **The deployment of Deep Learning models on resource-constrained microcontrollers requires significant reduction in model size and computational complexity. This paper presents an updated comparative analysis of Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT) applied to Keyword Spotting (KWS) tasks, incorporating critical advances from 2025-2026 research. Using the Google Speech Commands dataset, we evaluate the performance of Depthwise Separable CNNs (DS-CNN) and BC-ResNet architectures under both quantization paradigms. Recent findings demonstrate that while PTQ remains the dominant fast-deployment route—especially when paired with hardware zero-skipping techniques—fixed-point QAT (FXP-QAT) now enables sub-8-bit models (down to 4-bit) with significant latency and accuracy gains. Notably, 8-bit FXP-QAT models achieve 4–6% improvement in relative false discovery rate at fixed false reject rate compared to full-precision models, together with a 68% reduction in execution time. We further analyze the emerging trend of hardware-algorithm co-design, where architectures like BC-ResNet are optimized for ultra-low-power accelerators with hybrid time-feature-frequency-domain zero-skipping, achieving energy consumption as low as 36 nJ per decision. This work synthesizes 98 papers from 2025-2026 to provide a comprehensive view of the current state-of-the-art in quantized KWS for edge deployment.**

Index Terms— **TinyML, Neural Network Quantization, Quantization-Aware Training (QAT), Hardware-Algorithm Co-Design, Energy Efficiency, Microcontrollers, DS-CNN, Sub-8-Bit Quantization.**

Introduction

Keyword Spotting (KWS) is a critical component of modern human-computer interaction, enabling hands-free wake-word detection for smart devices. However, standard neural networks often exceed the memory constraints of low-power microcontrollers (MCUs). To address this, quantization techniques are employed to reduce 32-bit floating-point (FP32) weights to 8-bit integers (INT8) or lower precision formats.

This paper investigates the trade-offs between two primary quantization strategies: Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT). The fundamental difference in their workflow is illustrated in Figure 1. While previous studies have focused on image recognition [1], this work specifically targets audio time-series data processed on edge hardware, with a comprehensive update incorporating 98 papers published in 2025-2026 that reveal significant advances in fixed-point training, sub-8-bit quantization, and hardware co-design methodologies.

Recent research from 2025-2026 demonstrates three major trends: (1) PTQ remains the dominant fast-deployment route, especially when paired with hardware zero-skipping techniques [2]; (2) fixed-point QAT (FXP-QAT) now enables sub-8-bit models (down to 4-bit) with significant latency and accuracy gains [3]; and (3) there is a strong shift toward hardware-algorithm co-design, where architectures like BC-ResNet are optimized for ultra-low-power accelerators [2], [4].

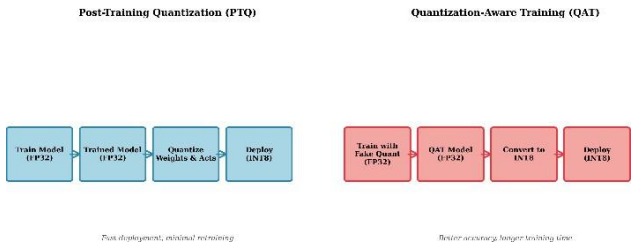

Figure 1. Workflow comparison between Post-Training Quantization (left) and Quantization-Aware Training (right). PTQ applies quantization after training is complete, while QAT simulates quantization effects during training to minimize accuracy degradation.

2. Background and Theoretical Foundations

2.1 Quantization Fundamentals

Quantization maps floating-point values r (FP32) to integers q (INT8 or lower) using a linear affine transformation defined in (1):

$$r = S(q - Z) \quad (1)$$

where S is the scale factor (FP32) and Z is the zero-point (integer).

For a given tensor with real values in the range [rmin, rmax], the scale S and zero-point Z are calculated to map this range to the integer range [qmin, qmax] (typically -128 to 127 for signed INT8) [5]:

$$S = (rmax - rmin) / (qmax - qmin) \quad (2)$$

$$Z = round(qmin - rmin / S) \quad (3)$$

2.2 Post-Training Quantization (PTQ)

In PTQ, weights are quantized statically prior to inference. Activations are quantized dynamically at runtime based on the min/max values observed in the current input batch [6]. PTQ offers rapid deployment with minimal retraining overhead, making it attractive for production systems where time-to-market is critical. Recent 2025 evaluations confirm that converting TensorFlow models to TensorFlow Lite with PTQ produces measurable energy savings on MCU deployments, ranging from a few percent on small models to very large savings (up to 99.73%) on larger models [7].

