JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 31 No. 7, pp. 1-11, July 2026
https://doi.org/10.9708/jksci.2026.31.07.001

Performance Characterization of Distributed Inference on Heterogeneous Edge Device Clusters Compared with Single-Device Omnimodal Inference

Pil-Seong Jeong*

*Professor, Dept. of Information and Communication Engineering, Myongji College, Seoul, Korea

[Abstract]

This study compares modality-specific specialized models distributed across a heterogeneous edge cluster with single-device omnimodal deployment under identical tasks. A cluster of Jetson AGX Orin, Raspberry Pi 5+Hailo-8, and Orange Pi 5 Plus runs MobileViT-S, Whisper-tiny, and DistilBERT via gRPC, benchmarked against LLaVA-1.5-7B FP16, its INT4 quantization, and Qwen2-VL-2B FP16. The distributed configuration is 80–288× faster on Vision, 3.7–25× faster on Text, and up to 806× more energy-efficient, with Text F1 within 2.8 percentage points across all four configurations. Effect decomposition attributes 3.93× to quantization and 1.75× to model size reduction, with an additional 63.5× from distribution itself, confirming independence from model size and quantization. The 17–195× cost-normalized advantage is retained across five wired-to-wireless network scenarios; distributed clusters thus suit concurrent multi-modal serving, while omnimodal deployment suits cross-modal reasoning.

▸ **Key words**: Edge AI, Distributed Inference, Heterogeneous Edge Cluster, Specialized Model, Omnimodal Model, Performance Characterization

[요 약]

본 연구는 이기종 엣지 클러스터의 모달리티별 특화 모델 분산 배치를 단일 디바이스 옴니모달 배포와 동일 태스크에서 비교한다. AGX Orin, Raspberry Pi 5+Hailo-8, Orange Pi 5 Plus에서 MobileViT-S, Whisper-tiny, DistilBERT를 gRPC 분산 추론하고 LLaVA-1.5-7B FP16·INT4 및 Qwen2-VL-2B FP16과 비교한 결과, 분산 구성은 Vision 80~288배, Text 3.7~25배 빠르고 최대 806배 효율적이며 Text F1 격차 2.8%p 이내를 유지하였다. 효과 분리에서 양자화 3.93배·모델 크기 1.75배에 더해 분산 자체로 추가 63.5배가 확인되어 분산의 이점이 모델 크기·양자화와 독립적임이 입증되었다. 17~195배의 비용 정규화 우위는 5종 네트워크 시나리오에서도 유지되었고, 동시 멀티모달 서빙에는 분산, 교차 모달 추론에는 옴니모달이 적합하다.

▸ **주제어**: 엣지 인공지능, 분산 추론, 이기종 엣지 클러스터, 특화 모델, 옴니모달 모델, 성능 특성 분석

• First Author: Pil-Seong Jeong, Corresponding Author: Pil-Seong Jeong
*Pil-Seong Jeong (ibetter.kr@gmail.com), Dept. of Information and Communication Engineering, Myongji College
• Received: 2026. 03. 19, Revised: 2026. 05. 23, Accepted: 2026. 07. 06.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

스마트 홈·IoT 등 멀티모달 데이터를 동시에 다루는 응용이 확산되면서 영상·음성·텍스트를 결합한 엣지 AI 추론 수요가 증가하고 있다. 클라우드 기반 추론은 네트워크 지연·개인정보·통신 비용·가용성 문제를 동반하므로 데이터를 발생 지점 근처에서 처리하는 엣지 AI(Edge AI)가 대안으로 부각되었으나, 엣지 디바이스는 제한된 메모리(수~수십 GB), 전력(수~수십 W), 가속기(수~수백 TOPS) 자원에서 동작해야 한다[1]. 본 논문에서는 옴니모달 모델(omnimodal model)을 단일 모델로 다수 모달리티 입력을 처리하는 모델(LLaVA, Qwen2-VL 등), 멀티모달 추론(multimodal inference)을 다수 모달리티의 입출력을 다루는 추론 일반, 분산 추론(distributed inference)을 다수 디바이스에 모델을 분배하여 수행하는 추론으로 정의하여 일관되게 사용한다.

이러한 자원 제약 환경에서 멀티모달 추론을 구현하는 배포 전략은 두 가지 패러다임으로 구분된다. 첫째, 단일 고성능 디바이스에 옴니모달 모델을 배포하는 전략으로, LLaVA, Qwen2-VL과 같은 7B급 비전-언어 모델(Vision-Language Model, VLM)을 NVIDIA Jetson 등에 직접 배포하는 접근이다[2-4]. 옴니모달은 단일 추론 경로로 다중 모달리티를 처리하고 교차 모달 추론(cross-modal reasoning, VQA 등)을 지원한다. 둘째, 이기종 엣지 디바이스 클러스터에 모달리티별 특화 모델을 분산 배치하는 전략이다. MobileViT(이미지)[5], Whisper-tiny(음성)[6], DistilBERT(텍스트)[7] 등 경량 모델을 GPU·NPU·CPU에 매핑하여 동시 병렬 처리하는 접근이며[8-10], 모델 간 데이터 의존성(data dependency, 한 모델의 출력이 다른 모델의 입력으로 연결되는 관계)이 없는 단일 모달리티 태스크에서 가속기 활용도와 처리량을 동시에 향상시킬 잠재력이 있다.

두 패러다임의 실측 기반 정량 비교는 제한적이다. 기존 분산 추론 연구는 DNN 파티셔닝[9,11]이나 동질적 가속기 클러스터의 단일 모달리티 추론[8]에 집중되어, 이기종 가속기에 모달리티 단위로 분산한 구성과 옴니모달의 직접 비교 사례는 거의 보고되지 않았다[12]. 또한 엣지 NPU(Hailo-8 등)는 2D CNN 위주로 설계되어 1D 컨볼루션·Transformer 호환성 제약이 있으나[13,14] GPU+NPU+CPU 통합 추론의 실측 평가는 부재하며, 비용 비대칭 클러스터의 공정 비교를 위한 정규화 지표(Throughput/$, Accuracy/W, Accuracy/J)는 정착되지 않은 상태이다[15].

본 연구의 목적은 “이기종 엣지 디바이스 클러스터의 모달리티별 특화 모델 분산 배치 전략이 단일 디바이스 옴니모달 배포와 비교하여 어떤 성능 특성을 보이는가”라는 질문에 정량적으로 답하는 것이다. 이를 위해 3대 이기종 클러스터(GPU+NPU+CPU)에서 3종 특화 모델을 4종 런타임으로 gRPC 분산 추론하고, 동일 AGX Orin에 LLaVA-1.5-7B FP16과 분산 구조·모델 크기·양자화 효과 분리를 위한 LLaVA INT4, Qwen2-VL-2B 베이스라인을 비교 측정하였다. 본 연구의 주요 기여는 첫째, 분산과 옴니모달의 동일 태스크 실측 비교에 양자화·경량 VLM 베이스라인을 추가하여 분산 구조 효과와 모델 크기·양자화 효과를 분리 분석한 점, 둘째, 비용 정규화 통합 평가 프레임워크 제안, 셋째, 엣지 NPU 호환성 제약과 CPU fallback 설계 패턴의 정량 보고이다.

