AI Agent 기반 로보틱스와 저지연 추론시스템을 위한 KV 캐시 최적화 및 하드웨어 가속 기술 동향

Trends in KV Cache Optimization and Hardware Acceleration for AI Agent-Based Robotics and Low-Latency Inference

이영민 (Y.M. Lee, yeongmin.lee@etri.re.kr) 초거대AI반도체연구실 선임연구원
김혜지 (H.J. Kim, hyejikim@etri.re.kr) 초거대AI반도체연구실 선임연구원
여준기 (C.G. Lyuh, cglyuh@etri.re.kr) 초거대AI반도체연구실 책임연구원/실장

ABSTRACT

Advancements in artificial intelligence (AI) have led to the emergence of AI agents built on large-scale multimodal models. These agents extend beyond simple question answering to perceive complex contexts, perform step-by-step reasoning, utilize external tools, and self-evaluate on their outputs, thereby enabling autonomous and adaptive intelligence. The rise of physical AI has further enabled embodied human–machine interaction, integrating vision, language, and action to usher in a new era of embodied AI agents. However, deploying such advanced agents in real-time environments demands low-latency, high-efficiency optimization to handle substantial computational and memory loads. To describe the corresponding trends, we review the foundational technologies of physical AI, highlights Google DeepMind's Gemini Robotics, and explores KV cache optimization and specialized hardware acceleration as effective approaches for alleviating memory bottlenecks in real-time AI agent systems.

KEYWORDS AI Agent, Physical AI, AI 가속기, KV 캐시 경량화, 메모리 최적화

I. 서론

최근 인공지능의 발전은 대규모 멀티모달 모델을 기반으로 한 AI 에이전트(AI Agent) 개념의 확산을 가속하고 있다[1-4]. AI 에이전트는 단순 질의응답을 넘어, 복잡한 상황을 인식하고 문제를 단계적으로 사고하며, 외부 도구를 활용하여 스스로 답변과 행동을 재검토하는 자율적 지능 시스템으로 진화하고 있다. AI 시스템의 고도화는 이미 우리 일상 속에 깊숙이 스며들었다. 여행 계획, 운동 코칭, 요리 레시피, 나아가 간단한 응급처치와 육아 조언, 심지어 심리상담까지 대화를 통한 다양한 지식 전달과 조

* DOI: https://doi.org/10.22648/ETRI.2025.J.400503
* 공동 제1저자 이영민, 김혜지 (공동 제1저자 김혜지 추가, 2025. 12. 1. 수정)
* 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 정보통신 방송연구개발사업의 일환으로 하였음[RS-2024-00399817, 인공지능 기반 실감형 3D 렌더링 및 모델링 가속 AI반도체 개발].

력자 역할을 수행하고 있다. 나아가 최근 주목받고 있는 Physical AI의 등장으로 인간과 기계 간 행동 기반 상호작용이 가능해졌다. 이러한 기술적 진보는 AI를 온라인 환경에 국한된 존재에서 해방시켰고, 로보틱스 분야로 확장시켜 시각-언어-행동을 통합한 Embodied AI Agent의 새로운 시대를 열고 있다.

고도의 AI Agent를 실시간 시스템에 적용하기 위해서는 방대한 연산량과 메모리 요구량을 줄이기 위한 저지연-고효율 최적화 기술이 필수적이다. LLM(Large Language Model)이 Agent화되면서, 장기적 맥락을 유지하고, 과거의 경험과 현재 상태 바탕으로 의사결정을 내리는 기능의 중요성이 커졌고, 이에 따라 대용량 계층적 메모리 구조의 필요성이 대두되고 있다. 이는 과거에 전통적인 메모리 병목 지점이던 LLM 모델 파라미터의 용량을 넘어서는 수준이며, 이제는 모델 연산에 필요한 KV 캐시가 실시간 응답 시스템의 직접적인 병목 요소로 주목받고 있다[5-7].

본고에서는 Physical AI를 가능하게 한 AI Agent의 배경 기술과 Google DeepMind의 Gemini Robotics 사례를 통해 현재 AI 로보틱스 분야의 기술 발전 수준을 간략히 살펴본다. 이어, 실시간 저지연 응답 시스템에서 발생하는 메모리 병목 문제를 해결하기 위한 연구 분야 중의 하나인 KV 캐시 최적화 방법론과, 이에 특화된 하드웨어 가속기 연구를 소개한다.

II. AI Agent 기반 로보틱스 기술

AI Agent 기반 로보틱스는 지능적 추론, 상황 인식, 계획 수립, 행동 실행의 전 과정을 통합하는 차세대 로봇 지능 구조를 지향한다. 전통적인 로봇제어 시스템이 사전 정의된 규칙과 센서 입력에만 의존했다면, AI Agent 로봇은 복잡한 절차를 단계적으로 계획하고, 환경과 상호작용을 하며, 스스로의 판단을 재검토해 정밀성과 안전성을 높인다. 이 장에서는 전통적으로 텍스트 생성에 특화됐던 LLM을 목표 지향 및 적응형 지능 시스템으로의 확장을 이루게 된 배경 기술을 간략히 소개하고, Embodied AI Agent를 활용하여 범용형 AI 로보틱스를 구현한 최신 기술을 소개한다.

