An AR/VR SoC with Dual-Mode Gaze Guided Perception and Few-Shot Keyword Spotting

Yuncheng Lu[1], Junyi Qian[1,2], Aobo Li[1], Chengshuo Yu[1], Zehao Li[1], Ruixin Mao[1,3], Yucen Shi[1], Tianyi Liu[1], Yujin Lee[1], Junying Li[1], Chufeng Yang[1], Sen Cao[1], Jun Zhou[3], Weiwei Shan[2], Tony Tae-Hyoung Kim[1]
[1]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[2]School of Integrated Circuits, Southeast University, China
[3]School of Information and Communication Engineering, University of Electronic Science and Technology of China

***Abstract*—This paper presents an energy-efficient multimodal SoC for AR/VR that supports gaze tracking (GT), gaze-guided object recognition (OR), and few-shot keyword spotting (KWS). The SoC adopts: 1) a hybrid frame/event GT pipeline for low-cost always-on gaze sensing; 2) a neural processing unit with hierarchical inter-core data reuse and parallel integer Softmax for efficient DNN/Transformer execution; and 3) a progressive prototype matching scheme for few-shot KWS without backpropagation or dedicated matching hardware. Fabricated in 28 nm CMOS with 4 mm² area, the chip achieves 21.4 TOPS/W and 6.5 TOPS/W peak energy efficiency under INT4 and INT8, respectively. Compared with prior work, the proposed GT achieves 46× lower latency and 2586× higher energy efficiency, while the OR and KWS pipelines deliver competitive latency and energy efficiency.**

***Keywords*—AR/VR, edge AI, multi-modal SoC, gaze tracking, few-shot keyword spotting, Transformer accelerator**

I. Introduction

Augmented and virtual reality (AR/VR) devices have emerged as an important class of edge AI systems, typically requiring real-time audio and visual recognition to enable natural human–machine interaction. To further enhance interaction capability, the proposed system flow, shown in Fig. 1(a), supports real-time gaze tracking (GT), followed by gaze-guided object recognition on the user-attended region. In addition, it enables user-defined keyword spotting (KWS) through a few-shot learning scheme for personalized voice interaction. However, realizing such a multimodal AR/VR system on edge hardware remains challenging under tight power and latency constraints, as illustrated in Fig. 1(b). First, gaze tracking is an always-on function in AR/VR systems, and continuously running neural-network inference on incoming visual data leads to high power consumption and prolonged occupation of computing resources. Second, Transformer-based recognition suffers from severe latency bottlenecks caused by limited inter-core data reuse and sequential Softmax computation. Third, few-shot KWS generally relies on either floating-point backpropagation or dedicated hyperdimensional-computing (HDC) modules, both of which incur substantial hardware overhead.

To overcome these challenges, this work presents an energy-efficient multimodal SoC for AR/VR. It integrates: 1) a hybrid frame/event GT pipeline for low-cost continuous gaze sensing; 2) a neural processing unit with hierarchical interconnects and a parallel integer Softmax unit for efficient DNN/Transformer execution; 3) a progressive prototype matching scheme for few-shot user-defined KWS.

II. Proposed Energy-Efficient SoC for AR/VR

A. Overall Architecture

Fig. 2 shows the overall architecture of the proposed SoC,

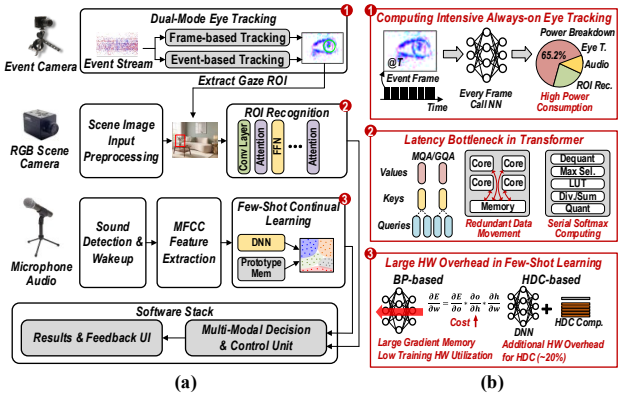

Fig. 1. (a) Overall processing flow and (b) main challenges.