본 논문의 구성은 다음과 같다. 2장에서는 엣지 AI 추론 프레임워크, 모델 경량화, 엣지 LLM/VLM 배포, 분산·협력 추론, 이기종 가속기 통합, 정규화 평가 방법론에 관한 선행 연구를 분석한다. 3장에서는 시스템 설계, 4장에서는 실험 설계, 5장에서는 실험 결과를 제시하며, 6장에서 논의 및 한계, 7장에서 결론 및 향후 연구 방향을 제시한다.

II. Preliminaries

1. Inference Frameworks and Model Compression for Edge AI

엣지 AI 배포의 핵심 기반은 가속기별 추론 프레임워크와 모델 경량화 기법이다. NVIDIA TensorRT는 레이어 융합과 커널 자동 튜닝으로 NVIDIA GPU 추론을 가속하며 Jetson 시리즈의 표준 런타임이고, ONNX Runtime은 ARM CPU 환경의 동적 양자화 INT8 추론에 적합하며, HailoRT는 Hailo-8 NPU 전용으로 사전 컴파일된 HEF 모델을 INT8로 실행한다. llama.cpp[16]는 GGUF 포맷의 4-bit 양자화(Q4_K_M 등)를 지원하여 자원 제약 환경에서 LLM/VLM 실행을 가능하게 하였다. MLPerf Inference[15]는 P50/P95 지연시간 등 측정 프로토콜의 표준을 제시하여 본 연구 측정 프로토콜 설계의 기반이 되었다.

모델 경량화는 경량 아키텍처, 양자화, 증류·가지치기로 구분된다. 본 연구가 사용한 MobileViT[5]는 5.6M 파라미터로 ImageNet Top-1 78.3%를 달성하였고, Whisper-tiny[6]는 39M 다국어 ASR, DistilBERT[7]는 BERT를 지식 증류로 압축하여 66M으로 BERT 성능의

97%를 유지한다. INT4 양자화는 LLM 배포의 표준이 되고 있으나 구현체별 호환성 차이가 커 bitsandbytes의 NF4 등은 Jetson aarch64에서 동작하지 않는다. 이 환경에서는 llama.cpp Q4_K_M, MLC-LLM, TensorRT-LLM AWQ INT4 등 대안 경로가 필요하며[16,17], 본 연구는 이를 우회하기 위해 LLaVA-1.5-7B INT4 베이스라인 측정에 llama.cpp Q4_K_M을 채택하였다.

2. Edge LLM and VLM Deployment

옴니모달 모델 측면에서 LLaVA[2]는 CLIP ViT-L 비전 인코더와 Vicuna 언어 모델을 결합한 7B, 13B VLM이며, Qwen2-VL[3]은 동적 해상도와 멀티모달 RoPE를 도입한 경량 VLM 계열(2B, 7B, 72B)로 자원 제약 환경에 적합하다. Edge LLM 서베이[4]는 KV 캐시 압축, Speculative Decoding 등 엣지 배포 최적화 기법을 정리하였다. 이러한 옴니모달 모델은 단일 디바이스에서 다중 모달리티를 통합 처리하는 장점이 있으나, 7B 이상 FP16 추론에 13GB 이상의 GPU 메모리와 본 연구 측정에서 8.2초의 단일 이미지 VQA 지연시간이 발생하여 실시간성에 한계가 있다. Audio 모달리티는 LLaVA, Qwen2-VL이 미지원이며 Qwen2-Audio[18] 등 별도 모델이 필요하다.

3. Distributed Inference and Heterogeneous Accelerator Integration

분산 또는 협력 추론은 DNN 레이어 분할, 멀티 모델 협력, 모달리티 단위 분산의 세 가지 흐름으로 구분된다. 레이어 분할 계열에서는 Neurosurgeon[9]이 모바일-클라우드 간 최적 분할 지점을 동적으로 결정하였고, DeepThings[8]는 IoT 엣지 클러스터에서 fused tile partitioning으로 CNN을 분산 실행하였으며, CoFormer[11]는 GPT2-XL을 이기종 엣지에 자동 분해 배포하여 메모리 76.3% 절감을 달성하였다. 이 접근은 단일 모델의 레이어를 분할하므로 디바이스 간 데이터 전송이 필수적이며, 통신 비용이 분산의 이점을 상쇄할 위험이 있다. 모델 협력 계열의 S2M3[12]는 멀티모달 모델을 기능 수준에서 분리하고 공통 모듈을 공유하여 메모리·지연시간을 50% 이상 감소시켰고, SLO 기반 우선순위 라우팅과 강화학습 기반 라우팅 등도 제안되고 있다. 모달리티 단위 분산 계열의 Nanomind[19]와 DAISY[20]는 모달리티별 분리 처리를 시도하였으나 이기종 가속기 매핑(GPU, NPU, CPU 혼합)을 명시적으로 다루지 않거나 옴니모달과의 정량 비교를 제공하지 않았다.

이기종 가속기 통합은 단일 SoC와 다중 디바이스 차원에서 진행된다. 단일 SoC 측면에서 ADMS[10]는 CPU·GPU·NPU 동시 활용으로 멀티 DNN 4배 가속을 달성하였고, 다중 디바이스 측면에서는 Triton Inference Server 등이 gRPC 서빙 인프라를 제공하나 동질적 서버 클러스터를 가정한다. 엣지 NPU의 호환성 제약은 분산 설계의 주요 변수이며, Hailo-8과 같은 비전 중심 NPU는 2D CNN에 최적화되어 1D 컨볼루션, Transformer, Attention을 완전히 지원하지 못하므로[13,14] 모달리티별 가속기 매핑은 연산자 호환성 검증 후 결정되어야 한다.

4. Differentiation of This Work

기존 연구는 DNN 파티셔닝[8,9,11], 모델 협력[12], 모달리티 분산[19,20]의 세 계열로 구분되며, 본 연구와의 차별성을 Table 1에 정리하였다. 본 연구의 차별성은 (1) 모달리티별 독립 모델로 데이터 의존성을 제거하여 완전 병렬 처리, (2) 3종 가속기·4종 런타임의 gRPC 실측 통합, (3) 양자화·경량 VLM 베이스라인 동시 측정으로 분산 구조 효과와 모델 크기·양자화 효과의 분리 분석, (4) 비용 정규화 평가 프레임워크 도입, (5) NPU 호환성 제약과 CPU fallback 설계 패턴의 정량 보고에 있다.