1. AI Agent 배경 기술

1.1 Chain-of-Thought

Chain-of-Thought(CoT)[7]는 LLM이 생각의 흐름을 스스로 생성하여 복잡한 추론 과정을 단계별로 풀어서 처리하도록 유도하는 프롬프트 기법이다. LLM은 정답을 바로 내놓지 않고 생각의 사슬(CoT)을 따라 중간 Reasoning 단계를 명시하도록 유도하여 복잡한 연산 및 논리적 추론 성능을 크게 향상시켰다.

1.2 ReAct

ReAct[8]는 모델이 생각(Reasoning)뿐만 아니라 단순한 텍스트 응답을 넘어 툴을 사용하거나 API를 호출하는 행동(Acting)을 교차적으로 수행하여 실제 정보에 기반한 정답을 도출하는 프레임워크 기술이다. 모델이 생각하고 행동을 실행한 뒤 그 결과를 관찰한 후 다시 생각을 이어가도록 외부 환경과 상호작용 하도록 구성되어, CoT의 추론력과 Agent형 시스템의 행동력을 결합한 Agent형 LLM을 가능하게 했다.

1.3 Reflexion

Reflexion[9]은 모델이 자신의 답변이나 추론 과정을 스스로 검토하여 피드백을 제공하여 더 나은 답변을 생성하기 위해 미래 행동을 수정하도록 유도하는 기법이다. 모델은 최초의 답변을 한 후 자

신의 답이 맞는지 혹은 다른 접근이 필요한지를 스스로 평가하고, 필요하면 다시 답을 도출하는 자기 피드백(Self-Feedback) 메커니즘을 활용함으로써 모델이 과거의 행동을 반성(Reflect)하고 개선하는 AI Agent 시스템의 기반이 되었다.

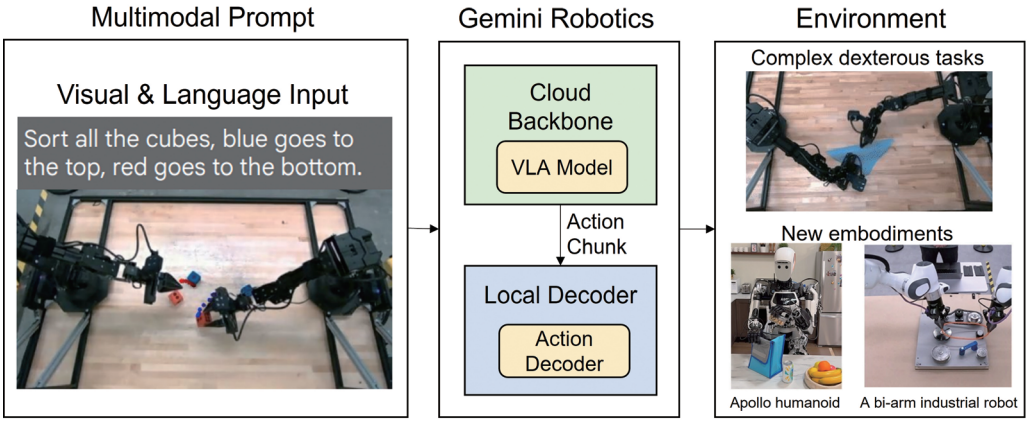
2. Embodied AI Agent를 이용한 로봇제어 모델: Gemini Robotics

Gemini Robotics[4]는 Google의 DeepMind에서 개발한 Vision-Language-Action(VLA) 모델로, Vision-Language Model(VLM) 기반의 Gemini 2.0[10]을 미세조정하여 물리적 로봇 제어 분야로 확장한 로보틱스 전용 모델이다. 이 모델은 단순한 로봇 제어 기능을 넘어, 외부 환경과 로봇 센서 정보를 기반으로 내부 상태를 갱신하고 계획을 수정하기 위해 도구 및 API를 활용하는 멀티스텝 행동 계획을 수행할 수 있다. 특히, Chain-of-Thought(CoT) 및 ReAct 개념을 결합한 AI Agent형 Embodied Reasoning 모델(Gemini Robotics-ER)을 통해 로봇을 직접 제어하는 것이 가능함을 보여주었으며, 일상 언어를 물리적 행동으로 연결하는 End-to-End 모델(Gemini Robotics)로 확장하였다. 이를 통해 복잡한 물체를 조작하고 자연어 지시를 이해하며, 돌발 상황에 대응하는 등 다양하고 복잡한 과제를 수행할 수 있다. 또한, 모델 재학습 없이 새로운 환경과 플랫폼에 적용 가능하며, 소량의 시연만으로 새로운 작업을 학습하고 수행한다.