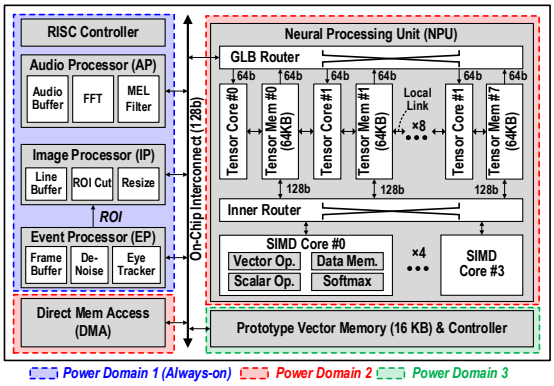

Fig. 2. Overall hardware architecture.

including: 1) a RISC controller for dataflow orchestration and global scheduling; 2) an audio processor (AP) for audio pre-processing, such as FFT and Mel-frequency filter; 3) an event processor (EP) for real-time eye tracking from DVS event streams; 4) an image processor (IP) for gaze-guided region of interest (ROI) extraction; 5) a neural processing unit (NPU) for accelerating deep-learning workloads; 6) a DMA for off-chip memory access; and 7) a 16 KB prototype memory for storing prototype vectors in few-shot learning. The NPU contains eight tensor cores (TCs) for matrix-matrix and matrix-vector computation, with parameters stored in eight tensor memories (TMs). To support flexible inter-core data reuse, the NPU integrates a global router. In addition, four SIMD cores accelerate vector processing and non-linear operations such as normalization and Softmax.

B. Compute-Efficient Event-based Eye Tracking Pipeline

Eye tracking is a key task in AR/VR systems, as it provides the gaze location for downstream tasks such as gaze-guided object recognition and foveated rendering. However, prior works [5], [6] process each event frame using computation-intensive neural networks (NN), leading to high power consumption. Moreover, continuous eye tracking occupies

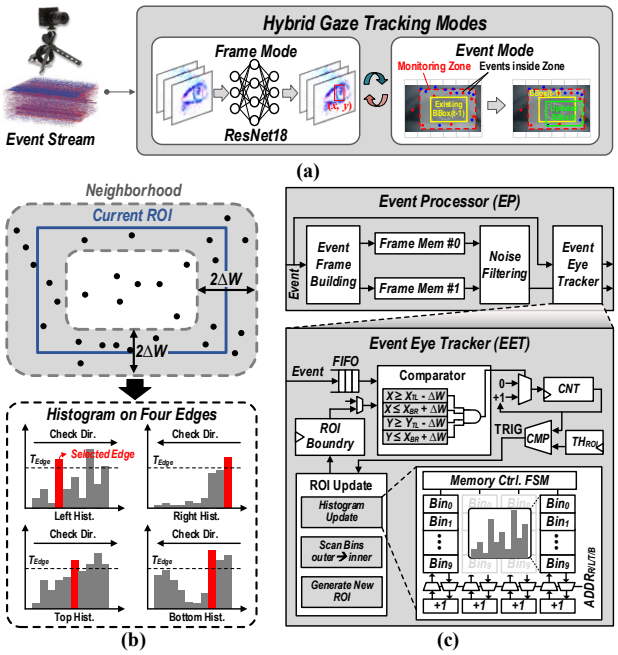

Fig. 3. (a) Algorithm flow of the event-based gaze tracking with hybrid modes, (b) principles of eye tracking in event mode and (c) hardware architecture of the EP.

computing resources that could otherwise serve concurrent AI tasks, thereby increasing overall system latency. To address this issue, we propose a hybrid-mode eye-tracking scheme, as shown in Fig. 3(a). In frame mode, events within a fixed time window are accumulated into a frame and processed by an NN to obtain an accurate gaze position. Since limited gaze deviation is tolerable, the system then switches to a lightweight event mode for continuous tracking.