Table 1. Comparison with Prior Work

Criterion	partition	coop.	Modality	Proposed
Hetero. GPU+NPU +CPU	Partial	No	Partial	Yes
4-way omnimodal comp.	No	No	No	Yes
Cost-norm. + NPU evid.	No	No	No	Yes

III. System Design

1. Hardware Configuration

본 연구에서는 3종 이기종 가속기(GPU, NPU, CPU)를 탑재한 엣지 디바이스 3대로 분산 추론 클러스터를 구성하였다. Table 2는 각 디바이스의 사양과 역할을 요약한다. 중앙 노드인 Jetson AGX Orin 64GB는 NVIDIA Ampere GPU(2,048 CUDA 코어, 64 Tensor 코어)를 탑재하여 275 TOPS(INT8)의 연산 성능을 제공하며, 오케스트레이터 역할과 TensorRT FP16 기반 이미지 분류 추론을 동시 수행한다. Raspberry Pi 5 8GB에는 Hailo-8 AI

Table 2. Hardware Specifications of the Heterogeneous Edge Device Cluster

Device	Specification	AI Accelerator	Role	Cost
AGX Orin 64GB	Ampere GPU, 64GB	275 TOPS (INT8)	• Vision(Classification) • Orchestrator	$699
RPi5 + Hailo-8 HAT+	Cortex-A76, 8GB	NPU 26 TOPS	Audio(Whisper, CPU), Vision(YOLO, NPU)	$150
OPi5 Plus + Hailo-8	RK3588, 32GB	NPU 26 TOPS	Text(DistilBert, CPU)	$180

HAT+(PCIe 3.0)를 장착하여 26 TOPS(INT8)의 NPU 가속 성능을 확보하였으며, 음성 인식(Whisper-tiny, PyTorch CPU)을 수행한다. Orange Pi 5 Plus 32GB에는 Hailo-8 NVMe(M.2)를 연결하여 ablation study용 NPU를 확보하고, 텍스트 QA(DistilBERT, ONNX Runtime INT8)는 RK3588 CPU에서 실행한다. 세 디바이스는 1Gbps 유선 이더넷으로 연결하였으며, C1(Distributed)의 총 하드웨어 비용은 약 $1,030이다.

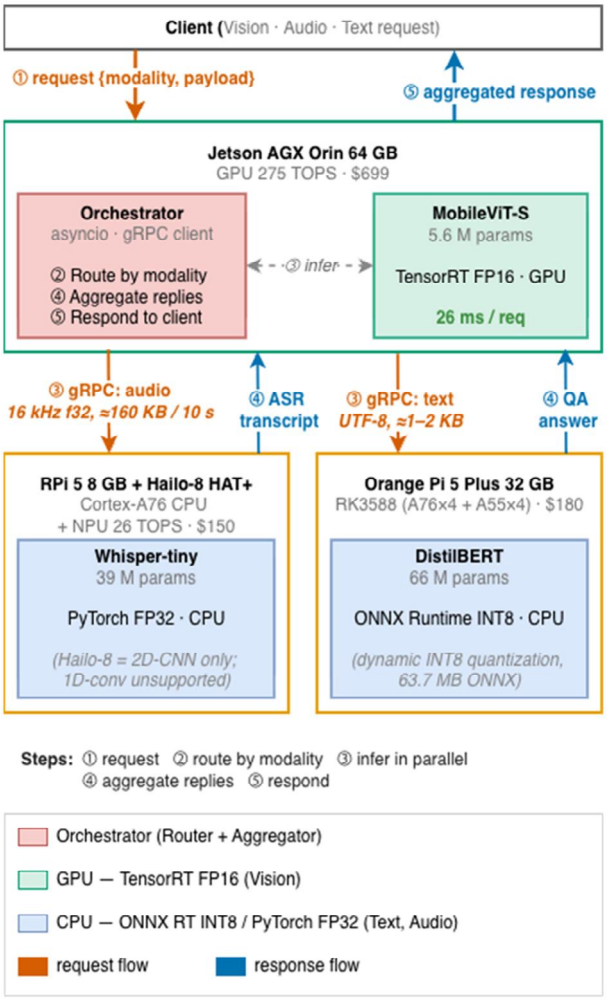

Fig. 1. System architecture of the heterogeneous edge cluster with gRPC star topology and request flow

2. Software Architecture

Fig. 1은 gRPC 기반 Star 토폴로지의 분산 추론 아키텍처를 나타낸다. AGX Orin의 오케스트레이터는 asyncio.gather 기반 비동기 병렬 gRPC 호출로 요청의 모달리티를 판별하여 해당 에이전트로 라우팅하고 결과를 수집한다. 이미지는 바이트 배열, 텍스트는 UTF-8 문자열, 오디오는 16kHz float32 waveform(~160KB/10초 샘플)으로 인코딩하여 전송한다. 주요 소프트웨어 환경은 AGX Orin(JetPack 6.2.2, CUDA 12.6, TensorRT 10.3, PyTorch 2.5.0), RPi5(Debian 13, PyTorch 2.5.1 CPU), OPi5(Ubuntu 24.04, ONNX Runtime 1.19.2)이며, 전 디바이스에서 Python 3.10과 gRPC 1.68.1을 사용한다.

3. Model Configuration

분산 조건의 특화 모델은 Vision 분류용 MobileViT-S[5](5.6M, TensorRT FP16), Audio 음성 인식용 Whisper-tiny[6](39M, PyTorch FP32), Text QA용 DistilBERT[7](66M, ONNX Runtime INT8)로 구성되며 각각 AGX Orin GPU, RPi5 CPU, OPi5 Plus CPU에 배치된다. DistilBERT는 ONNX Runtime 동적 양자화로 INT8 변환하여 실행한다.

옴니모달 조건은 LLaVA-1.5-7B[2](7B, PyTorch FP16)를 AGX Orin에서 단독 실행한다. 이 VLM은 Vision과 Text를 처리하나 Audio는 미지원이다. bitsandbytes INT4 양자화가 Jetson aarch64의 CUDA custom kernel을 지원하지 않아 FP16으로 실행하였으며, llama.cpp 및 vLLM은 LLaVA-1.5의 멀티모달 인코더(CLIP ViT-L)를 완전히 지원하지 않아 적용하지 못하였다. 옴니모달 조건은 분산 조건과 동일 AGX Orin에서 순차 실험하여 하드웨어 변수를 통제한다.