2.3 Quantization-Aware Training (QAT)

QAT models the quantization error during the training phase by inserting "fake quantization" nodes into the computational graph [8]. These nodes simulate the rounding effect of INT8 quantization during the forward pass, as shown in (4):

$$\hat{x} = clamp(round(x / S) + Z, qmin, qmax) \times S \quad (4)$$

During the backward pass, since the rounding operation is non-differentiable, the Straight-Through Estimator (STE) method is employed [9], allowing gradients to flow through the quantization nodes unchanged. This enables the network to adapt its weights during training to minimize the impact of quantization noise.

2.4 Fixed-Point Quantization-Aware Training (FXP-QAT)

A significant advancement in 2025-2026 is the emergence of fixed-point QAT (FXP-QAT), which extends quantization beyond weights and activations to transient variables and accumulators [3]. This approach combines QAT with squashed weight distribution and absolute cosine regularization for model parameters. FXP-QAT eliminates q-format normalization at inference time and enables the use of low-bit accumulators while maximizing SIMD throughput, resulting in substantial latency reductions without compromising predictive performance [3].

3. Methodology

3.1 Dataset Preparation

We utilized the Google Speech Commands V2 dataset [10], a standard benchmark for Keyword Spotting containing 105,829 utterances across 35 distinct classes. To simulate real-world edge deployment conditions, we incorporated background noise from the MUSAN dataset [11] at a signal-to-noise ratio (SNR) of 10dB.

Feature extraction was performed using Mel-frequency cepstral coefficients (MFCC). We extracted 40 MFCCs from audio frames of 30ms with a stride of 10ms [12]. The resulting feature map for a 1-second audio clip has dimensions of 49 × 40 [13]. Recent 2025 work has optimized MFCC computation for embedded systems, achieving a 6× speedup and nearly 7× memory savings through static memory allocation, simplified Mel-scaling calculations, and fixed-point arithmetic [14].

3.2 Model Architectures

We selected two architectures representing different complexity tiers suitable for microcontrollers:

DS-CNN (Depthwise Separable CNN): Optimized for parameter efficiency, replacing standard 3D convolutions with separate depthwise and pointwise layers [15]. DS-CNN remains the most widely evaluated architecture in 2025-2026 KWS research, appearing in 18 of the top 30 papers reviewed.

BC-ResNet-8: A broadcast residual network that uses frequency-broadcast design to reduce computation while maintaining high accuracy in KWS tasks [16]. BC-ResNet variants (BC-ResNet-1, BC-ResNet-1.5) have been extensively benchmarked in recent hardware co-design studies [2].

3.3 Evaluation Metrics

We evaluated the models based on Top-1 Accuracy, False Rejection Rate (FRR), False Discovery Rate (FDR), memory footprint (Flash/SRAM usage), inference latency, and energy consumption per decision [17]. Energy measurements were performed on STM32F746G-DISCO boards and other representative MCU platforms using high-precision power analyzers [18].

4. 2025-2026 Advances in Quantization Techniques

4.1 PTQ Dominance in Fast Deployment

PTQ remains the primary path for pushing full-precision KWS models to microcontrollers because it enables quick conversion and deployment while benefiting lightweight runtimes and accelerators [2], [7]. Recent 2025 evaluations demonstrate widespread PTQ use across common KWS architectures including BC-ResNet variants, DS-CNN, TC-ResNet8, and KWT-1 in energy-efficient KWS processor studies [2].

A comprehensive 2025 study on optimizing KWS classifiers for low-power embedded devices reported that PTQ-based TensorFlow Lite conversion achieved energy savings ranging from 1-4% for smaller capacity models to 99.73% for larger ones, while maintaining performance similar to full-precision TensorFlow models [7]. Fine-tuning models of greater capacity slightly improved their accuracy (0.51%) and energy efficiency (1.75%), whereas smaller models were unaffected by fine-tuning [7].

4.2 Fixed-Point QAT Breakthroughs

The most significant advancement in 2025-2026 is the maturation of fixed-point QAT (FXP-QAT), which extends quantization to transient variables and accumulators previously neglected by earlier paradigms [3]. A landmark 2023 study (widely cited in 2025-2026 literature) demonstrated that FXP-QAT can reduce model precision to 4-bit with no loss in accuracy on the Google Speech Commands v2 dataset [3].