2.1 클라우드-로컬 연계형 구조

Gemini Robotics-ER과 같은 대형 VLM 모델은 온보드(On-Board) 환경에서 실행하는 경우 추론 속도가 느리고 추가적인 전용 하드웨어가 필요하므로 실시간 제어에 부적합하다. 이에 따라 Gemini Robotics는 클라우드와 로컬에 기능을 분리하는 구조를 채택하였다(그림 1).

클라우드에 위치한 백본(Backbone) 모델은 Gemini Robotics-ER을 지식 증류(Distillation)방식으로 압축한 버전으로, 기존 수 초에 달하던 응답 지연시간을 160ms 이하로 최적화하였다. 이 모델은 복잡한 언어 지시와 시각 정보를 해석하고 고차원 추론을 처리하며, Embodied Reasoning 기능을 통해 물체의 공



출처 Reproduced from S. Abeyruwan et al., "Gemini Robotics: Bringing AI into the Physical World," arXiv preprint, 2025. doi: 10.48550/arXiv.2503.20020

그림 1 Google DeepMind의 Gemini Robotics 전체 구조도

간상 위치 및 상호작용 가능성 등에 관한 물리적 추론을 수행한다. 또한, 로봇 제어에 필요한 고수준 행동 절차(Action Plan)를 생성하여 실행을 지원한다.

로컬에 위치한 디코더(Decoder)는 로봇의 온 보드 컴퓨터에서 실행되며, 백본에서 발생하는 지연시간을 보완하여 로봇의 부드러운 동작과 반응성 있는 행동을 가능하게 한다. 또한, 백본에서 전달된 행동 절차를 해석하여 로봇이 실행 가능한 저수준 제어 명령으로 변환한다.

백본과 디코더를 통합한 전체 지연시간은 약 250ms 이내로, 한 번에 다수의 동작을 묶어 처리함으로써 실질적 제어 주파수는 50Hz를 달성하였다.

2.2 Gemini Robotics의 성능평가 요소

로봇을 가정과 산업 분야에 대규모로 배치하기 위해선 강건한 일반화 능력이 중요하다. 모델은 작업 수행에 직접적인 영향을 미치지 않는 장면에서는 불변성을 유지하면서, 자연어 명령어 표현의 다양성 및 동등성을 이해해야 한다. 또한, 학습된 동작을 새로운 상황에 적응시키고, 여러 동작을 조합-합성할 수 있어야 한다. 본 연구는 이러한 일반화 성능을 표 1에 정리된 것과 같이 기본적인 조작 기능부터 Embodied Reasoning 기반의 고차원 기능, 그리고 일반화 성능과 고난이도 과제에 대한 적응 능력까지 여러 각도로 평가하여 범용형 AI 로보틱스의 실현이 가능함을 확인했다.

III. AI Agent를 위한 KV 캐시 압축 및 관리 SW-HW 기술

AI Agent를 활용한 실시간 서비스 시스템에서 LLM은 저지연 응답을 위하여 각 입력 토큰마다 모든 레이어에서 생성된 키(Key)와 값(Value)을 KV 캐시에 미리 저장하고 이후 토큰 생성 시 반복 연산을 피하는 데 활용한다. 그러나 대화가 길어지고 컨

표 1 Gemini Robotics의 작업 능력 및 평가 요소

평가 범주	세부 평가 항목	설명
기본 조작 및 범용성	단기 정밀 조작 작업	물체 집기, 옮기기, 포장 등 짧은 순차적 동작들의 과제 수행 능력
	자연어 지시 수행	다양하고 구체적이며 복잡도가 높은 명령어를 해석하고 실행하는 능력
Embodied Reasoning 기능	3D 인지	장면의 3차원 구조와 물체의 위치 및 형상을 인식하는 능력
	포인터 지시	언어 및 시각 정보의 결합을 통해 특정 물체나 위치를 지시하는 능력
	로봇 상태 추정	로봇의 자세, 위치, 관절각 등 기계적 현재 상태를 추론하는 능력
	Affordance 예측	잡기, 밀기 등 물체를 기반으로 하는 조작 작업을 코드로 예측하는 능력
일반화 성능	시각적 일반화	배경, 조명, 방해 물체, 질감의 변화에 강건하게 성능을 유지하는 능력
	명령어 일반화	표현 변화, 오탈자, 다른 언어, 구체성 수준의 변화에 대응하는 능력
	행동 일반화	초기에 없던 신규 물체의 등장, 조건이나 상황의 변화에 적응하는 능력
적응 능력	장기 정밀 조작 특화	종이접기, 옷 개기 등 다단계-고난도 조작 과제를 수행하는 능력
	의미 기반 추론 강화	의미를 이해하여 시공간을 추론하고 최적화하는 능력
	신규 작업 및 환경의 빠른 적응	최대 100개 이하의 데모를 통해 새로운 작업에 적응하는 능력(Few-Shot Learning)
	새로운 로봇 구현체 적응	휴머노이드, 양팔 로봇 등 전혀 다른 하드웨어 형태에 적용하여 제어할 수 있는 능력