IV. Experimental Design

1. Conditions, Tasks, and Scenarios

분산 조건(3대 클러스터, ~$1,030)과 옴니모달 조건(AGX Orin 단독 LLaVA-7B, ~$700)의 47% 비용 비대칭

을 보정하기 위해 정규화 지표(Throughput/$, Accuracy/W, Accuracy/J)를 도입하였다. 두 조건은 동일 입력 샘플을 사용하며, 옴니모달의 Audio 미지원으로 Vision·Text에서만 직접 비교하고 교차 모달 태스크(VQA 등)는 분산 구조의 비대상이므로 제외하였다. 시나리오는 (1) 개별 태스크 비교, (2) 3종 동시 서빙, (3) gRPC 통신 오버헤드의 세 가지이다.

Table 3. Evaluation Tasks and Datasets

Task	Dataset	Samples	Metric
Image Classification	COCO val2017	50 (224×224)	Top-1 Accuracy (%)
Text QA	SQuAD v2.0 dev	50 (≤512 tokens)	F1 Score
Speech Recognition	LibriSpeech test-clean	50 (≤10s)	WER (%)

절대 지표로는 E2E Latency(P50/P95), Throughput(req/s), Energy/Request(J)를, 정규화 지표로는 Throughput/$1K, Accuracy/W, Accuracy/J를 채택하였다. 여기서 Throughput/$1K는 하드웨어 비용 1천 달러당 초당 처리 요청 수를 의미하며, 조건 간 비용 비대칭(분산 vs. 옴니모달)을 보정하는 역할을 한다. 전력은 AGX Orin은 tegrastats(INA3221) 내장 센서로, RPi5와 OPi5 Plus는 FNIRSI-FNB58 USB 전력계로 측정하였다. 요청당 에너지는 평균 전력과 추론 시간의 곱($E = \overline{P} \times t$)으로 산출하였으며, 이는 추론 구간 내 전력이 일정하다는 가정에 기반한 근사값이다. 실측 결과 RPi5의 Whisper-tiny 추론은 7.52~11.68W 범위에서 전력이 변동하였으나, 4-way 비교에 동일 근사 가정을 일관되게 적용하여 조건 간 상대 비교의 타당성을 확보하였다.

2. Statistical Analysis

TensorRT engine build, CUDA context 초기화 후 warm-up 5회를 실행하였다. 표본 수는 분산 조건 모달리티당 n=50, 옴니모달 조건 n=20(요청당 3~11초의 시간 제약)이며, Cohen's d > 2에서 t-test 필요 표본이 약 5(α =0.05, power=0.80)이므로 두 표본 모두 충분한 검정력을 확보한다. 검정 절차는 두 갈래로 분리하여 적용하였다. 분산 조건 내 개별 서빙과 동시 서빙 간 비교(paired comparison)에는 차분 분포에 대해 Shapiro-Wilk 정규성 검정을 선행하여 $p > 0.05$이면 paired t-test, $p \leq 0.05$이면 Wilcoxon signed-rank test를 적용하는 결정 규칙을 채택하였다. 본 연구 데이터의 차분 분포 정규성 검정 결과는 Vision $W=0.894$ $p=2.96\times10^{-4}$, Text $W=0.911$ $p=1.14\times10^{-3}$, Audio $W=0.941$ $p=1.44\times10^{-2}$로 세 모달리티 모두 정규성 가정을 기각하였으므로 paired 비교에는 일관되게 Wilcoxon signed-rank test를 채택하였다. 분산 조건과 옴니모달 조건 간 비교(independent comparison)는 옴니모달의 raw sample을 보유하지 못하여 mean·SD·n의 요약 통계 기반 Welch's t-test를 적용하고 Cohen's d로 효과 크기를 보고한다. 다중 비교에는 모달리티별 두 비교(개별-동시, 분산-옴니모달)를 묶어 Bonferroni 보정(α=0.025)을 적용하였다. 검정 선택의 강건성 점검을 위해 paired t-test sensitivity check를 병행한 결과 Vision $t=-25.63$ $p=4.81\times10^{-30}$, Text $t=-14.40$ $p=3.08\times10^{-19}$, Audio $t=-0.04$ $p=0.965$로 Wilcoxon 결과와 동일한 유의성 결론을 보였다.

V. Experimental Results and Analysis

1. Per-Task Performance Comparison

분산 구성을 옴니모달 베이스라인 3종(FP16, INT4 양자화, 경량)과 동일 태스크 조건에서 비교하였다. INT4 양자화 베이스라인은 동일 LLaVA-1.5-7B 모델의 Q4_K_M 양자화를 llama.cpp로 실행한 구성이며, 경량 베이스라인은 동일한 PyTorch+transformers 런타임에서 모델 크기를 1/3.5로 축소한 Qwen2-VL-2B FP16이다. 4종 조건 모두 동일 AGX Orin에서 측정하여 하드웨어 변수를 통제하였다. INT4 양자화·경량 베이스라인은 30회 반복 측정(warm-up 3회)이며, 그 외 조건은 본 연구의 1차 실험 표본을 유지한다.

Fig. 2는 3종 태스크의 지연시간 분포를 나타낸다. Table 4는 3종 태스크에 대한 4-way 지연시간 성능을 나타낸다. 오디오 모달리티는 옴니모달 3종 모두 입력 미지원이므로 측정 불가하며, 향후 Qwen2-Audio 등과의 4-way 비교는 7장에 명시하였다. Table 5는 4-way 통계 검정과 전체 데이터셋 품질 평가 결과를 통합하여 나타낸다.

분산 조건은 Vision에서 옴니모달 3종 대비 80-288배, Text에서 3.7-25배 빠르며 모든 비교가 $d > 1.4$, $p < 1\times10^{-5}$로 통계적으로 유의하였다. INT4 양자화 베이스라인과의 비교에서도 우위가 유지되어 분산 구성의 이점이 모델 양자화 효과와 독립적으로 존재함이 확인된다. 분산 조건의 품질 평가 범위는 SQuAD v2.0 dev 전체 5,928개, COCO val2017 전체 5,000장, LibriSpeech test-clean 전체 2,620개로 확장하였으며, 옴니모달 3종은 추론 시간

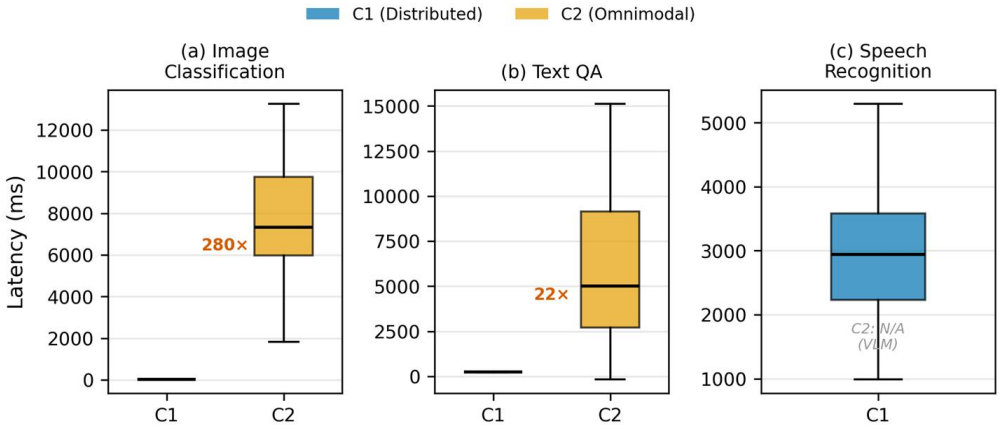

Fig. 2. End-to-end latency distributions for individual tasks under Distributed and Omnimodal conditions. (a) Image classification (Distributed: 26ms vs Omnimodal: 7,532ms), (b) Text QA (Distributed: 232ms vs Omnimodal: 5,806ms), (c) Speech recognition (Distributed only). Boxplots are reconstructed from summary statistics (mean, SD, n).