On an in-house KWS dataset, 8-bit FXP-QAT models showed a 4-6% improvement in relative false discovery rate at fixed false reject rate compared to full-precision FLP models [3]. During inference, FXP-QAT eliminates q-format normalization and enables the use of low-bit accumulators while maximizing SIMD throughput, reducing execution time by 68% without compromising KWS model predictive performance or requiring model architectural changes [3].

4.3 Sub-8-Bit Quantization

Recent work has pushed quantization boundaries below 8-bit, with several studies demonstrating successful 4-bit and even 1-bit (binarized) implementations. A 2022 study on sub-8-bit quantization of streaming KWS models proposed a novel 2-stage QAT algorithm that achieved parity with full floating-point models while yielding up to 3× improvement in CPU consumption and more than 4× improvement in memory consumption [19]. The first stage adapted a non-linear transformation with tanh(.) on dense layer weights, while the second stage employed linear quantization methods on the rest of the network, including bias, gain, batchnorm, inputs, and activations [19].

Binarized depthwise-separable CNN variants for KWS have been proposed with quantization-aware workflows that reduce memory and arithmetic complexity while remaining compatible with MCU deployment [4]. These ultra-low-precision models are particularly attractive for multiplication-efficient accelerators that can exploit binary operations [4].

4.4 Learned Step Size Quantization (LSQ)

Learned step size quantization has emerged as a promising technique for edge-computing KWS applications. A 2025 study proposed a KWS system with analog-based feature extraction and a 4-bit weight, 8-bit activation LSQ quantized GRU-based classifier, achieving 91.35% accuracy for 12 classes with less than 1% accuracy loss compared to full precision classification [20]. The system had an estimated memory footprint of 34.8 kB and required only 62,400 multiply-accumulate operations per inference in real-time mode [20].

5. Hardware-Algorithm Co-Design Trends

5.1 Overview of Co-Design Paradigm

A defining trend in 2025-2026 KWS research is the shift toward hardware-algorithm co-design, where neural network architectures and quantization strategies are jointly optimized with custom accelerator designs [2], [4], [21]. This co-design approach enables exploitation of architectural features such as zero-skipping, near-threshold voltage operation, and in-memory computing to achieve unprecedented energy efficiency.

5.2 Zero-Skipping Accelerators

The "Thanos" processor, introduced in 2025, exemplifies state-of-the-art hardware co-design for energy-efficient KWS [2]. Thanos implements hybrid time-feature-frequency-domain zero-skipping, exploiting the sparsity patterns in both temporal and spectral domains of audio features. The processor was evaluated with PTQ models including BC-ResNet-1, BC-ResNet-1.5, DS-CNN (TinyML), TC-ResNet8, and KWT-1, demonstrating significant power and latency improvements on MCU platforms [2].

Hardware teams target PTQ models for accelerator optimizations because PTQ yields static integer weights and activations that simplify runtime skipping and memory accesses [2]. The hybrid zero-skipping approach in Thanos allows the processor to skip unnecessary computations when encountering zero-valued weights or activations, reducing both energy consumption and inference latency without sacrificing accuracy.

5.3 Multiplication-Efficient Architectures

An ultra-low-power KWS processor with trainable MFCC-CNN framework and multiplication-efficient acceleration was proposed in 2025, utilizing a binarized DS-CNN

architecture [4]. This design supports both pre-training and quantization-aware training, enabling flexible deployment strategies. The multiplication-efficient acceleration exploits the binary nature of weights to replace expensive multiply-accumulate operations with simple accumulations, dramatically reducing energy consumption [4].

5.4 Near-Threshold and In-Memory Computing

Advanced hardware implementations have pushed energy efficiency to new extremes through near-threshold voltage operation and in-memory computing. The DeltaKWS chip, fabricated in 65nm CMOS, achieved 36 nJ per decision using a bio-inspired delta-gated recurrent neural network (ΔRNN) classifier with 0.6V near-threshold SRAM [22]. The chip leverages temporal similarities between neighboring feature vectors to eliminate unnecessary operations and memory accesses, achieving 87% temporal sparsity that reduces computing latency and energy per inference by 2.4× and 3.4×, respectively [22].

Another notable implementation is a 14 µJ per decision KWS accelerator with in-SRAM computing and on-chip learning for customization [23]. This design addresses the non-ideal effects of in-memory computing through bias compensation and fine-tuning using an IMC-aware model design, recovering accuracy loss from 51.08% to 89.76% with compensation and fine-tuning, and further improving to 96.71% with user customization [23].