As shown in Fig. 3(b), the event mode monitors only a ring-shaped neighborhood around the current ROI boundary instead of the full event frame. To estimate boundary movement efficiently, each ROI edge is associated with a 10-bin 1-D histogram covering a ±5-pixel offset range, and local events are accumulated according to their relative positions. Once the total local event count exceeds T_{ROI} (set to 50), the boundary is updated by selecting the first bin whose count exceeds T_{edge} for each edge; otherwise, the boundary remains unchanged for stability. After each update, the histograms are re-aligned to the new boundary for the next cycle. This event-driven scheme replaces continuous NN-based frame inference with simple histogram accumulation, making it hardware-friendly while suppressing noise-induced fluctuations.

The event eye tracker (EET) in the EP performs real-time ROI updates from the event stream (Fig. 3(c)). When an event (X, Y) falls within the ROI neighborhood, the counter (CNT) is incremented. Meanwhile, four 1-D histograms corresponding to four ROI edges are stored in four counter groups and updated in parallel. Once CNT exceeds T_{ROI}, the ROI boundary is updated by scanning each histogram from the outermost to the innermost bin and selecting the first bin whose count exceeds the threshold T_{edge}. The frame-based NN gaze tracker consumes 9.52 µJ and 719.6 µs per estimation, whereas the event mode requires only 0.13 µJ and 5.38 µs, achieving significantly lower energy and latency. However, prolonged event-mode operation introduces accumulated gaze error; therefore, frame-based recalibration is triggered periodically to restore accuracy. As shown in Fig. 4(a) and (b), increasing the recalibration interval reduces power and latency

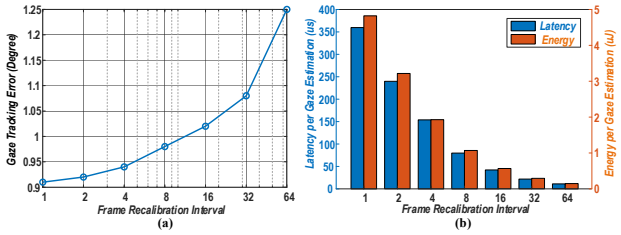

Fig. 4. Impact of frame recalibration interval on (a) gaze tracking error and (b) gaze tracking latency and energy.

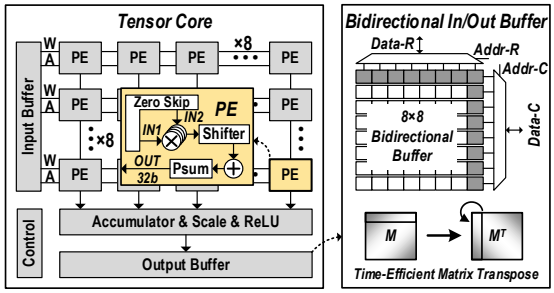

Fig. 5. Architecture of tensor core.

at the cost of higher gaze tracking error. Since an error of 0.9°–1.15°, corresponding to about 14–18 pixels on the display, is acceptable for gaze-guided object recognition [5], the recalibration interval is set to 32 in this work.

C. NPU with Flexible Data Reuse Scheme

The NPU contains eight tensor cores (TC) for parallel matrix computing. As shown in Fig. 5, each TC contains 64 processing elements (PE) supporting INT4/INT8 multiply and accumulate (MAC) operations with intermediate partial sums stored locally in 32-bit *Psum* registers. To eliminate the latency caused by frequent matrix transposition in attention operations, such as $Q \times K^T$, the input and output buffers are implemented using bi-directional 8-bit 8×8 register files (RF). These RFs support both row-wise and column-wise access, enabling direct data transfer to the tensor memory (TM). Compared with conventional row-wise buffers, this design enables seamless matrix transposition without waiting for an entire tile to be loaded, thereby improving TC utilization by 17.2% at the cost of 6% area and 5.2% power overhead per TC.

