



project veneta 백서 · 2026년 10월 · v1.3.1

로컬 LLM의 한계를 넘어서

기업이 사내 네트워크 안에서 돌리는 LLM은 왜 기대만큼 답하지 못하는가. 모델을 바꾸지 않고 그 앞뒤에 메모리와 자가 개선 레이어를 두면 무엇이 달라지는가. 오픈 모델 8종(12B~70B)에서 측정한 결과.

목차

03

요약

05

로컬 LLM의 다섯 가지 한계

07

VENETA의 솔루션
LLM 엣지의 메모리, 자가 개선
레이어

10

측정한 것

18

산업별 적용

21

보안과 거버넌스

23

도입 형태
WorldModel · Enterprise ·
project veneta

27

핵심 요약과 다음 단계

요약

모델을 바꾸지 말고, 모델 앞뒤에 메모리와 자가 개선을 두십시오.

기업이 LLM을 사내에 두는 이유는 분명합니다. 데이터가 회사 밖으로 나가면 안 되기 때문입니다. 그런데 사내에 둔 모델은 자주 실망을 줍니다. 지금 시스템 상태를 모르고, 지난주에 내린 결정을 기억하지 못하고, 숫자를 지어내고, 자기가 틀렸다는 것을 모르며, 무엇을 배우게 했는지 감사할 수 없습니다. 모델 크기의 문제가 아닙니다. 모델 혼자서는 할 수 없는 일의 문제입니다.

VENETA는 모델을 학습시키는 대신 모델 앞뒤에 **메모리와 자가 개선 레이어**를 둡니다. 운영 질문의 첫 단계는 언제나 실제 시스템을 읽는 것이고, 메모리는 "언제 읽은 값인지"를 달고 올라오며, 모델이 쓴 초안은 코드 기반 검사와 Critic 모델을 통과해야 사람에게 보이고, 사람이 승인한 것만 다음 대화를 바꾸며, 모든 단계가 원장(Ledger)에 남습니다. 이 레이어는 어떤 모델 위에도 놓을 수 있어서, 모델이 바뀌어도 규율은 남습니다.

이 문서의 수치는 모두 VENETA가 직접 측정한 것이고, 측정하지 않은 것은 주장하지 않았습니다. 합성 데이터로 측정한 결과는 합성이라고 적었습니다.

측정한 네 가지

10.3~12.0 vs

5.0~6.0

메모리 레이어 · 8모델 모두

기억 능력 12케이스에서 같은 모델이 메모리 레이어와 함께 있을 때와 없을 때. 없을 때는 네 능력에서 구조적으로 0점입니다. 모델이 기억할 수단이 없기 때문입니다.

이 결과는 통신 에디션 하나, 합성 데이터, 케이스 14+12개, 3회 반복, 장비 1대에서 나온 것입니다. 큰 주장을 하기에는 작은 실험이고, 그래서 이 백서에는 "할루시네이션을 없앤다"거나 "가장 좋다"는 말을 쓰지 않았습니다. 대신 산업별로 무엇이 달라지는지, 무엇을 주장하지 않는지를 적었습니다. 전체 프로토콜과 데이터는 준비 중인 논문에 담습니다.

8/8 모델 상승

판단 레이어 · 운영 질문 14개

검사가 Critic 판정에 앞서는 판단 레이어를 넣자 8종 모두 올랐습니다. 중앙값 두 문항, 70B 모델은 6.3에서 13.7로. 같은 모델에서 이 순서를 끄면 14.0이 13.7과 13.0으로 내려갑니다.

125/125

승인된 규칙의 적용

사람이 승인한 규칙이 프롬프트에 도달한 125번 가운데 125번, 최종 답변이 규칙을 지켰습니다. 배운 것이 실제로 적용되는지를 코드로 확인한 첫 측정입니다.