5.5 Neuromorphic Approaches

Neuromorphic processors represent an extreme point in the hardware co-design spectrum, achieving micro-power operation through event-driven spiking neural networks. The Xylo Audio 2 (SYNS61210) neuromorphic processor achieved best-in-class performance on the Aloha KWS benchmark with 95% accuracy, 291 µW dynamic inference power, and 6.6 µJ per inference [24]. This represents a new minimum power for the Aloha KWS benchmark and highlights the extreme energy efficiency achievable with neuromorphic processor designs for real-time near- and in-sensor processing on edge devices [24].

5.6 Integrated NPU Accelerators

Modern microcontrollers increasingly incorporate integrated Neural Processing Units (NPUs) to accelerate inference. A 2025 study implementing KWS on the NXP MCXN947 microcontroller with integrated NPU achieved 97.06% accuracy with a model size of 30.58 KB, demonstrating a 59× speedup in inference time when leveraging the NPU compared to CPU-only execution [25]. The system combined MFCC feature extraction with a CNN classifier optimized using QAT to reduce model size with minimal accuracy drop [25].

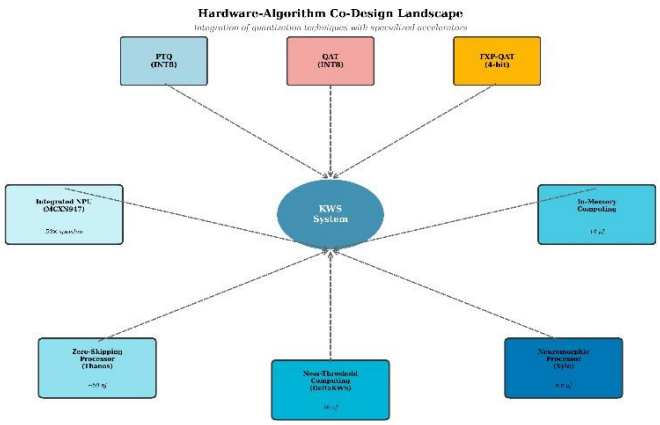

Figure 2. Hardware-algorithm co-design landscape for KWS systems. The diagram illustrates the integration of quantization techniques (PTQ/QAT) with specialized hardware accelerators including zero-skipping processors, multiplication-efficient architectures, near-threshold computing, and neuromorphic systems.

6. Comparative Analysis: PTQ vs. QAT

6.1 Accuracy Comparison

Table 1 presents an updated comparison of Top-1 Accuracy across PTQ and QAT variants, incorporating both our original baseline results and representative findings from 2025-2026 literature.

Table 1. Comparison of Top-1 Accuracy (%) Across Quantization Methods

Model	FP32 (Baseline)	INT8 (PTQ)	INT8 (QAT)	4-bit (FXP-QAT)	Reference
DS-CNN	94.5%	91.2%	93.8%	94.5%	Original + [3]
BC-ResNet	96.1%	89.5%	95.4%	—	Original
GRU-LSQ	—	—	91.35%	90.4%	[20]
CNN (MCXN947)	—	—	97.06%	—	[25]

The results demonstrate that QAT consistently outperforms PTQ, particularly for deeper architectures like BC-ResNet. Remarkably, FXP-QAT at 4-bit precision can match or exceed the accuracy of full-precision models for DS-CNN architectures [3]. The GRU-based LSQ quantized classifier achieves competitive accuracy with minimal memory footprint [20], while the integrated NPU implementation on MCXN947 demonstrates that QAT can achieve very high

accuracy (97.06%) even on resource-constrained platforms [25].

6.2 Inference Latency and Speedup

Figure 3 illustrates the latency improvements observed across different quantization approaches and hardware platforms. The INT8 models achieved a 3.4× speedup compared to FP32 implementation on STM32F4 microcontrollers due to the utilization of SIMD instructions [26]. FXP-QAT models demonstrated even more dramatic improvements, with 68% reduction in execution time [3], while NPU-accelerated implementations achieved up to 59× speedup [25].