출처 Reproduced from S. Abeyruwan et al., "Gemini Robotics: Bringing AI into the Physical World," arXiv preprint, 2025. doi: 10.48550/arXiv.2503.20020

텍스트가 확장될수록, KV 캐시는 입력 토큰과 생성 토큰 수에 비례해 선형적으로 증가하며, 이는 전통적인 병목 지점이던 모델파라미터의 메모리 사용량을 넘어섰다. 특히 AI Agent는 다중 추론, 도구 호출(Tool-Use) 및 외부 환경과의 상호작용 등으로 컨텍스트를 길게 유지하면서 세션 상태를 지속적으로 쌓아가야 하므로 KV 캐시 의존도가 높아지고, 메모리 사용량이 폭발적으로 증가한다. 그 결과 메모리 부족, 대역폭 병목, 연산 지연에 따른 응답성 저하 현상이 발생하게 된다. 이러한 문제를 해결하고자 KV 캐시 압축 및 효율적인 관리 기법에 관한 연구가 활발히 진행되고 있다.

본 장에서는 알고리즘 중심의 KV 캐시의 최적화 기법을 살펴본 뒤, 하드웨어 가속기를 접목한 통합 시스템 연구를 소개한다.

1. CAKE

CAKE(Cascading & Adaptive KV Cache Eviction with Layer Preferences)[11]는 KV 캐시의 최대 메모리 사용량을 제한 내로 유지하면서 효율적으로 삭제(Eviction)하기 위해 Attention 레이어의 동적 특성과 레이어 선호도를 계단식으로 분석하여 전체 레이어 관점에서 캐시 사용을 조절하는 점진적 관리 기술을 제안한다(그림 2).

기존의 캐시 제거방식은 전체 레이어에 균등한 비율을 적용했으나, 실제 Attention 패턴은 레이어마다 다르므로 각 레이어의 중요한 정보를 유지하면서 캐시 크기를 줄이는 적응형 최적화 방법을 적용했다. Attention의 중요도는 한 토큰과 다른 토큰 사이에 발생하는 공간적 Attention 분산 특성과 시간이 지남에 따라 영향력이 큰 토큰의 변화를 추적하는 시간적 Attention 이동 특성 두 가지를 이용하여 계산한다. 또한, 단계적 캐시 메모리 관리 방식을 제안하여 각 레이어의 프리필링(Prefilling)을 순차적으로 처리하였으며, 선호도 점수에 따라 캐시 예산을 동적으로 재분배하고 Attention의 시간적 변화에 강인한 제거 지표를 기반으로 캐시를 제거하고 업데이트한다. 이를 통해 전체 메모리 사용량을 일정하게 유지하면서도 비 단계적 방식과 동일한 캐시 분배 및 제거 결과를 도출했다.

해당 연구는 벤치마크[12,13]를 통한 성능평가에서 전체 KV 캐시의 3.2% 만으로 성능을 유지하면서 10배 이상의 캐시 디코딩 속도를 향상했다.

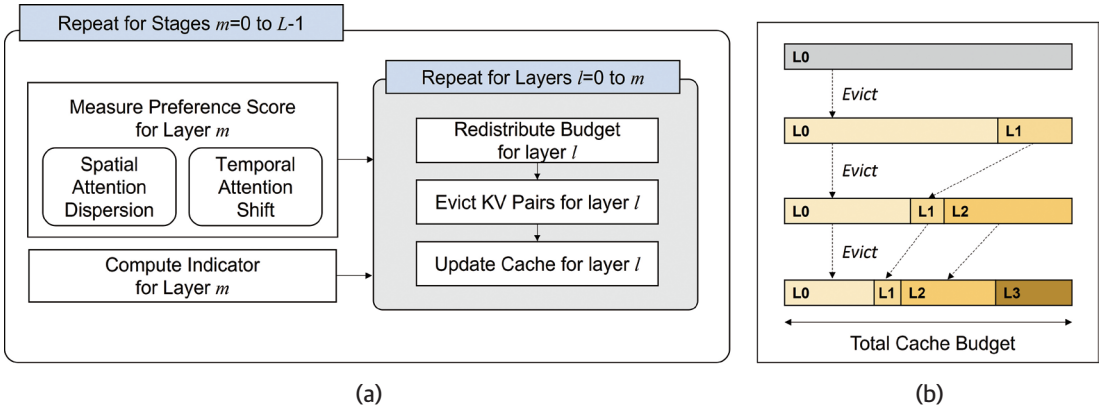


그림 2 (a) CAKE 알고리즘 구조 (b) 선호도 기반 동적 캐시 할당 예시

2. SQuat

SQuat(Subspace-orthogonal KV Cache Quantization)[14]는 쿼리(Query) 기반 정보를 키(Key) 양자화에 활용하여 양자화로 인한 Attention 결과의 오차를 최소화하면서 KV 캐시 사용량을 줄이는 기술을 제안한다(그림 3).