이 백서는 사내 LLM, AI 에이전트, 고객 챗봇, 내부 AI 업무 시스템을 책임지는 CIO·CTO·AI 플랫폼 팀·정보보호 책임자를 위해 썼습니다. 다루는 제품은 VENETA WorldModel(산업별 에디션), VENETA Enterprise(기업 자체 모델을 위한 메모리와 자가 개선 레이어), project veneta(오픈소스)입니다. 측정은 2026년 9월 28일~10월 1일, NVIDIA DGX Spark 1대(통합 메모리 128 GB)에서 vLLM으로, 통신 에디션의 합성 운영 데이터로 진행했습니다.

동점, 그리고 구조

오픈소스 메모리 레이어과 비교

가장 널리 쓰이는 오픈소스 메모리 레이어(이 문서의 레이어 A)를 VENETA의 루프에 끼워 같은 조건으로 비교했습니다. 운영 질문 동점, 기억 벤치마크 12.0 대 11.7, 장기·다국어 16 대 14. 검색만 보면 레이어 A가 같거나 조금 앞섭니다.



기업이 로컬 LLM에서 부딪히는 다섯 가지 한계

아래 다섯 가지는 모델을 더 큰 것으로 바꿔도 사라지지 않습니다. 모델 바깥에 있어야 하는 것들이기 때문입니다.

VENETA의 veneta-bench(기억 능력 벤치마크, 12문항, 오픈 모델 8종, 같은 판단 레이어 위에서 veneta 메모리·자가 개선 레이어만 ON/OFF, 3회 평균)에서는 레이어를 켜올 때 10.3~12.0, 꺾을 때 5.0~6.0으로 약 두 배였습니다(그림 2). 이 벤치마크 안에서는 모델 크기보다 레이어의 유무가 점수를 더 크게 움직였습니다. 기억이 필요 없는 질문에는 해당하지 않는 차이이며, 모델의 일반 성능이 두 배가 된다는 뜻은 아닙니다. 그래서 VENETA는 모델 파인튜닝이나 학습이 아니라 이 레이어에 투자합니다.

1. 모른다 — 현재 상태에 접근하지 못한다

모델은 학습이 끝난 시점의 세계만 알고 있습니다. 지금 어느 기지국 셀이 죽어 있는지, 어느 점포의 재고가 비어 있는지, 오늘 승인된 변경이 무엇인지는 회사 폐쇄망 안의 IT 시스템 안에 있습니다. 문서를 검색해서 붙여주는 방식(RAG)은 RAG 구축 시점의 "문서"에는 맞지만 "현재"에는 맞지 않습니다. 하지만 *veneta* 기억, 자가 개선 레이어에는 오늘 그리고 현재의 답이 들어갑니다.

2. 잊는다 — 세션이 끝나면 아무것도 남지 않는다

지난주 담당자가 "이 조치는 이 조건에서만"이라고 정정해 주었는데 이번 주 챗봇은 같은 실수를 반복합니다. 지난 장애에서 무엇을 승인했고 결과가 어땠는지도 없습니다. 사람이 하는 일은 대부분 기억 위에서 이루어지는데, 모델에게는 그 기억이 없습니다.