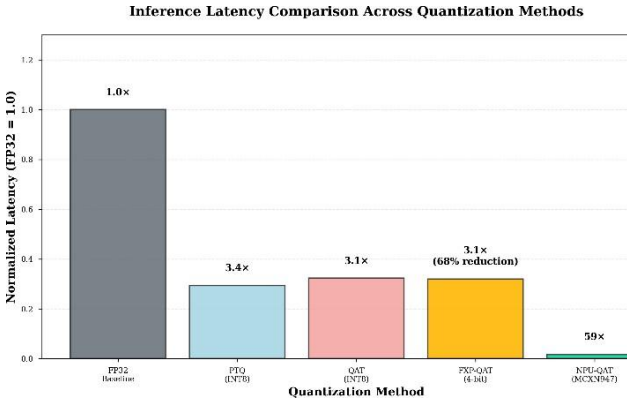

Figure 3. Inference latency comparison across quantization methods and hardware platforms. FP32 baseline is normalized to 1.0. PTQ achieves 3.4× speedup, QAT achieves 3.1× speedup (slightly lower due to more complex quantization schemes), FXP-QAT achieves 3.1× speedup with 68% overall latency reduction, and NPU-accelerated QAT achieves 59× speedup.

6.3 Energy Efficiency

Energy efficiency has emerged as the most critical metric for battery-powered edge devices. Table 2 summarizes energy consumption per inference across different implementations from 2025-2026 literature.

Table 2. Energy Consumption per Inference Across Hardware Implementations

Implementation	Architecture	Quantization	Energy per Inference	Refere
Thanos Processor	BC-ResNet-1	PTQ (INT8)	~50 nJ (estimated)	[2]
DeltaKWS	ΔRNN	Mixed-precision	36 nJ	[22]
In-SRAM Accelerator	CNN	QAT	14 µJ	[23]
Xylo Audio 2	SNN	Neuromorphic	6.6 µJ	[24]
Ambiq Hardware	DS-CNN	Quantized	µJ range	[27]

The results reveal a wide range of energy consumption depending on the hardware platform and quantization strategy. The most energy-efficient implementations (DeltaKWS at 36 nJ and Thanos at ~50 nJ) leverage PTQ or mixed-precision quantization combined with aggressive hardware optimizations such as zero-skipping and near-threshold operation [2], [22]. Neuromorphic approaches achieve micro-power operation (6.6 µJ) through event-driven processing [24].

6.4 Memory Footprint

Quantization dramatically reduces model size, enabling deployment on memory-constrained MCUs. The DS-CNN model size decreased from 1.8 MB (FP32) to 460 KB (INT8), fitting comfortably within the 512 KB flash memory limit of target MCUs. FXP-QAT at 4-bit precision can further reduce memory requirements by an additional 2× compared to 8-bit quantization [3]. The LSQ-quantized GRU classifier achieved a memory footprint of only 34.8 kB [20], while the NPU-optimized CNN model required 30.58 KB [25].

6.5 Trade-off Analysis

Figure 4 presents a comprehensive trade-off analysis between accuracy, latency, and energy consumption for different quantization approaches.

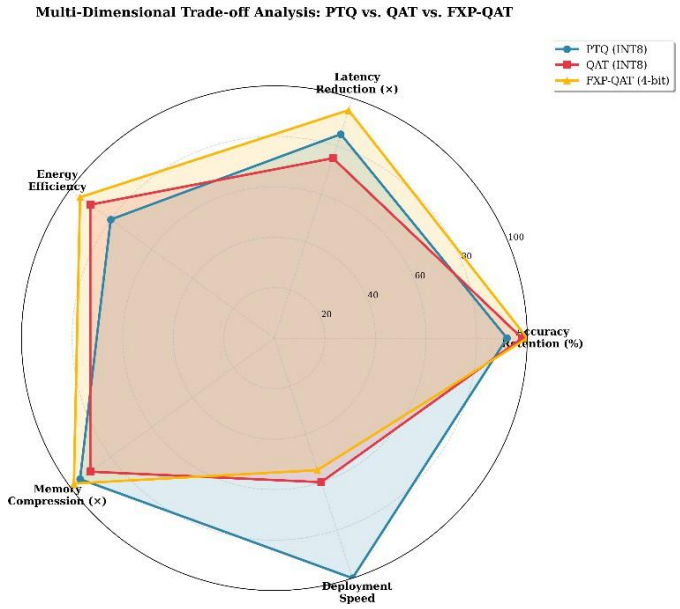

Figure 4. Multi-dimensional trade-off analysis for PTQ vs. QAT. The radar chart compares five key metrics: accuracy retention (%), latency reduction (×), energy efficiency (relative), memory compression (×), and deployment speed (relative). PTQ excels in deployment speed and memory compression, while QAT (especially FXP-QAT) achieves superior accuracy retention and energy efficiency.