Table 4. Per-Task Latency – 4-way Comparison (P50 / Mean±SD ms)

Task	C1 Distributed	C2-FP16 LLaVA-7B	C2-INT4 LLaVA-7B Q4	C2-Light Qwen2-VL-2B
Vision (n)	26.15 / 26.15 ±0.38 (50)	8,232 / 7,531.64 ±3,068.56 (20)	2,096 / 2,159.22 ±337.21 (30)	4,704 / 5,448.79 ±2,156.84 (30)
Text (n)	231.52 / 232.28 ±1.92 (50)	3,176 / 5,805.79 ±4,556.67 (20)	878 / 1,332.35 ±1,063.81 (30)	846 / 864.13 ±203.96 (30)
Audio (n)	2,767.65 / 2,950.36 ±1,044.70 (50)	N/A (VLM unsupported)	N/A	N/A
Vision Runtime	TensorRT FP16	PyTorch	llama.cpp	PyTorch
Text Runtime	ONNX RT INT8	PyTorch	llama.cpp	PyTorch

Table 5. Statistical Significance and Quality Metrics – 4-way Comparison

Task / Pair	Latency P50 ratio (C1 advantage)	Welch's t / Cohen's d	p-value	Quality (n / metric)
Vision C1 vs C2-FP16	288×	-10.94 / d=4.63	1.22×10^{-9}	C1: Conf 0.487±0.262 (n=5,000), MobileViT Top-1 78.3% (ImageNet[5])
Vision C1 vs C2-INT4	80×	-34.65 / d=8.97	$<1\times10^{-12}$	(VQA free-form text, direct accuracy comparison not feasible)
Vision C1 vs C2-Light	180×	-13.81 / d=3.55	5.4×10^{-14}	
Text C1 vs C2-FP16	25×	-5.47 / d=2.31	2.81×10^{-5}	F1 / EM (SQuAD v2.0 dev): C1 0.621/0.530 (n=5,928), C2-FP16: 0.623/0.536, C2-INT4: 0.595/0.484, C2-Light: 0.620/0.504 (each n=1,000)
Text C1 vs C2-INT4	3.8×	-5.66 / d=1.46	4.3×10^{-6}	Δ F1 $\leq$ 2.8 pp (all 4 within 0.59-0.62, comparable)
Text C1 vs C2-Light	3.7×	-16.92 / d=4.39	$<1\times10^{-12}$	
Audio C1 only (Whisper-tiny)	–	–	–	WER 15.65%±19.14%, CER 10.33%±15.73% (LibriSpeech test-clean, n=2,620)

제약으로 각 1,000개에 한정하였다. 분산 조건의 DistilBERT INT8(66M)은 LLaVA-7B FP16/Q4 및 Qwen2-VL-2B와 사실상 동등한 F1 품질(0.62)을 달성하면서 Text에서 최소 3.7배 빠른 응답을 제공한다. Whisper-tiny의 WER 15.65%는 원논문에서 보고된 5.6% 대비 경량 모델과 CPU 추론에 따른 손실이며, Vision은 출력 형식 차이(분류 vs. VQA)로 인해 직접 정확도 비교가 불가능하다.

2. Concurrent Serving and Normalized Metrics

Table 6은 개별-동시 서빙 간 지연시간 변화를 나타낸다.

Table 6. Individual vs Concurrent Latency (Distributed, n=50, Wilcoxon signed-rank test)

Metric	Vision	Text	Audio
Individual (ms)	26.15 ±0.38	232.28 ±1.92	2,950.36 ±1,044.70
Concurrent (ms)	29.08 ±0.80	315.93 ±40.88	2,951.48 ±1,086.97
Chage	+11.2%	+36.0%	+0.04%
Wilcoxon p-value	<0.001	<0.001	0.660
Cohen's d (paired)	3.66	2.06	0.006

Table 7은 비용 정규화 처리량을 나타낸다.

Table 7. Cost-Normalized Throughput (Throughput/$1K)

Task	Basic	Dist.	Omni.	Dist./Omni.
Vision	Per-Device	54.6	0.19	288x
Text	Per-Device	23.9	0.25	97x
Vision	Total cluster	37.1	0.19	195x
Text	Total cluster	4.2	0.25	17x

Table 8에서 INT4 양자화 베이스라인은 FP16 옴니모달 대비 Vision 에너지를 4.67배, Text 에너지를 3.37배 절감하였다. INT4 양자화 베이스라인 대비 분산 구성은 Vision에서 추가로 172배 효율적이다.

Accuracy/W에서 Text는 분산 조건(5.39)과 옴니모달 조건(5.62)이 동등하나(0.96배), Accuracy/J에서는 분산 조건(22.25)이 옴니모달 조건(1.43) 대비 15.6배 높다. 이는 전력뿐 아니라 처리 시간을 함께 고려해야 함을 의미한다.

동시 서빙 시 Vision은 11.2%, Text는 36.0% 지연시간이 증가하였으나 Audio는 사실상 변화가 없었다(0.04%, Wilcoxon p=0.660). Vision과 Text의 증가는 오케스트레이터의 gRPC 동시 연결 관리 부하에 기인하며, Audio는 추론 시간(~2.2초)이 통신 오버헤드를 압도하기 때문이다. 분산 조건의 wall-clock P50은 옴니모달 조건의 순차 추정치 대비 4.9배 빠르다.

에너지 효율에서 분산 조건 Vision은 0.135J/req로 옴니모달 조건(108.8J) 대비 806배 효율적이다. AGX Orin 전력은 분산 조건 MobileViT-S 추론 시 5,522mW(유휴 대비 +692mW)인 반면, 옴니모달 조건 LLaVA-7B는 12,056mW(+7,226mW)였다. RPi5는 Whisper 추론 시 평균 9.6W(유휴 4.0W), OPi5는 DistilBERT 추론 시 평균 12.3W(유휴 4.1W)를 기록하였다. 동시 서빙 클러스터 전체는 13.15J/round로 옴니모달 조건(156.0J) 대비 11.9배 효율적이다.