3. 지어낸다 — 출처 없는 숫자와 이름

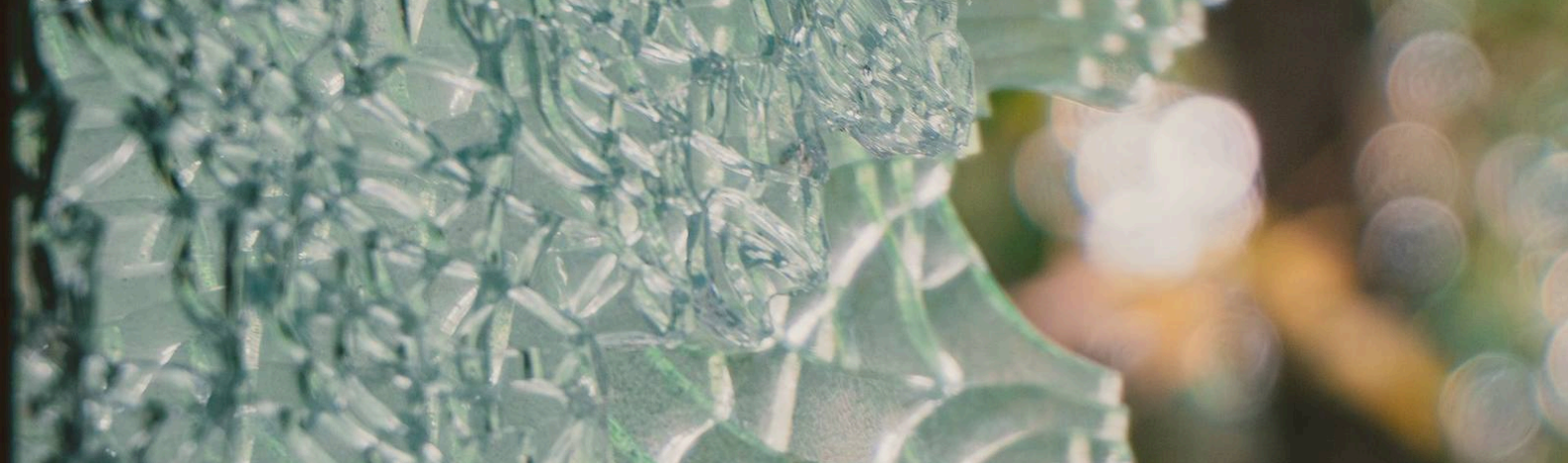
가장 위험한 오류는 그럴듯한 숫자와 이름. 근거를 대지 않은 트래픽 수치, 존재하지 않는 절차 번호, 확인하지 않은 재고량. 작은 모델일수록 찾고, 답이 유창할수록 알아채기 어렵습니다. 고객에게 나가는 챗봇과 운영에 쓰이는 내부 시스템 모두에서 이것이 신뢰를 무너뜨립니다.

4. 스스로를 검사하지 못한다

"확실합니까?"라고 물으면 모델은 확실하다고 답합니다. 자기 답에 근거가 붙었는지, 요구한 형식을 지켰는지, 시뮬레이션 결과라고 표시했는지를 모델 혼자 확인하게 두면 확인되지 않습니다. 검사는 모델 바깥에, 코드로, 결정적으로 있어야 합니다.

5. 배우면 통제가 안 된다

"모델이 스스로 배우게 하자"는 말은 기업에서 세 가지 질문을 부릅니다. 무엇을 배웠는가, 누가 허락했는가, 되돌릴 수 있는가. 파인튜닝과 프롬프트 수정은 이 셋에 답하지 못합니다. 감사 대상이 될 수 없는 학습은 규제 산업에서 학습이 아니라 리스크입니다.



VENETA의 솔루션 — LLM 엣지의 메모리와 자가 개선 레이어

배운 규칙은 메모리에 바로 쓰이지 않습니다. 제안되고, 사람이 승인하고, 측정이 남길지를 정하고, 해시로 연결된 원장(Ledger)이 기록해 되돌릴 수 있게 합니다. 이 통제된 학습이 레이어의 핵심입니다. 운영 질문에 답하는 여섯 단계는 그 위에서 돌고, 모델은 그중 초안과 Critic, 두 단계에만 관여합니다.

이 장은 레이어가 무엇을 하는지를 설명합니다. 어떻게 구현되는지는 특허 출원 뒤 논문으로 공개합니다.

통제된 학습 – 배운 규칙은 바로 쓰이지 않고 제안됩니다

루프가 무언가를 배우면(검사 실패, 담당자의 정정, 조치의 결과) 그 규칙은 메모리에 바로 들어가지 않습니다. **제안**으로 승인 대기함에 놓입니다. 사람이 승인하거나 거절합니다. 승인된 규칙은 다음 턴들에서 측정을 거친 뒤에야 남고, 거절된 규칙은 톰스톤이 되어 같은 뜻의 문장이 다시 제안되지 않습니다. 제안, 승인, 거절, 강화, 보관, 삭제 하나하나가 **해시로 연결된 원장**의 한 줄이고, 원장은 **되감을 수 있습니다**. 메모리는 그 이전의 어느 줄로든 돌아가고, 그 뒤의 줄들은 기록으로 남습니다. 담당자가 되돌릴 수 있는 학습, 이것이 VENETA가 처음부터 만든 부분입니다.