To support diverse data-reuse patterns in modern DNN and Transformer workloads, the NPU adopts a hierarchical interconnect that combines low-cost local links with a lightweight global router, as shown in Fig. 2 and Fig. 6. The local links provide direct communication paths between adjacent TCs and TMs, enabling efficient inter-TC pipelining for sequential computations. This is effective not only for inter-layer pipelining in DNNs, but also for chained operations in Transformer attention, such as $Q \times K^T$ followed by $S \times V$. By forwarding intermediate results directly between neighboring TCs, the proposed design avoids repeated write-back and reload through global shared memory, thereby reducing processing latency by 25.5% - 67.8% compared with conventional non-pipelined designs [7].

To further exploit cross-core data reuse, the global router in NPU supports broadcast and multicast transmission. This is particularly beneficial for attention variants such as multi-query attention (MQA) and Grouped-query attention (GQA), where multiple query heads share the same or grouped K/V tensors. Instead of duplicating data across multiple TMs, the global router allows one TM output to be concurrently

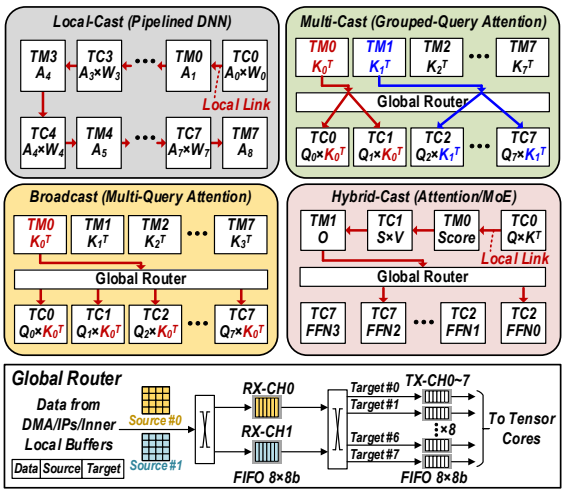

Fig. 6. Different inter-tensor core data scheduling in the NPU and implementation details of the global router.

delivered to multiple TCs, significantly reducing redundant data movement and external memory access (EMA). To balance flexibility and hardware cost, the global router adopts a 2-to-8 interconnection rather than a fully connected 8-to-8 topology. Since most reuse patterns require only a small number of simultaneous global sources, this design captures the majority of multicast cases with much lower area overhead. Moreover, the global router can operate jointly with the local links to form a hybrid-cast dataflow, supporting both inter-TC pipelining and cross-TC data sharing. ResNet18 and ViT-Base have been mapped onto the proposed NPU, achieving 1.25 ms and 2.53 mJ per frame for ResNet18, and 422 µs and 12.5 µJ per token for ViT-Base. As shown in Fig. 7, compared with conventional designs without flexible inter-core data sharing [3], [4], the proposed architecture significantly reduces latency and EMA while improving hardware utilization.

D. Parallel Integer Softmax Unit

Softmax is a critical operator in Transformer attention and is repeatedly executed across layers and heads. However, unlike matrix multiplication, it is less hardware-friendly due to the serial reduction, non-linear exponentiation, and division. Moreover, existing designs often rely on floating-point computation or sequential dequantization and requantization [7], leading to high hardware cost and delay.

To address this issue, this work adopts the log2-quantized integer Softmax scheme from SOLE [8] to enable a fully integer implementation. Instead of directly computing the exponential function, the exponent output is mapped into the log2 domain and approximated by simple shift-and-add operations. To avoid overflow, safe Softmax is adopted by subtracting the maximum input X_{max} from each X_i. The exponent term is then approximated as Q_i in (1), avoiding large LUTs and multipliers. The normalization step divides each Q_i by the accumulated sum S, which can be expressed as (2), where the division is decomposed into a shift term determined by the leading-one positions of Q_i and S, and a residual term $1/(1+\alpha)$, with $S = 2^{KS} \times (1+\alpha)$, and $\alpha \in (0,1)$.

$$Q_i = Log2EXP(X) \approx -round(X + X \gg 1 - X \gg 4) \quad (1)$$

$$Q_i / S = 2^{-(K_{Qi}+K_S)} \cdot (1/1+\alpha) \quad (2)$$

Fig. 8 shows the proposed parallel integer Softmax unit, which consists of four pipeline stages. First, a 32-input 4-level

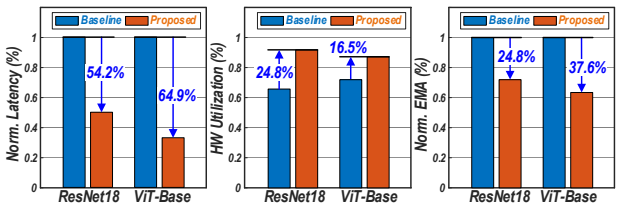

Fig. 7. Performance improvement compared with conventional design.