비용 정규화(Throughput/$1K)에서 분산 조건은 per-device 기준 97~288배, 총 클러스터 비용 기준 17~195배 높은 효율을 보여, 47% 높은 하드웨어 비용에도 분산 조건이 비용 효율적이다.

3. gRPC Communication Overhead

Table 9는 gRPC 통신 오버헤드를 나타낸다. Vision과 Text의 gRPC 오버헤드는 E2E 지연시간의 10% 이내(1.56~6.32ms)로 제한적이다. 동시 서빙 시 각각 1.7배, 1.6배 증가하였으나 절대값이 작아 실용적 영향은 미미하다. Audio는 749ms(24.7%)로 높은 오버헤드를 보이며, 이는 float32 raw waveform(~160KB)의 직렬화/역직렬화 비용에 기인한다. 압축 전송이나 스트리밍 gRPC를 통해 개선 가능하다.

Table 8. Energy Efficiency

Metric	C1 Distributed	C2-FP16	C2-INT4	C2-Light	C1/C2-INT4
Vision Energy/Req (J)	0.135	108.8	23.3	46.6	172×
Text Energy/Req (J)	2.98	47.2	14.0	6.8	4.7×
Audio Energy/Req (J)	10.03	–	–	–	–
Avg Power (Vision, W)	5.52	12.06	10.78	8.55	–
Δ Power vs idle (W)	+0.69	+7.23	+6.56	+4.33	–
GPU Memory (Vision, MB)	~2,000	13,931	~3,800	4,215	–
Concurrent Energy/Round (J)	13.15	156.0	–	–	–

Table 9. gRPC Communication Overhead. Individual vs Concurrent Serving

Task	Mode	E2E P50 (ms)	gRPC OH (ms)	OH (%)
Vision	Individual	26.15	1.56 ±0.25	5.9
Vsion	Concurrent	29.08	2.72	9.3
Text	Individual	232.28	3.86 ±0.21	1.7
Text	Concurrent	315.93	6.32	2.0
Audio	Individual	2,950.36	749.04 ±326.87	24.7
Audio	Concurrent	2,951.48	746.67	24.0

4. Network Environment Sensitivity Analysis

Table 10은 분산 추론의 네트워크 환경 의존성을 평가하기 위해 5종 네트워크 시나리오에 측정한 성능을 나타낸다.

Table 10. Network Sensitivity – P50 Latency (n=20 per scenario)

Scenario	RTT / BW / Loss	Vision (ms)	Text (ms)	Audio (ms)
Wired (baseline)	0.5ms / 1Gbps / 0%	31.7	234.0	4,745.6
Wi-Fi 6	5ms / 600Mbps / 0.1%	44.6	238.1	4,797.4
Wi-Fi 5 office	15ms / 200Mbps / 0.5%	68.4	248.6	6,853.3
5G NR	30ms / 100Mbps / 1%	89.6	263.6	11,179.1
Worst	100ms / 10Mbps / 5%	162.1	357.2	31,512.0

Audio는 RPi5의 eth0 인터페이스에 Linux tc-netem을 적용한 실측이며, Vision과 Text는 Python 측 지연 삽입(RTT + payload/bandwidth)으로 시뮬레이션하였다. S2 Wi-Fi 6 환경은 baseline 대비 1% 이내 변화로 사실상 동등하며, S3와 S4에서도 Vision은 최대 2.8배, Audio는 2.4배 증가에 그친다. S5 최악 조건에서 Audio가 6.6배 증가하는 것은 raw waveform(~160KB) 대역폭 제약과 5% 패킷 손실에 따른 TCP 재전송이 결합한 결과이다. 그럼에도 분산 구성은 옴니모달 대비 우위를 유지한다. S5에서 Vision 162ms는 INT4 양자화 베이스라인(2,096ms) 대비 13배, Text 357ms는 INT4 양자화 베이스라인(878ms) 대비 2.5배 빠르다. Audio의 무선 민감도는 Opus 인코딩(~10KB로 16배 압축)이나 스트리밍 gRPC로 7~8초 수준 개선이 가능하다. 무선 burst loss와 핸드오버 영향은 향후 연구 과제이다.

5. NPU Compatibility and Runtime Stability

Hailo-8 NPU는 2D CNN 전용으로 Whisper(1D conv)와 DistilBERT(Transformer)를 지원하지 못하여 CPU fallback이 필요하였다. 런타임별 추론 시간 변동 계수(coefficient of variation, CV)는 OPi5의 ONNX Runtime INT8(CPU)이 0.8%, AGX Orin의 TensorRT FP16(GPU)이 1.4%, RPi5의 PyTorch FP32(CPU)가 35.4%로, 양자화된 정적 그래프 런타임이 가장 안정적이며 동적 메모리 할당과 GIL 영향을 받는 PyTorch가 가장 큰 변동을 보였다.

VI. Discussion

1. Key Findings and Comparison with Prior Work

관찰된 분산 구성의 우위는 특화 경량 모델과 범용 대형 모델의 파라미터 규모 차이, 양자화 부재에 따른 메모리 대역폭 병목, 모달리티별 디바이스 병렬 처리의 복합 효과이다. 본 절에서는 INT4 양자화 베이스라인과 경량 베이스라인을 추가 측정하여 이 세 효과를 분리 분석한다.

1.1 Decoupling Quantization and Model Size Effects

동일 LLaVA-1.5-7B 모델을 FP16에서 Q4_K_M(INT4)로 양자화 시 Vision P50은 2,096ms(3.93배), Text P50은 878ms(3.62배)로 단축되며 Vision 에너지는 23.3J(4.67배) 감소로 함께 개선되며, 메모리 대역폭과 KV 캐시 크기에 직접 영향을 받는 정밀도 효과이다. 동일 PyTorch+transformers 런타임에서 모델을 7B에서 2B로 축소한 경량 베이스라인(Qwen2-VL-2B FP16)은 Vision P50 4,704ms(1.75배), Text P50 846ms(3.76배)로 단축되어, 동일 런타임 가정 하에서 모델 크기로 설명되는 가속 한도가 약 1.75-3.76배임이 확인된다.