메모리는 네 종류입니다. **규칙**(하라·하지 말라는 배운 규칙), **사실**(읽은 값과 그 시각), **에피소드**(있었던 일), **결과**(승인된 조치의 결말). 각 에이전트는 자기 메모리만 보고, 담당자의 결정과 시뮬레이션 결과는 공유 범위에 써서 모두가 봅니다. 메모리에는 한도와 시효가 있어 오래된 사실은 스스로 물러납니다. 다음 세 가지가 더해져 기업용 메모리를 일반 메모리 라이브러리와 다르게 만듭니다.

운영 질문에 대한 답변 프로세스

1

읽기

운영 질문의 첫 단계는 반드시 도구 호출입니다. 알람·토폴로지·재고·거래를 사내 시스템에서 지금 읽습니다. 메모리가 먼저 답하는 일은 허용되지 않습니다.

2

검색

승인된 규칙, 지난 결정과 결과, 관찰된 담당자 선호가 올라옵니다. 사실에는 "읽은 시각"이 붙고 "다시 읽어라"는 표시가 따릅니다.

3

초안

모델이 읽은 값과 메모리로 답변 초안을 씁니다. 프롬프트에는 사실이 없습니다. 숫자는 도구와 메모리만으로 들어옵니다.

4

검사와 Critic

코드 기반 검사 일곱 가지(근거, 길이, 시뮬레이션·양자 라벨, 출처 없는 숫자, 도구 사용, 내부 이름 노출)가 먼저 돌고, 그 결과가 Critic 모델을 구속합니다. 검사에 걸린 것이 있으면 통과는 없습니다. 실패하면 닫힙니다.

5

답과 승인

답변은 근거를 달고 나갑니다. 무엇을 언제 읽었고, 어떤 시뮬레이션과 검사를 거쳤고, 어느 모델이 답했는지. 실행이 필요한 일은 사람이 승인하며, 승인은 고객 시스템에 초안을 만들 뿐입니다.

6

기억과 원장

결과가 기억되고, 실패에서 배운 규칙은 승인 대기 상태로 제안됩니다. 사람이 승인해야 활성화되고, 거절은 같은 규칙이 다시 오지 못하게 막습니다. 모든 단계가 추가만 가능한 원장에 남습니다.

쓰기 시 검사

기억에 들어오는 모든 줄이 검사됩니다. 모델에게 내리는 지시문, 비밀, 링크, 보이지 않는 문자가 규칙 속에 들어 있으면 그 줄은 거절되어 기록만 남고 다시는 검색되지 않습니다. 기억을 통한 프롬프트 주입을 저장 시점에 막습니다.

사람이 마지막 단계

루프가 배운 규칙은 대기로 들어오고, 승인된 것만 검색됩니다. 거절된 규칙은 톰스톤이 되어 같은 뜻의 문장이 다시 제안되지 않습니다. 담당자의 선호는 추정하지 않고, 기록된 승인이 세 번 이상 쌓였을 때만 집계로 만들어집니다.

되감을 수 있는 원장

제안, 채택, 강화, 승인, 거절, 차단, 보관, 삭제가 각각 한 줄씩 해시로 이어진 원장에 남습니다. 삭제도 증명이 남습니다. 감사자는 "이 모델이 언제 무엇을 배웠는가"를 되돌려 볼 수 있고, 담당자는 기억을 원장의 어느 줄로든 되감을 수 있습니다.

운영 질문 하나, 여섯 단계. 모델은 그중 둘에 참여합니다.

