반도체

Sector Issue Brief 2026. 3. 30

TurboQuant: 메모리 절약에 따른 EGDE AI 산업 확장

Check Point

- 터보퀀트 → Edge AI 산업 확대
- 제한적 연산 환경에서 우월한 성능
- 일반적 연산 환경에서 추가 검증 필요

| Analyst **정우성** | wsjung@ls-sec.co.kr

HBM Long

AI 추론 과정에서 연산은 개별 토큰 생성 시마다 모델 파라미터(Model Weights)와 함께, 과거 컨텍스트에 해당하는 KV cache를 반복적으로 참조하는 구조를 갖는다. 이 과정에서 KV cache는 컨텍스트 길이에 비례하여 메모리 사용량이 선형적으로 증가하며, 특히 HBM(또는 온칩 SRAM)의 주요 병목 요인으로 작용해왔다.

TurboQuant는 KV cache를 평균 약 3.5bit 수준으로 압축할 수 있음을 제시하며, 이는 기존 FP16(16bit) 대비 약 4~5배 수준의 메모리 절감 효과에 해당한다. 이론적으로 동일한 메모리 자원 하에서 유지 가능한 컨텍스트 길이를 그만큼 확장할 수 있음을 의미한다. 즉, 기존 대비 더 긴 문맥을 유지한 상태에서 추론이 가능해지며, 장문 이해, 멀티-턴 대화, RAG 기반 응용에서 정보 활용 범위가 확대된다. 기존 B2C 챗봇 환경에서는 대화가 길어질수록 KV cache 증가로 인해 latency(답변 시간)가 점진적으로 상승하는 현상이 발생해왔다.

TurboQuant는 이러한 메모리 부담을 완화함으로써 동일한 시스템에서도 더 긴 대화 구간까지 안정적인 응답 속도를 유지할 수 있게 한다. 결과적으로 사용자 경험 개선으로 이어질 가능성이 높다. 더 긴 컨텍스트를 활용할 수 있는 환경에서는 사용자 요구 수준이 상승하며, multi-step reasoning, agent 기반 반복 추론 등 더 복잡한 워크로드가 증가하게 된다. 이는 전체 연산량 및 메모리 트래픽 증가로 이어질 수 있으며, HBM 수요 감소보다는 오히려 구조적 수요 유지 또는 확대 요인으로 작용할 가능성이 높다.

Conventional DRAM Super-Long (중장기적)

이러한 SW 차원의 메모리 효율 개선은 HBM 탑재가 제한적인 Edge AI 환경에서 더욱 큰 의미를 가진다. 자율주행, 로봇 등 Edge 디바이스는 일반적으로 수십 GB 수준의 LPDDR 메모리에 의존하며, 이로 인해 모델 크기 및 컨텍스트 길이에 제약이 존재해왔다. 현재 해당 영역에서는 sLLM 중심의 경량 모델이 활용되고 있는 것도 이러한 메모리 제약에 기인한다.

TurboQuant 적용 시 KV cache 메모리 부담이 크게 완화되면서, 동일한 하드웨어 환경에서도 더 긴 컨텍스트 유지 또는 상위 모델 탑재가 가능해진다. 이는 단순한 성능 개선을 넘어, 기존에는 구현이 어려웠던 온디바이스 AI 시나리오를 현실적으로 가능하게 만드는 요인으로 작용한다. 특히 장기 문맥 기반 의사결정이 필요한 **자율주행 및 로보틱스 영역에서 AI 적용 범위 확대를 가속화**할 것으로 판단된다.

다만, TurboQuant는 기본적으로 Transformer 기반 LLM의 KV cache 압축을 다루는 연구로, 현재 Edge AI에서 활용되는 비전 중심 모델(VLM, 멀티모달 모델)이나 센서 융합 기반 시스템과는 거리가 있다. 따라서 TurboQuant가 Edge AI 전반의 성능을 즉각적으로 끌어올리는 직접적인 산업 촉매로 해석하기보다는, LLM 기반 추론 구조의 메모리 효율을 개선함으로써 향후 멀티모달/Agent 기반 Edge AI 아키텍처로 확장될 수 있는 기반 기술로 보는 것이 적절하다.