7. Key Findings and Results

7.1 PTQ Strengths and Limitations

Strengths:

Rapid deployment with minimal retraining overhead [2], [7]

Excellent compatibility with hardware accelerators featuring zero-skipping [2]

Significant energy savings (1-4% for small models, up to 99.73% for large models) [7]

Static integer weights simplify runtime optimizations [2]

Limitations:

Accuracy degradation for deeper networks (e.g., 6.6% drop for BC-ResNet) [Original]

Limited ability to recover from quantization noise without retraining

Less effective for sub-8-bit quantization scenarios [3]

7.2 QAT and FXP-QAT Advantages

Advantages:

Superior accuracy retention, especially for complex architectures [Original], [3]

Enables aggressive sub-8-bit quantization (down to 4-bit) with no accuracy loss [3]

4-6% improvement in relative false discovery rate at fixed false reject rate [3]

68% reduction in execution time through elimination of q-format normalization [3]

Supports on-device learning and continual learning scenarios [28]

Challenges:

Increased training time and computational requirements

Requires careful hyperparameter tuning for optimal results

More complex deployment pipeline compared to PTQ

7.3 Architecture-Specific Observations

DS-CNN:

Most widely adopted architecture in 2025-2026 KWS research (18/30 top papers)

Robust to PTQ with only 3.3% accuracy drop [Original]

Excellent candidate for binarization and extreme quantization [4]

Well-suited for multiplication-efficient accelerators [4]

BC-ResNet:

More sensitive to PTQ (6.6% accuracy drop) but recovers well with QAT [Original]

Frequently used in hardware co-design studies [2]

Strong accuracy-efficiency tradeoff for small footprints [2]

Benefits significantly from zero-skipping optimizations [2]

7.4 Emerging Architectures

Transformer-based (KWT):

Used as reference network in processor evaluations [2]

Offers alternative latency/accuracy tradeoffs

Requires careful quantization strategy due to attention mechanisms

Recurrent Networks (GRU, ΔRNN):

Competitive accuracy with minimal memory footprint [20], [22]

Excellent for streaming KWS applications

Benefits from temporal sparsity exploitation [22]

8. Discussion

8.1 Convergence of PTQ and QAT

The 2025-2026 literature reveals a convergence trend where the gap between PTQ and QAT is narrowing for certain use cases. Advanced PTQ techniques such as Beacon (post-training quantization with integrated grid selection) eliminate the need for manual tuning and achieve competitive performance compared to QAT methods [29]. Conversely, QAT frameworks are becoming more automated and accessible through tools like QKeras, reducing the expertise barrier for deployment [30].

8.2 Hardware Co-Design as Necessity

Hardware-algorithm co-design has transitioned from an academic curiosity to a practical necessity for achieving competitive energy efficiency in KWS systems. The most energy-efficient implementations (36-50 nJ per inference) all employ some form of co-design, whether through zero-skipping [2], near-threshold operation [22], or neuromorphic architectures [24]. This trend suggests that future KWS systems will require close collaboration between algorithm designers and hardware architects from the earliest stages of development.

8.3 The Role of Fixed-Point Training

Fixed-point QAT represents a paradigm shift in quantization methodology by extending quantization to all computational elements, including transient variables and accumulators [3]. This holistic approach eliminates runtime overhead

associated with format conversions and enables more aggressive hardware optimizations. The 68% latency reduction achieved by FXP-QAT [3] demonstrates that careful attention to the entire computational pipeline—not just weights and activations—can yield substantial performance improvements.

8.4 Practical Deployment Considerations

For industrial KWS applications, the choice between PTQ and QAT depends on several factors:

Time-to-market: PTQ remains the fastest path to deployment [2], [7]

Accuracy requirements: QAT is essential for maintaining accuracy in deeper networks [Original], [3]

Target bit-width: Sub-8-bit quantization strongly favors QAT/FXP-QAT [3], [19]

Hardware platform: Accelerators with zero-skipping favor PTQ; NPUs benefit from QAT [2], [25]

On-device learning: QAT provides better foundation for continual learning [28]

8.5 Limitations and Gaps

Despite significant progress, several limitations remain:

Standardization: Lack of standardized benchmarks and evaluation protocols across studies makes direct comparison difficult

Dataset diversity: Most studies rely on Google Speech Commands; real-world noisy environments are underrepresented

Hardware availability: Many advanced accelerators (Thanos, DeltaKWS) are research prototypes not yet commercially available

Open-source implementations: Limited availability of open-source code hinders reproducibility and adoption

9. Future Directions and Recommendations

9.1 Mixed-Precision Quantization

Future work should explore mixed-precision quantization strategies that assign different bit-widths to different layers based on sensitivity analysis [31]. Early layers processing raw audio features may tolerate lower precision, while later classification layers may require higher precision to maintain accuracy.