1.2 Net Effect of the Distributed Architecture After Controlling for Quantization and Model Size

양자화와 모델 크기 효과를 통제한 뒤에도 분산 구성은 명확한 추가 우위를 유지한다. INT4 양자화 베이스라인 대비 Vision은 33ms(63.5배), Text는 231ms(3.8배) 빠르며, 동일 런타임의 경량 베이스라인 대비 Vision은 142배, Text는 3.7배 빠르다. 통계적으로 분산 구성과 INT4 양자화 베이스라인 비교의 Cohen's d는 Vision 8.97, Text 1.46로 큰 효과 크기를 보였다. 이 차이는 모달리티별 가

속기 특화(GPU TensorRT / CPU INT8 ONNX / CPU PyTorch)에 따른 단위 연산 효율과 모달리티 단위 병렬 처리에 따른 wall-clock 단축이 결합된 구조적 효과이며, 모델 크기·양자화로 환원되지 않는다. 더 나아가 지연시간 격차(316배)와 평균 전력 격차(5.52W vs 12.06W, 2.18배)가 곱으로 결합되면서 Vision 에너지/요청 격차는 806배로 비선형적으로 확대된다. 분산 통신 비용은 작다. MobileViT-S를 AGX Orin에서 gRPC 없이 단독 실행한 P50은 23.78ms(SD=0.32)로 gRPC 경유(26.15ms) 대비 2.37ms(9.1%)의 오버헤드만 추가되며 오케스트레이터 부하는 0.5-1.2ms/req에 그친다.

1.3 Quality Equivalence

전체 데이터셋 기반 평가에서 Text QA F1은 4종 모두 0.59~0.62의 좁은 범위($\Delta \leq 2.8\%$p)에 분포하며 세부 수치는 Table 5에 제시하였다. Vision은 분산 조건 COCO val2017 5,000장에서 Mean Confidence 0.487±0.262(631종 클래스), Audio는 LibriSpeech test-clean 2,620개에서 WER 15.65%±19.14%로 측정되었다. 즉, 단일 모달리티 태스크에서 범용 대형 모델(7B LLaVA)의 추가 파라미터가 66M DistilBERT 또는 2B Qwen2-VL 대비 품질 향상으로 이어지지 않음을 정량적으로 확인하였으며, 모달리티 단위 독립 배치로 모델 간 데이터 의존성을 제거하여 완전 병렬 처리를 달성한 점에서 Neurosurgeon[9]의 DNN 파티셔닝과 차별된다.

2. Limitations

본 연구의 한계점은 다음과 같다.

첫째, 평가 범위 측면에서 분산 조건은 COCO val2017 전체 5,000장, SQuAD v2.0 dev 전체 5,928개, LibriSpeech test-clean 전체 2,620개로 확장 평가하였으나 옴니모달 3종은 추론 시간 제약(FP16 옴니모달은 5,000장 처리 시 약 11시간)으로 1,000샘플에 한정되었다. 다만 모든 비교의 Cohen's $d > 1.4$, $p < 1\times10^{-5}$로 유의성은 확보되었다.

둘째, 네트워크 환경과 확장성 측면에서 5종 네트워크 시나리오를 RPi5의 tc-netem(Audio)과 Python 측 지연 삽입(Vision/Text)으로 측정하였으나(Table 10), 무선 환경의 burst loss와 핸드오버, 5-10대 클러스터 확장 시 단일 오케스트레이터 병목, batch > 1 처리량은 본 연구 범위를 벗어난다.

셋째, 옴니모달 베이스라인 선정과 전력 측정 정확성에 한계가 있다. Audio를 지원하는 옴니모달(Qwen2-Audio 등)은 Jetson 안정 배포 경로 부재로 제외되었고, TensorRT-LLM AWQ INT4도 일정상 미적용이다. RPi5와 OPi5 전력은 USB 전력계로 측정하여 OS와 네트워크 프로세스 전력이 포함되므로 idle 차감과 3회 반복으로 영향을 최소화하였으나, 향후 INA226 션트 분리 측정이 필요하다.

넷째, YOLO11-nano의 HEF 변환이 미완료된 상태이므로 객체 탐지 태스크와 NPU+CPU 동시 서빙은 벤치마크에 포함되지 못하였으며, 본 연구의 Vision 비교는 분류(MobileViT-S)에 한정된다.

3. Practical Implications

동시 멀티모달 서빙 환경(스마트 홈, 산업 모니터링)에서는 이기종 클러스터 분산 배치가 처리량·에너지 모든 측면에서 유리하다. 단일 모달리티나 교차 모달 추론(cross-modal reasoning)이 필요한 환경에서는 옴니모달 배치가 시스템 복잡성 대비 효율적이다. 본 연구의 효과 분리 분석은 배치 전략 선택의 정량적 근거를 제공한다. 단일 디바이스 환경에서는 양자화(INT4 등)·경량 모델 채택만으로도 옴니모달 추론을 5~13배 가속할 수 있으나, 다중 모달리티 동시 서빙·실시간 응답이 요구되는 환경에서는 모달리티 단위 디바이스 병렬화의 추가 이득(63.5배)이 결정적이므로 분산 배치가 우선되어야 한다. NPU의 모델 호환성을 사전 확인하고 CPU fallback 전략을 수립하는 것이 이기종 가속기 통합의 핵심 설계 요소이다. Audio 등 대용량 데이터 전송 시 gRPC 오버헤드(24.7%)가 상당하므로 데이터 압축 또는 스트리밍 최적화가 필요하다.

VII. Conclusions

본 연구에서는 이기종 엣지 클러스터(AGX Orin + RPi5+Hailo-8 + OPi5 Plus)에서 3종 가속기와 4종 런타임을 gRPC로 통합한 분산 추론의 성능을 실측하고, LLaVA-1.5-7B FP16, 동일 모델의 INT4 양자화 버전(llama.cpp Q4_K_M), 경량 비전-언어 모델(Qwen2-VL-2B FP16)의 3종 옴니모달 베이스라인과 비교하였다.

본 연구의 핵심 결과는 분산 구성의 우위를 양자화, 모델 크기, 분산 구조의 세 효과로 단계적으로 분리한 것이다. 동일 LLaVA-1.5-7B 모델을 FP16에서 INT4로 양자화할 때 Vision 3.93배, Text 3.62배가 가속되며(양자화 효과), 동일 PyTorch+transformers 런타임에서 7B에서 2B로

모델 크기를 축소할 때 Vision 1.75배, Text 3.76배가 가속된다(모델 크기 효과). 양자화와 모델 크기 효과를 통제한 뒤에도 INT4 양자화 베이스라인 대비 분산 구성은 Vision 63.5배, Text 3.8배 추가로 빠르다(분산 구조의 순수 효과). 이는 분산 구조의 이점이 양자화, 모델 크기 효과와 독립적이며 세 효과가 곱셈적(multiplicative)으로 누적됨을 의미한다. Vision의 단계별 우위(양자화 3.93배 × 모델 크기 1.75배 × 분산 구조 63.5배)가 결합되어 분산 구성과 FP16 옴니모달 사이 288배의 총합 격차가 형성된다.