기존의 KV 캐시 양자화 기법은 키(Key) 양자화로 인하여 QK 내적으로 계산되는 Attention 스코어에 오류가 누적되는 현상이 있었다. 본 연구는 쿼리(Query) 텐서에 SVD(Singular Value Decomposition)를 적용하여 쿼리 데이터가 주로 존재하는 정보의 방향을 압축한 저차원 정보 공간(Subspace)을 구성하고 키(Key)의 양자화 오차 정보는 쿼리 공간과 서로 직교한다는 제약조건을 Attention 연산 과정에 추가한다. 이를 통해 키(K)의 양자화 오차 정보가 쿼리(Q)로 향하는 방향 성분은 계산 과정에서 재보정되므로 양자화에 의한 왜곡을 최소화할 수 있다. 최종적으로 키는 양자화되었지만, Attention이 필요로 하는 쿼리 방향의 정보는 그대로 유지하므로 성능 저하 없이 KV 캐시 메모리 사용량을 줄인다.

해당 연구는 모델의 재학습 및 오프라인 캘리브레이션 없이 실시간으로 양자화할 수 있어 추론 흐름에 자연스럽게 통합할 수 있다. 벤치마크[12,15]를 통한 성능평가에서 기존의 비보정(Tuning-Free) 양자화 기술 대비 최대 2.82배 메모리 사용량 감소 및 3.6배 처리량 증가를 달성하였다.

3. Oaken

Oaken[6]은 실시간으로 KV 벡터를 세 가지 데이터 그룹으로 분류하여 각 그룹에 적합한 양자화 기법과 데이터 포맷을 적용함으로써 메모리 사용을 최적화하는 방법을 제시한다. 이와 동시에 양자화 전용 가속기와 캐시 메모리 관리 엔진을 하드웨어로 설계하여 LLM 추론 서비스에 적용 가능한 저지연 고효율 KV 캐시 관리 솔루션을 제안한다.

3.1 Optimization Methods

본 연구의 양자화 알고리즘은 실시간 오프라인 연산의 지연시간을 최소화하면서 양자화에 의한 손실을 줄이고 압축률을 향상하도록 균형 있는 기법을 제안하고 있다(그림 4). 데이터 분포의 비균일성을 고려하여 값의 범위에 따라 세 그룹으로 분류하는 과정에서 동적인 임곗값을 사용하지 않고, 미리 정의된 임곗값을 사용하여 데이터 분류에 소모되