TurboQuant 개요

등장 배경: 대규모 언어모델(LLM)의 성능은 단순히 모델의 크기뿐 아니라, 얼마나 긴 문맥(Context)을 기억하고 활용할 수 있는지에 의해 크게 좌우된다. 이때 핵심 역할을 하는 것이 바로 KV Cache이다. 모델은 이전 토큰들의 정보를 Key-Value 형태로 저장하고, 새로운 토큰을 생성할 때마다 이를 반복적으로 참조한다.

문제는 KV Cache가 컨텍스트 길이에 비례하여 기하급수적으로 메모리를 소모한다는 점이다. 현재 대부분의 시스템은 FP16(16bit) 기준으로 KV Cache를 저장하고 있으며, 이는 HBM과 같은 고대역폭 메모리에 상당한 부담을 주는 구조다. 결과적으로 컨텍스트 길이는 메모리 용량에 의해 제한되고, 이는 곧 모델의 추론 능력과 직결된다.

이러한 배경 속에서 2025년 4월, Google Research를 중심으로 TurboQuant 논문이 발표되었다. 해당 연구는 KV Cache를 기존 대비 약 4~6배 수준으로 압축하면서도 모델 성능을 유지하는 것을 목표로 한다. 또한 2026년 4월 예정된 학회(**AI 국제학술대회 ICLR 2026, 4월 23~25일**)에서는 TurboQuant의 확장된 형태 또는 후속 연구가 공개될 가능성이 높다.

이론적 중요성: 기존의 벡터 양자화(Vector Quantization) 접근은 기본적으로 원본 벡터 x와 복원된 벡터 $\tilde{x}$간의 차이를 최소화하는 문제, 즉 MSE(Mean Squared Error)를 줄이는 데 초점을 맞춰왔다. Product Quantization(PQ) 계열을 포함한 대부분의 방법론 역시, 고차원 벡터를 분해하고 이를 효율적으로 근사함으로써 MSE 기준에서 높은 압축 효율을 달성하는 방향으로 발전해왔다.

이론적으로도 주어진 bit budget 하에서 달성 가능한 하한(lower bound) MSE가 존재하며, 기존 방법들은 이 MSE 기준에서 상당히 근접한 성능을 보여왔다. 즉, 벡터를 얼마나 비슷하게 복원할 수 있는가라는 문제 자체는 일정 부분 해결된 영역이었다.

그러나 LLM의 실제 작동 방식은 MSE 기준과 다르다. 모델은 벡터 자체를 비교하는 것이 아니라, Query와 Key 간의 내적(inner product = $\langle q, k \rangle$)을 통해 각 토큰의 중요도를 계산하고, 이 값의 상대적 크기(즉 Rank)에 따라 attention을 형성한다. 따라서 중요한 것은 벡터의 절대적 유사도가 아니라, 어떤 토큰이 더 중요한지에 대한 순서이다.

그림1 **목적 함수**

$$\text{(MSE)} \quad D_{\text{mse}} := \mathbb{E}_Q\left[\left\|x - Q^{-1}(Q(x))\right\|_2^2\right] \tag{1}$$

$$\text{(inner-prod error)} \quad D_{\text{prod}} := \mathbb{E}_Q\left[\left|\langle y, x\rangle - \langle y, Q^{-1}(Q(x))\rangle\right|^2\right]. \tag{2}$$

The expectations above are takes with respect to the randomness of the quantizer $Q(\cdot)$. Furthermore, for inner-product quantizers, we require unbiasedness of the inner product estimator, a desirable property for numerous applications. More precisely, we require:

$$\text{(unbiased inner-prod)} \quad \mathbb{E}_Q\left[\langle y, Q^{-1}(Q(x))\rangle\right] = \langle y, x\rangle.$$

We aim to design computationally efficient quantizers Q_{mse} and Q_{prod} that achieve optimal bounds for the distortion measures defined above, for any given bit-width b. Additionally, we aim for Q_{prod} to provide unbiased inner product estimates. In particular, assume that we are given n real-valued vectors $x_1, x_2, \ldots x_n \in \mathbb{R}^d$. We design the following primitives:

- QUANT: efficiently quantizes the dataset and computes $Q(x_1), Q(x_2), \ldots Q(x_n)$.
- DEQUANT: given a quantized dataset, can efficiently reconstruct original vectors by computing $Q^{-1}(Q(x_i))$ for any $i \in [n]$.