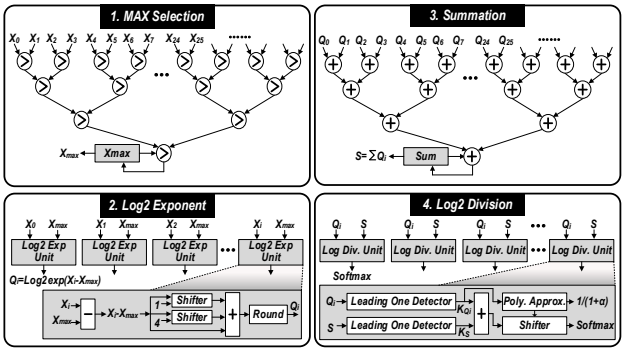

Fig. 8. Implementation details of the parallel integer Softmax unit.

comparator tree computes X_{max} in parallel. Second, 32 parallel Log2Exp units compute all Q_i using only shifters and adders. Third, a 32-input 4-level adder tree accumulates all Q_i to generate S. Finally, 32 parallel LogDiv units generate the Softmax outputs: for each element, the leading-one position of Q_i and S determine the shift factor (K_{Qi} + K_S), while the residual term $1/(1+\alpha)$ is obtained by polynomial approximation. Compared with conventional Softmax hardware, the proposed design removes serial dequantization and requantization and avoids LUT-based exponentiation and division. Owing to the fully integer datapath and parallel reduction structure, the proposed unit reduces Softmax latency by 89% and power consumption by 78.2% compared with [7], with only ~1% accuracy degradation when running ViT-B.

E. Few-Shot KWS Processing Pipeline

Few-shot KWS is adopted in this work to support user-defined voice commands, improving system flexibility and personalization. However, prior few-shot learning accelerators mainly rely on backpropagation-based adaptation, which requires floating-point computation, dedicated training hardware, and extra gradient storage, leading to high area and energy overhead. HDC-based methods avoid floating-point retraining, but still require dedicated hypervector-generation and matching modules [1], [2], introducing ~20% additional area and power.

To address these limitations, this work proposes a progressive prototype matching (PPM) scheme, as shown in Fig. 9(a). A meta-learned network is trained offline to extract discriminative speech embeddings. During inference, the input voice is converted into a 1024-D query vector (QV), while the mean embedding of each learned command is stored as a prototype vector (PV). Classification is then performed by Manhattan-distance comparison between the QV and PVs, avoiding gradient-based retraining and enabling lightweight continual adaptation. Unlike conventional prototype-based classifiers that compare only the final-layer embedding, the proposed method progressively matches compressed intermediate-layer features with their corresponding prototype features. If the similarity exceeds a layer-dependent threshold, inference terminates early. To ensure robustness, earlier layers use stricter thresholds than later layers. Although the

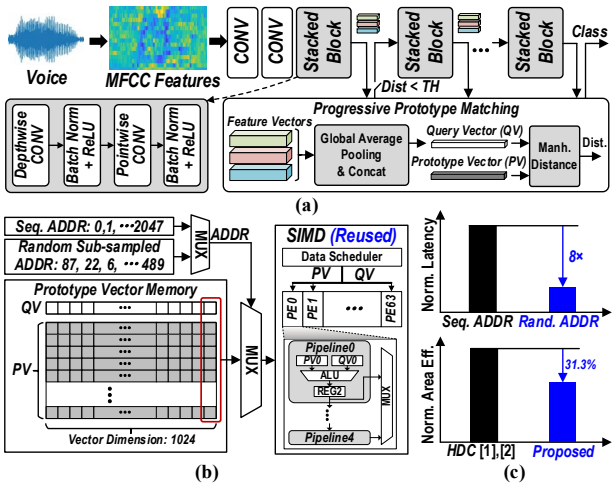

Fig. 9. (a) Algorithm flow of the few-shot KWS, (b) hardware processing pipeline for progressive prototype matching, and (c) performance improvement.