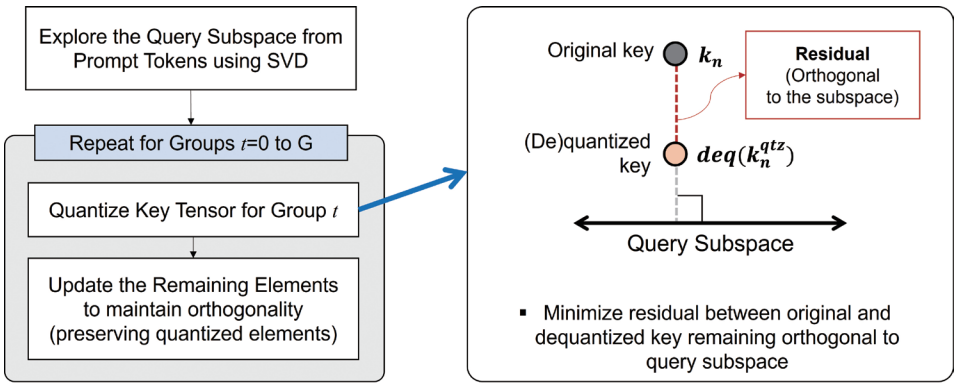


그림 3 SQuat 알고리즘 및 최적화 개념도

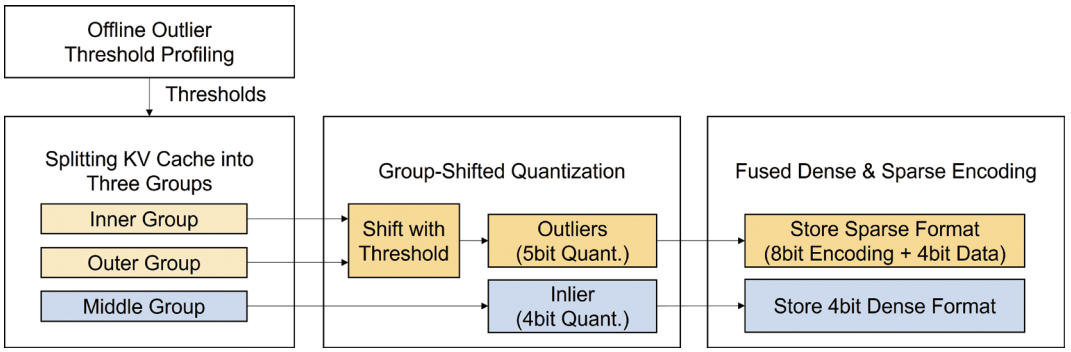


그림 4 Oaken의 KV 양자화 알고리즘 개념도

는 지연시간을 최소화하였다. 또한, 데이터 그룹에는 Uniform 양자화 기법을 적용하여 양자화 복잡도를 줄이는 동시에, 양자화 오차를 보상하고자 Inlier와 Outlier 그룹별 데이터의 분포 특성을 고려하여 간소화된 정규화 작업을 포함하였다. 전체 데이터의 90%에 해당하여 사용 빈도가 높은 Inlier 데이터는 4Bit 양자화에 밀집형(Dense) 포맷을 사용하여 접근 편의성을 높이고, 전체 데이터의 10%에 해당하는 Outlier 데이터는 5Bit 양자화에 희소형(Sparse) 포맷을 사용하여 압축률을 극대화하였다. 특히, 희소형 포맷은 인코딩용 8Bit와 데이터 4Bit로 분류하고 희소 데이터 4Bit는 밀집 데이터 저장공간에서 0이 저장된 곳을 재활용함으로써 압축률을 보다 개선했다.

3.2 HW Architecture

양자화가 적용된 실시간 토큰 생성 연산의 주요 병목 지점은 데이터의 양자화와 복원 연산, 그리고 양자화된 포맷 데이터의 전송 과정에서 발생한다. 본 연구는 양자화 엔진(Quntization Engine), 복원 엔진(Dequantization Engine), 그리고 메모리 관리 유닛(Memory Management Unit)을 DMA 파이프라인 내에 연계하여 실시간 저지연 시스템을 구축하였다(그림 5).

양자화 엔진은 KV 캐시를 실시간으로 압축하여 메모리 사용량을 줄이며, 데이터 분포를 기반으로 양자화 포맷을 결정하여 연산 정밀도를 유지한다. 복원 엔진은 압축된 KV 캐시를 스트리밍 방식으로 복원하며, 희소 데이터는 0을 삽입하여 원본 형태로 재구성한다. 메모리 관리 유닛은 KV 캐시의 접근을 페이지 기반으로 관리하여 희소-밀집형 데이터 포맷을 모두 지원하면서 버스트 접근을 통해 대역폭

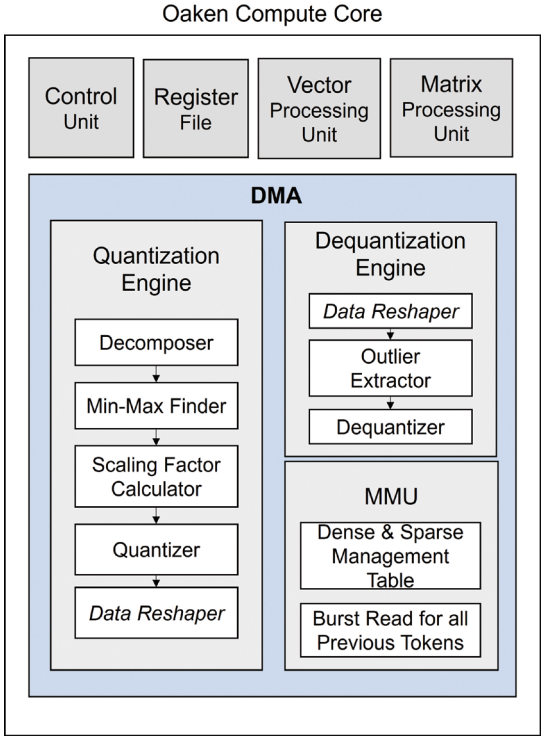


그림 5 Oaken의 Compute Core 구조도

활용도를 최적화한다. 이러한 구조는 KV 캐시의 관리 과정에서 발생하는 메모리 병목 현상을 해결하여 저지연 LLM 추론을 달성하였다.

해당 연구는 기존 4Bit 수준의 깊은 양자화를 적용해도 모델 정확도 손실은 0.54% 수준을 유지하였다. 또한, 토큰 처리량은 NVIDIA A100 GPU 대비 약 1.58배 향상하였다.

4. VEDA

VEDA[16]는 기존 KV 캐시 제거 및 관리 과정에서 발생하는 중요도 편향 문제를 해결하기 위해 Voting 기반으로 전체 레이어를 보고 중요도를 판단하고 데이터의 최소 유지조건을 만족하면서 불필요한 데이터를 선별하는 알고리즘을 제시한다. 또한, Voting 전용 가속기를 포함한 LLM 추론 프로세서를 설계하여 메모리 절감, 정확도 유지, 추론 성능 향상을 달성하였다.

4.1 Optimization Methods

Casual Attention 구조는 가장 최근에 생성된 토큰이 초기 토큰보다 Attention에 누적되는 횟수가 적기 때문에, 사용 빈도가 적은 이유로 쉽게 삭제될 위험이 있다. 이는 최근 토큰이 더 중요하다는 직관적 해석과도 상충되어 편향된 캐시 제거 문제를 야기한다. 이를 해결하기 위해 Voting 알고리즘은 각 토큰을 투표자로 간주하여 동등한 투표권을 부여하고, 토큰별 투표수를 기반으로 KV 제거 대상을 결정한다. 또한, 최근 토큰을 보호하기 위해 KV 캐시에 Reserved Stage 영역을 설정하여 일정 부분은 삭제 대상에서 제외한다(그림 6).