자료: Amir Zandieh, et al.(2025), LS증권 리서치센터

그림2 **TurboQuant의 MSE distortion 수준**

Lower Bound. In Theorem 3, we leverage Shannon's lower bound and Yao's minimax principle to prove that for any randomized quantization algorithm $Q : \mathbb{R}^d \to \{0,1\}^{bd}$ with bit-width b, there exist hard input instances $x, y \in \mathbb{R}^d$ with $\|x\| = 1$ such that the following lower bounds hold:

- $D_{\text{mse}}(Q) := \mathbb{E}\left[\left\|x - Q^{-1}(Q(x))\right\|_2^2\right] \geq \frac{1}{4^b}$
- $D_{\text{prod}}(Q) = \mathbb{E}\left[\left|\langle y, x\rangle - \langle y, Q^{-1}(Q(x))\rangle\right|^2\right] \geq \frac{\|y\|_2^2}{d} \cdot \frac{1}{4^b}$

As demonstrated by our lower bounds, TURBOQUANT's MSE distortion is provably within a factor of at most $\frac{\sqrt{3}\pi}{2} \approx 2.7$ of the information-theoretical lower bound. Notably, for smaller bit-widths, this factor significantly decreases. For instance, at a bit-width of $b = 1$ TURBOQUANT achieves a distortion that is only a factor of approximately **1.45** away from the optimal which is also confirmed by our experimental results, indicating its efficiency in low-bit-width scenarios.

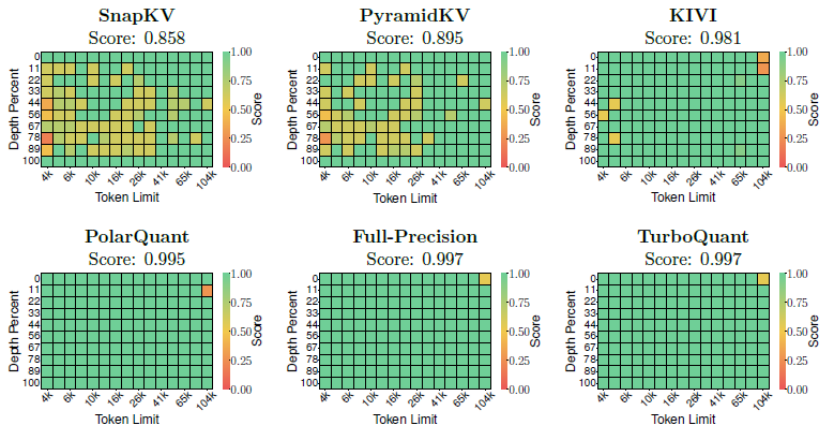
자료: Amir Zandieh, et al.(2025), LS증권 리서치센터

문제는 MSE를 최소화하는 양자화가 inner product까지 보존하지는 않는다는 점이다. 양자화 과정에서 발생하는 오차는 0으로 만들 수 없으며, 이 오차가 특정 방향으로 누적되면서 inner product에 편향(bias)이 발생한다. 그 결과, $\langle q, \tilde{k} \rangle \neq \langle q, k \rangle$이 되고, 특히 중요한 것은 이 값이 체계적으로 과소 또는 과대 평가되면서 토큰 간 중요도의 Rank가 뒤바뀌는 현상이 발생할 수 있다는 점이다. 이는 모델이 참고해야 할 정보를 잘못 선택하게 만들며, 환각이나 비일관적 답변의 원인이 된다.

TurboQuant의 핵심 기여는 바로 이 지점에서 등장한다. 이 연구는 단순히 MSE를 줄이는 것이 아니라, inner product의 편향을 제거하고 Rank를 유지하는 것을 목표로 문제를 전환한다. 구조적으로 보면 TurboQuant는 2단계 접근을 취한다. **첫 번째 단계**에서는 random rotation(기존의 구조를 유지)과 scalar quantization(벡터를 단일 상수로 양자화)을 통해 고차원 벡터를 좌표별로 독립적인 구조로 변환한 뒤, 약 2~3bit 수준의 coarse quantization을 수행한다. 이 단계에서 이미 MSE는 이론적 하한에 근접한 수준까지 줄어든다.