prototype memory increases by 4×, experiments on the Google Speech Commands dataset show that the proposed early-exit scheme skips 62.3% of the total computation on average with less than 2% accuracy degradation under the 32-way 5-shot setting.

As shown in Fig. 9(b), the QV and PVs are stored in prototype vector memory, while the SIMD units in the NPU are reused to perform parallel Manhattan-distance computation. In each cycle, the same element of the QV and all PVs are read out in parallel. To further reduce comparison latency, a random element sub-sampling scheme is adopted. When the sampled length is at least 128, the accuracy loss remains within 1%; therefore, a subsampling ratio of 1/8 is used, reducing comparison latency by 8×. Since the proposed matching fully reuses the existing SIMD datapath, no additional matching hardware is required, achieving 31.3% area reduction compared with HDC-based designs [1], [2].

III. Evaluation

Fig. 10 shows the chip micrograph and measured results. The proposed chip is fabricated in 28 nm CMOS and occupies an active area of 4 mm². It operates from 0.6 V to 1 V over 50–435 MHz, achieving peak energy efficiencies of 21.4 TOPS/W and 6.5 TOPS/W for INT4 and INT8, respectively. Table I summarizes the comparison with prior works. To the best of our knowledge, this work is the first multimodal SoC that unifies end-to-end gaze tracking (GT), gaze-guided object recognition (OR), and keyword spotting (KWS) on a single platform. Owing to the proposed hybrid-mode GT, it achieves 46× lower latency and 2586× higher energy efficiency than [10]. For OR and KWS, the chip also delivers competitive latency and energy efficiency among prior works, benefiting from the hierarchical inter-core data reuse, optimized integer Softmax pipeline, and adaptive inference skipping enabled by progressive prototype matching.

IV. Conclusion

This paper presents a multimodal AR/VR SoC that unifies always-on gaze tracking, gaze-guided object recognition, and few-shot keyword spotting on a single low-power platform. By combining a hybrid frame/event GT pipeline, a hierarchical NPU with parallel integer Softmax, and a PPM-based few-shot KWS scheme, the proposed chip efficiently supports heterogeneous AR/VR workloads. Fabricated in 28 nm CMOS, it achieves up to 21.4 TOPS/W and 6.5 TOPS/W under INT4 and INT8, respectively. Measured results validate substantial gains in latency and energy efficiency, demonstrating its suitability for always-on and personalized AR/VR edge intelligence.

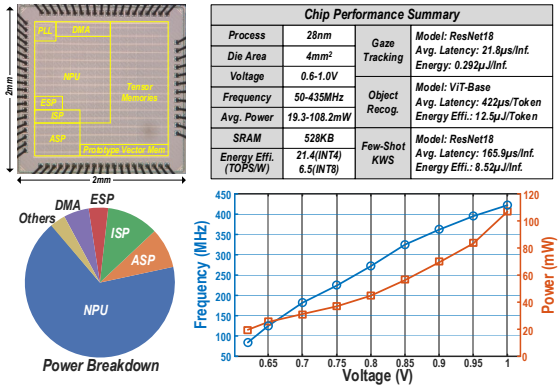

Fig. 10. Die photo and measurement results.