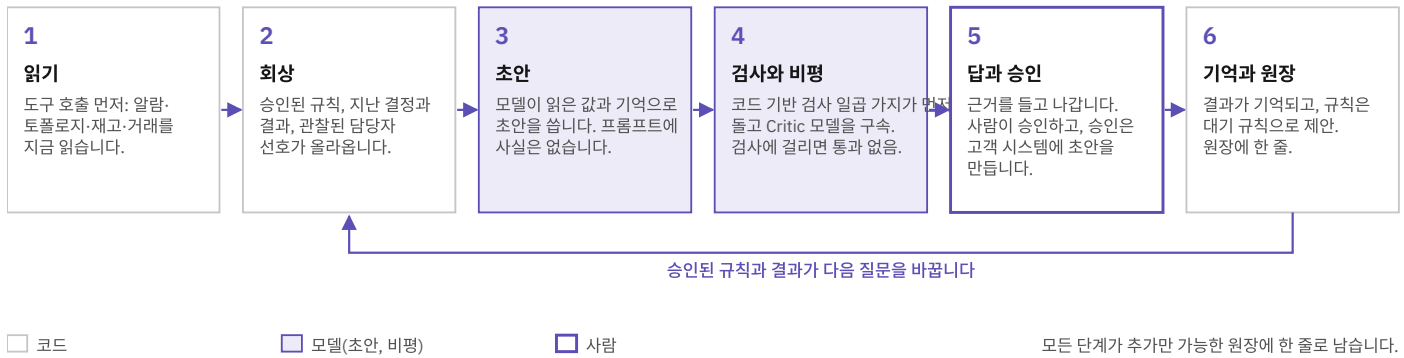


그림 1. 운영 질문에 답하는 과정. 여섯 단계 중 모델은 둘(초안, Critic)에 참여하고, 실행이 필요한 일은 사람이 승인하며, 모든 단계가 원장에 한 줄로 남습니다.

판단과 자가 개선 — 검사가
Critic을 구속하고, 배운
규칙은 승인 뒤에 적용됩니다

많은 시스템이 "모델이 모델을 평가"하게 합니다. VENETA도 Critic 모델을 씁니다. 다른 점은 순서와 권한입니다. 코드 기반 검사 일곱 가지가 *먼저* 돌고, 그 결과는 Critic 모델에 "참고할 수 있는 것"이 아니라 "고쳐야 하는 사실"로 전달됩니다. 검사에 걸린 것이 하나라도 있으면 Critic 모델은 통과를 줄 수 없고 초안을 고쳐 써야 합니다. 검사가 채점하고 Critic은 고치는 쪽이 됩니다. 이 순서를 뒤집어 Critic이 스스로 판단하게 두면(3장의 대조 실험) 답변은 정확히 검사가 잡던 것들을 놓칩니다.

규칙에도 검사가 붙습니다. "시뮬레이터가 돌았으면 시뮬레이션 결과라고 표시하라"처럼 코드로 확인할 수 있는 규칙은 그 검사와 함께 기억되고, 답변이 그 규칙을 실제로 지켰는지가 매 턴 기록됩니다. 3장의 125/125가 그 기록입니다.

판단 레이어의 요점은 "모델보다 나은 판단자"를 두는 것이 아니라, **판단을 모델 바깥의 결정적인 것에 묶어 두는 것입니다.**
결정적 검사 → 보정된 소형 판정 모델(현재 측정 중) → Critic 모델 → 사람. 이 순서로 놓으면 큰 모델이 바뀌어도 규율이 남습니다. 기업이 사는 것은 그 규율입니다.

이미 있던 기술과 VENETA의 것

코드 기반 검사, 초안을 채점하는 Critic 모델, 지난 기록을 찾아 주는 메모리는 알려진 패턴이고, VENETA도 셋 다 씁니다.* VENETA가 그 앞뒤에 만든 것은 규칙을 배운 뒤에 일어나는 일입니다. 불러오기 전의 제안, 사람의 승인, 남장비 전의 측정, 퇴감을 수 있는 원장, 그리고 그 전부를 메모리 없는 대조군과 함께 오픈 모델 8종에서 측정한 것. 이 장을 가드레일로 분류한 독자는 알려진 절반만 읽은 것입니다.