두 번째 단계에서는 남아 있는 residual 오차에 대해 1bit QJL(Quantized Johnson-Lindenstrauss) 기반 보정을 적용한다. 이 1bit 정보는 벡터를 정밀하게 복원하기 위한 것이 아니라, inner product 계산에서 발생하는 편향을 제거하기 위한 functional correction 역할을 수행한다. 결과적으로 Turbo Quant는, ①MSE 기준에서는 이론적 하한에 근접하고, ②inner product 기준에서는 unbiased한 estimator를 구성하며, ③토큰 간 Rank를 안정적으로 유지하는 구조를 동시에 달성한다. 또한, 선행 연구 결과물인 Product Quantization 대비 Data Dependency하지 않아서 Online(즉각적인) 데이터 처리에 활용될 수 있다는 점에서 실용성까지 확보했다.

그림3 Needle-In-A-Haystack test 에서 4 배 압축 기술별 성과 비교



자료: Amir Zandieh, et al.(2025), LS증권 리서치센터
주: 가로축은 문서 사이즈의 토큰 개수, 세로축은 Needle이 위치한 곳, 색깔은 탐색 성과

그림4 KV Cache 압축 방법에 따른 벡터 차원 개수별 양자화 시간 차이 (Llama-3.1-8B-Instruct, 4-

Approach	d=200	d=1536	d=3072
Product Quantization	37.04	239.75	494.42
RabitQ	597.25	2267.59	3957.19
TurboQuant	0.0007	0.0013	0.0021

자료: Amir Zandieh, et al.(2025), LS증권 리서치센터

TurboQuant 한계점

TurboQuant는 KV cache 압축 문제를 이론적으로 정교하게 해결한 연구이나, 실제 산업 적용 관점에서는 다음과 같은 한계가 존재한다. 첫째, 본 연구의 실험은 A100 GPU 환경을 기반으로 수행되었다. 이는 GPU 아키텍처에 최적화된 결과로 해석할 수 있으며, 향후 VR 계열 GPU, TPU, 또는 Edge NPU 등 **다양한 하드웨어 환경에서 동일한 효율과 성능이 재현될 수 있는지는 추가적인 검증이 필요**하다. 특히 메모리 구조와 연산 방식이 상이한 환경에서는 성능 편차가 발생할 가능성이 존재한다.

둘째, 실험이 주로 **8B 수준의 비교적 경량 모델을 대상으로 이루어졌다는 점**도 고려할 필요가 있다. 실제 산업에서 활용되는 LLM은 수백~수천억 개의 파라미터 규모로 확장되고 있으며, 모델 규모가 커질수록 attention 분포의 복잡성 및 quantization 민감도가 증가할 수 있다. 따라서 TurboQuant의 성능이 **대규모 모델에서도 동일하게 유지될지는 추가적인 검증이 요구**된다.

셋째, TurboQuant는 **이론적으로 inner product를 unbiased하게 유지하도록 설계되어 있으나, 이는 기대값 기준의 성질**에 해당한다. 실제 추론 과정에서는 장기 컨텍스트 누적, task 특성 등에 따라 오차가 누적될 가능성이 존재하며, 특정 상황에서는 성능 저하로 이어질 수 있다. 특히 agent 기반 반복 추론과 같이 error 확산 현상이 중요한 환경에서는 이러한 리스크가 확대될 수 있다.

그림5 KV Cache 압축 방법에 따른 LongBench-V1 성능 차이 (Llama-3.1-8B-Instruct, 4-bit 양자화 활용)

Method	KV Size	SingleQA	MultiQA	Summarization	Few shot	Synthetic	Code	Average
				Llama-3.1-8B-Instruct				
Full Cache	16	45.29	45.16	26.55	68.38	59.54	46.28	50.06
KIVI	3	43.38	37.99	27.16	68.38	59.50	44.68	48.50
KIVI	5	45.04	45.70	26.47	68.57	59.55	46.41	50.16
PolarQuant	3.9	45.18	44.48	26.23	68.25	60.07	45.24	49.78
TurboQuant (ours)	2.5	44.16	44.96	24.80	68.01	59.65	45.76	49.44
TurboQuant (ours)	3.5	45.01	45.31	26.00	68.63	59.95	46.17	50.06
				Ministral-7B-Instruct				
Full Cache	16	47.53	49.06	26.09	66.83	53.50	47.90	49.89
TurboQuant (ours)	2.5	48.38	49.22	24.91	66.69	53.17	46.83	49.62

자료: Amir Zandieh, et al.(2025), LS증권 리서치센터
주: Full Cache는 FP16, TurboQuant(KV Size = 3.5)와 Full Cache의 Average가 동일.