TABLE I COMPARASON WITH PRIOR WORKS

	This Work			JSSC'26[3]	JSSC'24[4]		JSSC'26[7]	SSCL'26[10]	JSSC'26[9]
Process (nm)	28			22	22		16	16	40
Die Area (mm^2)	4			5.8	12		10.15	16	0.74
Voltage (V)	0.6-1			0.55-0.9	0.44-1		0.45-0.85	0.85	0.6-1.1
Memory	528KB			576KB	3.98MB		1.32MB	4.5MB	71KB
Power (mW)	19.3-108.2			19.6-197.1	0.25-3.84		7.12-152.5	NA	11.6
Freq. (MHz)	50-435			75-400	70		60-450	345	150
Precision	INT4/8			INT4/8	INT8		INT4/8/16	FP8	INT4
Modality	Image/Audio/Event			Image	Image/Audio		Image/ Audio	Image	Audio
Application[3]	GT	OR	KWS	OR	OR	KWS	OR	GT	KWS
Latency	21.8 μs/Inf.[2]	422μs/Token[1] 1.25ms/Inf.[2]	165.9 μs/Inf.[2]	—	184.9 ms/Inf.[2]	—	567 μs/Token[1]	1.01 ms/Inf.[2]	0.59 ms/Inf.[2]
Energy	0.29 μJ/Inf.[2]	12.5 μJ/token[1] 2.53mJ/Inf.[2]	8.52 μJ/Inf.[2]	94.2 μJ/Token[1]	946 μJ/Inf.[2]	—	3.66 μJ/Token[1]	0.75 mJ/Inf.[2]	6.84 μJ/Inf.[2]
Accuracy	0.92°-1.3°[5]	94.4%[2,5] 76.8%[1,6]	97.0%[7]	—	94.5%[2,5]	—	76.7%[1,6]	—	98.8%[7]
Energy Effi. (TOPS/W)[4]	21.4 (INT4)[1] 6.5 (INT8)[1]			4.46 (INT8) 25.1 (INT4)	3.1 (INT8)		15.2-40.3 (INT8)	—	6 (INT4)

1: Evaluated using ViT-Base; 2: Evaluated using ResNet18;
3: GT: Gaze Tracking, OR: Object Recognition, KWS: Keyword Spotting; 4: Off-chip memory access power excluded.
5: EBVEYE Dataset; 5: COCO2017 dataset; 6: ImageNet Dataset; 7: Google Speech Commands Dataset.

References

[1] H. Yang, et al., "Fsl-hdnn: A 5.7 tops/w end-to-end few-shot learning classifier accelerator with feature extraction and hyperdimensional computing," *ESSERC*, 2024.

[2] C. E. Song, *et al.*, "Clo-HDnn: A 4.66 TFLOPS/W and 3.78 TOPS/W Continual On-Device Learning Accelerator with Energy-Efficient Hyperdimensional Computing via Progressive Search," *VLSI*, 2025.

[3] Q. Zhang, *et al.*, "MITTA: A Multi-Task Transformer Accelerator With Mixed Precision Structured Sparsity and Hierarchical Task-Adaptive Power Management," *JSSC*, 2026.

[4] Z. Fan, et al., "AIMMI: Audio and Image Multi-Modal Intelligence via a Low-Power SoC With 2-MByte On-Chip MRAM for IoT Devices," *JSSC*, 2024.

[5] S. Tan, *et al.*, "Toward Efficient Eye Tracking in AR/VR Devices: A Near-Eye DVS-Based Processor for Real-Time Gaze Estimation," *TCAS-I*, 2025.

[6] T. Han, *et al.*, "JaneEye: A 12-nm 2K-FPS 18.9-μJ/Frame Event-based Eye Tracking Accelerator," *ASP-DAC*, 2026.

[7] S. Moon, *et al.*, "T-REX: Hardware–Software Co-Optimized Transformer Accelerator With Reduced External Memory Access and Enhanced Hardware Utilization," *JSSC*, 2026.

[8] W. Wang, *et al.*, "SOLE: Hardware-software co-design of softmax and layerNorm for efficient transformer inference," *ICCAD*, 2023.

[9] D. d. Blanken, *et al.*, "Chameleon: A Multiplier-Free Temporal Convolutional Network Accelerator for End-to-End Few-Shot and Continual Learning from Sequential Data," *JSSC*, 2026.

[10] K. Feng, *et al.*, "Birch: A Real-Time Multi-Domain Multi-Task Extended Reality Perception Accelerator," *Solid-State Circ. Letters*, 2026.