Attention Score의 분포가 다양한 상황에서 단순 합산 방식으로 중요도를 판단하면 행(Row)마다 점수 기준이 달라 임곗값이 왜곡되는 문제가 발생한다. Voting 알고리즘은 Attention Score의 평균과 표준편차의 선형 조합으로 계산한 적응형 임계치를 사용하여 데이터 분포에 맞춰 행마다 다른 임곗값으로 KV 중요도를 평가한다.

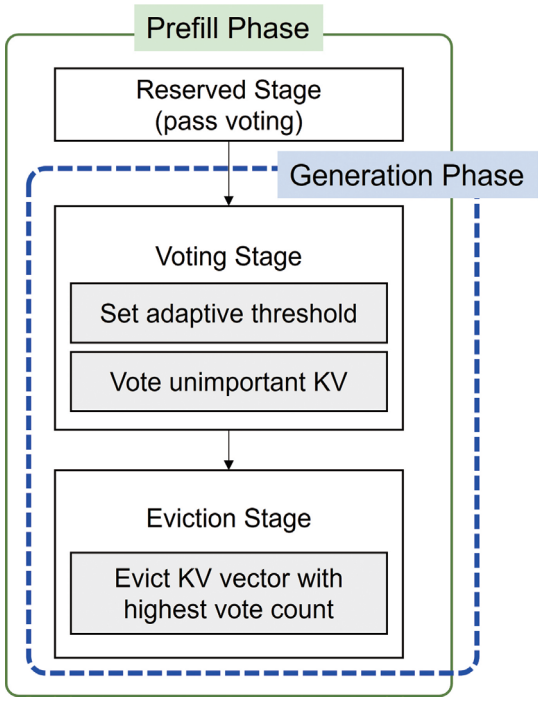


그림 6 VEDA의 Voting 알고리즘 개념도

또한, 일부 토큰이 위치에 따라 비정상적으로 높거나 낮은 Attention Score를 가질 수 있음에도, 편향적으로 큰 값으로 인해 항상 중요한 토큰으로 과대평가 받을 수 있다. Voting 알고리즘은 이러한 Outlier 토큰을 절댓값 크기가 아닌 투표 횟수 기반으로 평가하여 Attention Score가 과하게 높아서 지속적으로 중요 토큰으로 잘못 인식되는 문제를 차단한다.

4.2 HW Architecture

VEDA는 저지연 고효율 LLM 추론 프로세서로써, Voting 알고리즘을 지원하는 Voting 엔진을 포

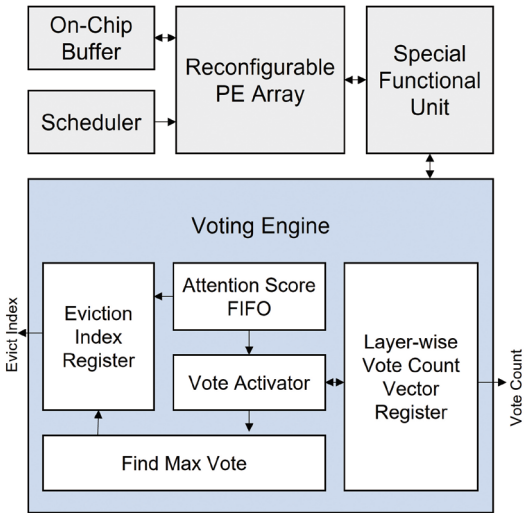


그림 7 VEDA 아키텍처 구조도

함하여 PE(Processing Element) 어레이, SFU(Special Function Unit), 스케줄러, 그리고 HBM 기반 오프칩 메모리로 구성된다(그림 7). Voting 엔진은 불필요한 KV 캐시 삭제와 중요도 평가 연산이 LLM 추론 시간을 저하시키지 않도록 토큰 생성 연산과 병렬로 동작한다. 이를 위해 Voting 엔진은 Softmax 결과를 입력받아 평균 및 표준편차를 계산하고, FIFO와 Reduction Unit을 통해 투표 판단, 투표 수 업데이트, 그리고 제거 대상 데이터의 인덱스 관리가 수행된다. Voting 연산은 토큰이 입력되는 Prefill 단계와 토큰을 생성하는 Decoding 단계 모두에서 수행되지만, 투표 결과를 기반으로 한 KV 캐시의 제거는 Decoding 단계에서만 실행하여 실시간 응답성을 유지한다. 또한, 데이터의 전치(Transpose) 여부에 따라 PE 어레이 내부의 데이터 전송 방식을 조정하여 KV 캐시의 포맷을 균일하게 유지함으로써 HBM 대역폭의 활용도를 극대화했다.