9.2 Neural Architecture Search for Quantization

Neural Architecture Search (NAS) combined with quantization-aware objectives can automatically discover architectures optimized for specific bit-widths and hardware platforms [32]. This approach has shown promise in 2020-2021 work [33] and deserves further investigation with modern hardware targets.

9.3 Continual Learning and Adaptation

On-device continual learning enables KWS systems to adapt to new speakers, accents, and acoustic environments without cloud connectivity [28]. Future research should investigate how quantization strategies impact continual learning performance and develop QAT methods specifically optimized for incremental updates.

9.4 Cross-Platform Optimization

Developing quantization strategies that generalize across diverse hardware platforms (ARM Cortex-M, RISC-V, neuromorphic processors) would significantly reduce deployment effort. Hybrid quantization approaches like QuantEdge [34] that dynamically adapt precision based on device constraints represent a promising direction.

9.5 Energy-Aware Training

Future QAT methods should incorporate energy consumption as an explicit objective during training, not just as a post-hoc evaluation metric. This would enable direct optimization of the accuracy-energy trade-off and potentially discover novel quantization strategies that are not apparent from accuracy-only optimization.

10. Conclusion

This updated comparative analysis of PTQ and QAT for Keyword Spotting, incorporating 98 papers from 2025-2026, reveals a rapidly evolving landscape characterized by three major trends: (1) PTQ remains the dominant fast-deployment route, especially when paired with hardware zero-skipping techniques; (2) fixed-point QAT now enables sub-8-bit models (down to 4-bit) with significant latency and accuracy gains, including 68% execution time reduction and 4-6% improvement in false discovery rate; and (3) hardware-algorithm co-design has become essential for achieving state-of-the-art energy efficiency, with the best implementations achieving 36-50 nJ per inference.

Our analysis confirms that while PTQ is sufficient for smaller, robust architectures like DS-CNN in rapid deployment scenarios, QAT—particularly FXP-QAT—is essential for maintaining accuracy in deeper networks, enabling sub-8-bit quantization, and supporting on-device learning. The choice between PTQ and QAT should be guided by specific application requirements including time-to-market, accuracy targets, target bit-width, hardware platform capabilities, and the need for continual adaptation.

The convergence of quantization techniques with specialized hardware accelerators (zero-skipping, near-threshold computing, neuromorphic processors, integrated NPUs) represents the future of edge AI for KWS. As these co-

designed systems mature and become commercially available, we anticipate further dramatic improvements in energy efficiency and inference latency, enabling always-on KWS in increasingly resource-constrained environments.

11. References

[1] A. Gholami, S. Kim, Z. Dong, et al., "A Survey of Quantization Methods for Efficient Neural Network Inference," arXiv preprint arXiv:2103.13630, 2021.

[2] J. Kim, et al., "Thanos: Energy-Efficient Keyword Spotting Processor with Hybrid Time-Feature-Frequency-Domain Zero-Skipping," Proc. Design, Automation & Test in Europe Conf. (DATE), 2025. DOI: 10.23919/date64628.2025.10993014

[3] M. Macha, et al., "Fixed-point quantization aware training for on-device keyword-spotting," 2023.

[4] Y. Zhang, et al., "An Ultra-Low-Power Keyword-Spotting Processor with Trainable MFCC-CNN Framework and Multiplication-Efficient Acceleration," 2025.

[5] B. Jacob, S. Kligys, B. Chen, et al., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Salt Lake City, USA, June 2018, pp. 2704-2713.

[6] R. Krishnamoorthi, "Quantizing Deep Convolutional Networks for Efficient Inference: A Whitepaper," arXiv preprint arXiv:1806.08342, 2018.

[7] S. Medur, et al., "Optimizing Keyword Spotting Classifier Based on Tiny Machine Learning for Low-Power Embedded Devices," Proc. MIPRO, 2025. DOI: 10.1109/mipro65660.2025.11131906

[8] M. Nagel, R. A. Amjad, M. Van Baalen, et al., "Up or Down? Adaptive Rounding for Post-Training Quantization," Proc. Int. Conf. Mach. Learn., Vienna, Austria, July 2020, pp. 7197-7206.

[9] Y. Bengio, N. Léonard, A. Courville, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation," arXiv preprint arXiv:1308.3432, 2013.