LS증권 리서치센터

LPU 활용 AI 추론 시간 모델링

아래는 당사의 AI 추론 시간 모델이다. 컨텍스트 토큰당 저장량인 m_{kv}가 4배 줄어들면, 출력 토큰당 활용할 수 있는 C가 4배 증가하거나 모델 가중치가 3/4 * $m_{kv}C$만큼 증가해도 추론 시간이 유지될 수 있다.

그림6 AI 추론 시간 모델 주요 수식

$$T = T_{SSD} + T_{DRAM} + \max\left(N_{out}\left(\frac{W}{B_{HBM}} + \frac{m_{kv}C}{B_{HBM}}\right), \frac{\phi N_{out} + \phi_{in} N_{in}}{P_{eff}}\right)$$

$$m_{kv} = 2 \cdot L \cdot H \cdot b \quad (\text{bytes/token})$$

자료: LS증권 리서치센터
주: m_kv는 컨텍스트 토큰당 저장량임. L은 AI 모델 레이어 수, H는 Hidden dimension, b는 KV 데이터 타입의 Byte(FP16이면 2, INT4이면 0.5)임.

LS증권 리서치센터

Compliance Notice

본 자료에 기재된 내용들은 작성자 본인의 의견을 정확하게 반영하고 있으며 외부의 부당한 압력이나 간섭 없이 작성되었음을 확인합니다(작성자: 정우성).

본 자료는 고객의 증권투자를 돕기 위한 정보제공을 목적으로 제작되었습니다. 본 자료에 수록된 내용은 당사 리서치본부가 신뢰할 만한 자료 및 정보를 바탕으로 작성한 것이나, 당사가 그 정확성이나 완전성을 보장할 수 없으므로 참고자료로만 활용하시기 바라며 유가증권 투자 시 투자자 자신의 판단과 책임하에 최종결정을 하시기 바랍니다.
따라서 본 자료는 어떠한 경우에도 고객의 증권투자 결과에 대한 법적 책임소재의 증빙자료로 사용될 수 없습니다.

본 자료는 당사의 저작물로서 모든 저작권은 당사에게 있으며 어떠한 경우에도 당사의 동의 없이 복제, 배포, 전송, 변형될 수 없습니다.

_ 동 자료는 제공시점 현재 기관투자가 또는 제3자에게 사전 제공한 사실이 없습니다.

_ 동 자료의 추천종목은 전일 기준 현재당사에서 1% 이상 보유하고 있지 않습니다.

_ 동 자료의 추천종목은 전일 기준 현재 당사의 조사분석 담당자 및 그 배우자 등 관련자가 보유하고 있지 않습니다.

_ 동 자료의 추천종목에 해당하는 회사는 당사와 계열회사 관계에 있지 않습니다.

투자등급 및 적용 기준

구분	투자등급 guide line (투자기간 6~12 개월)	투자등급	적용기준 (향후 12 개월)	투자의견 비율	비고
Sector (업종)	시가총액 대비 업종 비중 기준 투자등급 3 단계	Overweight (비중확대) Neutral (중립) Underweight (비중축소)			
Company (기업)	절대수익률 기준 투자등급 3 단계	Buy (매수) Hold (보유) Sell (매도)	+15% 이상 기대 -15%~ +15% 기대 -15% 이하 기대	88.7% 11.3%	2018 년 10 월 25 일부터 당사 투자등급 적용기준이 기존 ±15%로 변경
		합계		100.0%	투자의견 비율은 2025. 1. 1 ~ 2025. 12. 31 당사 리서치센터의 의견공표 종목들의 맨마지막 공표의견을 기준으로 한 투자등급별 비중임 (최근 1 년간 누적 기준. 분기별 갱신)