분산 구성은 옴니모달 3종 모두 대비 Vision 80-288배, Text 3.7-25배 낮은 지연시간을 달성하였고, Vision 에너지 효율은 INT4 양자화 베이스라인 대비 172배, FP16 옴니모달 대비 806배 우수하였다. 동시 서빙 시 클러스터는 라운드당 13.15J로 옴니모달(156.0J) 대비 11.9배 효율적이었으며, 비용 정규화 처리량은 47% 높은 하드웨어 비용에도 17~195배 우수하였다. 전체 데이터셋 평가(SQuAD 5,928개·COCO 5,000장·LibriSpeech 2,620개) 및 옴니모달 3종 1,000샘플 평가에서 4종 모두 Text QA F1 0.59-0.62의 좁은 범위로 품질 동등성이 확인되었다($\Delta \leq$ 2.8%p). 5종 네트워크 시나리오(Wired/Wi-Fi 6/Wi-Fi 5/5G NR/최악) 민감도 분석 결과, 분산 구성은 무선 환경에서도 옴니모달 대비 우위를 유지하였다(최악 조건에서도 Vision 162ms·Text 357ms로 INT4 양자화 베이스라인의 2,096ms·878ms보다 우수).

후속 연구로는 (1) YOLO11-nano HEF 변환을 통한 NPU+CPU 동시 서빙 검증, (2) 교차 모달 파이프라인(cross-modal pipeline) 구현, (3) 5-10대 클러스터 스케일링 및 무선 네트워크(Wi-Fi 6, 5G NR) 환경 평가, (4) 오디오 지원 옴니모달(Qwen2-Audio 등) 및 TensorRT-LLM AWQ INT4와의 4-way 확장 비교, (5) 분산 오케스트레이션을 통한 단일 오케스트레이터 병목 해소를 계획한다.

REFERENCES

[1] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge Intelligence: Paving the Last Mile of Artificial Intelligence with Edge Computing,” Proc. IEEE, vol. 107, no. 8, pp. 1738-1762, Aug. 2019. DOI: 10.1109/JPROC.2019.2918951

[2] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual Instruction Tuning,” inProc. NeurIPS, 2023. DOI: 10.48550/arXiv.2304.08485

[3] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, et al., “Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution,” arXiv preprint arXiv:2409.12191, 2024. DOI: 10.48550/arXiv.2409.12191

[4] Y. Zheng, Y. Chen, B. Qian, X. Shi, Y. Shu, and J. Chen, “A Review on Edge Large Language Models: Design, Execution, and Applications,” arXiv preprint arXiv:2410.11845, 2024. DOI: 10.48550/arXiv.2410.11845

[5] S. Mehta and M. Rastegari, “MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer,” inProc. ICLR, 2022. DOI: 10.48550/arXiv.2110.02178

[6] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision,” inProc. ICML, 2023. DOI: 10.48550/arXiv.2212.04356

[7] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter,” inProc. NeurIPS Workshop on Energy Efficient Machine Learning and Cognitive Computing, 2019. DOI: 10.48550/arXiv.1910.01108

[8] Z. Zhao, K. M. Barijough, and A. Gerstlauer, “DeepThings: Distributed Adaptive Deep Learning Inference on Resource-Constrained IoT Edge Clusters,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 37, no. 11, pp. 2348-2359, Nov. 2018. DOI: 10.1109/TCAD.2018.2858384

[9] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, “Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge,” inProc. ACM ASPLOS, Xi'an, China, Apr. 2017, pp. 615-629. DOI: 10.1145/3037697.3037698

[10] Y. Gao, Z. Zhang, P. K. Donta, C. K. Dehury, X. Wang, D. Niyato, and Q. Zhang, “Optimizing Multi-DNN Inference on Mobile Devices through Heterogeneous Processor Co-Execution,” arXiv preprint arXiv:2503.21109, 2025. DOI: 10.48550/arXiv.2503.21109

[11] G. Xu, Z. Hao, L. Shen, Y. Luo, F. Sun, X. Wang, H. Hu, and Y. Wen, “CoFormer: Collaborating with Heterogeneous Edge Devices for Scalable Transformer Inference,” arXiv preprint arXiv:2508.20375, 2025. DOI: 10.48550/arXiv.2508.20375

[12] J. Yoon, J. Lee, T. He, N. Choi, and B. Ji, “S2M3: Split-and-Share Multi-Modal Models for Distributed Multi-Task Inference on the Edge,” arXiv preprint arXiv:2508.04271, 2025. DOI: 10.48550/arXiv.2508.04271

[13] Hailo Technologies, "Hailo-8 Operator Support and Model Compatibility Guide,"Hailo Developer Zone, https://hailo.ai/developer-zone/, 2024.

[14] R. Jayanth, N. Gupta, and V. Prasanna, “Benchmarking Edge AI Platforms for High-Performance ML Inference,” arXiv preprint arXiv:2409.14803, 2024. DOI: 10.48550/arXiv.2409.14803

[15] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C.-J. Wu, et al., “MLPerf Inference Benchmark,” inProc.

ACM/IEEE ISCA, Valencia, Spain, May 2020, pp. 446-459. DOI: 10.1109/ISCA45697.2020.00045

[16] G. Gerganov et al., “llama.cpp: Port of LLaMA in C/C++ with GGUF Quantization,” GitHub repository, https://github.com/ggml-org/llama.cpp, 2024.

[17] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, et al., “AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration,” inProc. MLSys, Santa Clara, CA, USA, May 2024. DOI: 10.48550/arXiv.2306.00978

[18] Y. Chu, J. Xu, Q. Yang, H. Wei, X. Wei, Z. Guo, et al., “Qwen2-Audio Technical Report,” arXiv preprint arXiv:2407.10759, 2024. DOI: 10.48550/arXiv.2407.10759

[19] Y. Li, S. Zhang, Y. Zeng, H. Zhang, X. Xiong, J. Liu, P. Hu, and S. Banerjee, “Tiny but Mighty: A Software-Hardware Co-Design Approach for Efficient Multimodal Inference on Battery-Powered Small Devices,” arXiv preprint arXiv:2510.05109, 2025. DOI: 10.48550/arXiv.2510.05109

[20] J. Wolfrath, A. Achanta, and A. Chandra, “Leveraging Multi-Modal Data for Efficient Edge Inference Serving,” inProc. IEEE/ACM CCGrid, Philadelphia, PA, USA, May 2024, pp. 408-417. DOI: 10.1109/CCGRID59990.2024.00053

Authors

Pil-Seong Jeong received the B.S. degree in Electronic Engineering from Seoul National University of Science and Technology, Korea, in 2004, and the M.S. and Ph.D. degrees in Electronic and Communication Engineering from Kwangwoon University, Korea, in 2008 and 2013, respectively. Prof. Jeong is currently a Professor with the Department of Information and Communication Engineering, Myongji College, Seoul, Korea. His research interests include edge AI, AIoT, embedded systems, and real-time object detection on heterogeneous edge accelerators.