본 연구는 Llama-2[17] 모델을 기준으로, 제안된 KV 캐시 제거 알고리즘의 적용 여부에 따른 성능을 비교하였다. 제안된 알고리즘은 생성 문장 시퀀스의 길이에 관계없이 KV 길이를 고정값으로 유지하는 방식을 사용하여, KV 비고정 방식 대비 최대 10배 빠른 응답 속도를 달성하였다.

IV. 향후 전망 및 결론

이제 LLM 규모 경쟁의 열기는 한풀 꺾였다. 모델은 점차 경량화될 것이고, 다양한 모델을 유기적으로 결합해 서비스 중심의 안정적 시스템으로 구현하는 것이 더욱 중요해지고 있다. 이를 통해 AI는 디지털 영역을 넘어 Physical 계층으로 내려왔고, 로봇이라는 가면을 쓰고 인간과 더 직접적이고 능동적인 상호작용이 가능해졌다.

이 과정에서 장기 기억과 단기 기억을 구분해 관리하는 메모리 관리 시스템의 중요성이 커지고 있다. Physical AI 환경에서는 긴 문맥을 유지하면서도 실시간 반응성을 보장해야 하므로, 최소한의 메모리로 최대 효율을 달성하기 위한 메모리 압축 및 최적화 기법이 필수적이다. 나아가 이러한 메모리 관리 · 최적화를 전담하는 전용 하드웨어 가속기는 향후 실시간 추론시스템에서 선택이 아닌 필수 요소로 자리 잡게 될 것이다.

참고문헌

[1] J. Yang et al., “Magma: A Foundation Model for Multimodal AI Agents,” in Proc. IEEE Comput. Vis. Pattern Recognit., (Nashville, TN, USA), Jun. 2025, pp. 14203-14214.

[2] K. Black et al., “$\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization,” arXiv preprint, 2025. doi: 10.48550/arXiv.2504.16054

[3] Y. Mei et al., “Helix: Serving Large Language Models over Heterogeneous GPUs and Network via Max-Flow,” in Proc. ACM Int. Conf. Archit. Support Program. Lang. Oper. Syst., (Rotterdam, Netherlands), Mar. 2025, pp. 586-602.

[4] S. Abeyruwan et al., “Gemini Robotics: Bringing AI into the Physical World,” arXiv preprint, 2025. doi: 10.48550/arXiv.2503.20020

[5] C. Hooper et al., “KVQuant: Towards 10 Million Context Length LLM Inference with KV Cache Quantization,” in Proc. Int. Conf. Neural Inf. Process. Syst., (Vancouver, BC, Canada), Dec. 2024, pp. 1270-1303.

[6] M. Kim et al., “Oaken: Fast and Efficient LLM Serving with Online-Offline Hybrid KV Cache Quantization,” in Proc. Annu. Int. Symp. Comput. Archit., (Tokyo, Japan), Jun. 2025, pp. 482-497.

[7] J. Wei et al., “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” in Proc. Int. Conf. Neural Inf. Process. Syst., (New Orleans, LA, USA), Nov. 2022, pp. 24824-24837.

[8] S. Yao et al., “ReAct: Synergizing Reasoning and Acting in Language Models,” in Proc. Int. Conf. Learn. Representations, (Kigali, Rwanda), May. 2023.

[9] N. Shinn et al., “Reflexion: Language Agents with Verbal Reinforcement Learning,” in Proc. Int. Conf. Neural Inf. Process. Syst., (New Orleans, LA, USA), Dec. 2023, pp. 8634-8652.

[10] G. Team et al., “Gemini: A Family of Highly Capable Multimodal Models,” arXiv preprint, 2023. doi: 10.48550/arXiv.2312.11805

[11] Z. Qin et al., “CAKE: Cascading and Adaptive KV Cache Eviction with Layer Preferences,” in Proc. Int. Conf. Learn. Representations, (Singapore, Singapore), Apr. 2025.

[12] Y. Bai et al., “LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding,” arXiv preprint, 2023. doi: 10.48550/arXiv.2308.14508

[13] M. Li et al., “NeedleBench: Can LLMs Do Retrieval and Reasoning in 1 Million Context Window?,” arXiv preprint, 2024. doi: 10.48550/arXiv.2407.11963

[14] H. Wang et al., “SQuat: Subspace-orthogonal KV Cache Quantization,” arXiv preprint, 2025. doi: 10.48550/arXiv.2503.24358

[15] L. Gao et al., “A framework for few-shot language model evaluation,” Version v0. 0.1. Sep. 10, 2021, pp. 8-9.

[16] Z. Wang et al., “VEDA: Efficient LLM Generation Through Voting-based KV Cache Eviction and Dataflow-flexible Accelerator,” arXiv preprint, 2025. doi: 10.48550/arXiv.2507.00797

[17] H. Touvron et al. "Llama 2: Open foundation and fine-tuned chat models," arXiv preprint, 2023. doi: 10.48550/arXiv.2307.09288