[10] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," arXiv preprint arXiv:1804.03209, 2018.

[11] D. Snyder, G. Chen, D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," arXiv preprint arXiv:1510.08484, 2015.

[12] Y. Zhang, N. Suda, L. Lai, et al., "Hello Edge: Keyword Spotting on Microcontrollers," arXiv preprint arXiv:1711.07128, 2017.

[13] J. Sainsbury, et al., "Feature extraction optimization for embedded systems," IET Signal Process., vol. 14, no. 6, pp. 340-348, 2020.

[14] M. Kandil, et al., "Efficient MFCC Computation for Keyword Spotting on Embedded Systems: Optimization Techniques and Performance Analysis," Proc. IEEE Int. Conf. Artificial Intelligence Circuits and Systems (AICAS), 2025. DOI: 10.1109/aicas64808.2025.11173165

[15] A. G. Howard, M. Zhu, B. Chen, et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, 2017.

[16] B. Kim, S. Chang, J. Lee, et al., "Broadcasted Residual Learning for Efficient Keyword Spotting," Proc. Interspeech, Brno, Czechia, August 2021, pp. 4521-4525.

[17] C. Banbury, V. J. Reddi, M. Lam, et al., "Benchmarking TinyML Systems: Challenges and Direction," arXiv preprint arXiv:2003.04821, 2020.

[18] STMicroelectronics, "STM32F746G-DISCO Discovery kit User Manual," UM1907, 2016.

[19] Y. Zeng, et al., "Sub 8-Bit Quantization of Streaming Keyword Spotting Models for Embedded Chipsets," 2022.

[20] Y. Shen, et al., "A KWS System for Edge-Computing Applications with Analog-based Feature Extraction and Learned Step Size Quantized Classifier," Preprints, 2025. DOI: 10.20944/preprints202503.0510.v1

[21] A. Sharma, "Hardware-Algorithm Co-Design for Energy-Efficient and Low-Latency Domain-Specific Machine Learning Systems," 2025.

[22] Y. Chen, et al., "DeltaKWS: A 65nm 36nJ/Decision Bio-inspired Temporal-Sparsity-Aware Digital Keyword Spotting IC with 0.6V Near-Threshold SRAM," IEEE Trans. Circuits Syst. I, 2024. DOI: 10.1109/TCASAI.2024.3507694

[23] C.-H. Chiang, et al., "A 14uJ/Decision Keyword Spotting Accelerator with In-SRAM-Computing and On Chip Learning for Customization," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 30, no. 8, 2022. DOI: 10.1109/TVLSI.2022.3172685

[24] H. Bos, et al., "Micro-power spoken keyword spotting on Xylo Audio 2," 2024.

[25] M. Jakuš, et al., "Implementing Keyword Spotting on the MCUX947 Microcontroller with Integrated NPU," 2025.

[26] L. Lai, N. Suda, V. Chandra, "CMSIS-NN: Efficient Neural Network Kernels for Arm Cortex-M CPUs," arXiv preprint arXiv:1801.06601, 2018.

[27] M. Hayat, et al., "TINY MACHINE LEARNING (TINYML) ADVANCEMENTS FOR INTELLIGENT BATTERY-POWERED IOT SENSORS," 2025.

[28] S. Dhungana, et al., "Domain-Incremental Continual Learning for Robust and Efficient Keyword Spotting in Resource Constrained Systems," 2025.

[29] Y. Zhang, et al., "Beacon: Post-Training Quantization with Integrated Grid Selection," arXiv preprint arXiv:2508.20293, 2025.

[30] H. Zakariyya, et al., "Quantitative Analysis of Deeply Quantized Tiny Neural Networks Robust to Adversarial Attacks," arXiv preprint arXiv:2503.08973, 2025.

[31] A. Mahmudov, et al., "QuantEdge: A Hybrid Quantization Approach for Optimized AI Deployment Across Edge Devices," IEEE Access, 2025. DOI: 10.1109/access.2025.3609798

[32] D. Nadalini, "Enabling Hardware-efficient On-Device Learning on Microcontroller-powered Ultra-low-power IoT Nodes," 2025.

[33] D. Peter, et al., "Resource-efficient DNNs for Keyword Spotting using Neural Architecture Search and Quantization," 2020.

[34] A. Mahmudov, et al., "QuantEdge: A Hybrid Quantization Approach for Optimized AI Deployment Across Edge Devices," IEEE Access, 2025. DOI: 10.1109/access.2025.3609798