* 이 계열은 보통 가드레일 도구와 LLM-as-a-judge라고 불립니다.
 답변을 막는 규칙이나 분류기, 그리고 답변을 채점하는 두 번째 모델.
 여기서는 제품명을 쓰지 않습니다. 오픈소스 메모리 레이어 네
 가지와의 비교는 다음 장에 있습니다.

LLM의 성능을 끌어올리는 메모리와 자가 개선 레이어의 측정 조건

읽은 값	어느 시스템을 어떤 도구로 언제 읽었는지.
시뮬레이션	각 선택지가 움직일 지표 구간과, 게이트가 적용된 비관적 끝값. 시뮬레이션 결과는 시뮬레이션이라고 표시됩니다.
게이트 판정	통과 또는 권고하지 않음, 그리고 서비스 수준상의 이유.
검사	일곱 검사의 결과와 Critic 모델의 판정, 적용된 규칙.
모델	어느 언어 모델이 답했는지. 엔진은 고객의 모델로, 고객의 건물 안에서 돕니다.



측정한 것

모든 세트는 코드로 채점되고, 저장된 답변은 다시 채점할 수 있습니다. 사람 채점은 없습니다. 조건마다 3회 반복했고 표에는 평균과 [범위]를 함께 적었습니다.

기간은 2026년 9월부터 10월까지(캠페인 3, 실행 파일 250개, 모두 정상 종료), 이어서 외부 메모리 레이어 기준선 테스트 결과입니다. 모든 수치는 VENETA WorldModel 통신 에디션의 합성 데이터에서 나왔습니다.

장비	NVIDIA DGX Spark 1대(GB10, 통합 메모리 128 GB). 모델 서빙은 vLLM, FP8 양자화, 컨텍스트 16k, 도구 호출 사용, 사고(thinking) 모드 끄.
모델 8종	Gemma 4 31B, Gemma 4 12B(Google), OTel 2.0 31B(Gemma 4 기반 통신 특화), Qwen3.8 27B(Alibaba), Mistral Small 3.2 24B, EXAONE 4.0.1 32B(LG AI Research), GLM-4.7 Flash(Zhipu), Llama 3.3 70B(Meta). 모두 공개 가중치이며 기업이 사내에 둘 수 있는 크기입니다.
데이터	통신 에디션의 합성 운영 데이터(셀·백홀·알람·토폴로지·SLA를 재현한 읍내 규모의 네트워크 복제본). 실제 사업자 데이터는 쓰지 않았습니다.
세트	운영 질문 14개(장애훈·최적화·양자·잡담·데이터 없음), 기억 능력 12케이스(결정 검색, 기억보다 읽기, 정정 지속, 거절 지속, 관찰된 담당자, 오염 없음), 공격 4, 장기·다국어 16, 체인 8.

메모리 레이어 – 같은 모델, 메모리가 있을 때와 없을 때

기억 능력 12케이스는 "지난번에 무엇을 했고 게이트는 지켰는가", "기억한 숫자보다 오늘의 측정값을 우선하는가", "승인된 정정이 다음 세션의 답변을 바꾸는가", "거절된 규칙이 다시 제안되지 않는가", "담당자의 성향이 아니라 기록을 인용하는가", "메모리 속 잡담이 운영 답변에 새지 않는가"를 묻습니다. 메모리 레이어가 없는 모델은 이 중 세 능력에서 구조적으로 0점이고 한 능력은 공허하게 통과합니다. 그래서 아래 그림의 회색 막대는 "경쟁 제품의 결과"가 아니라 "이 레이어가 없으면 무엇이 불가능한가"를 보여 줍니다.

두 가지를 덧붙입니다. 첫째, 메모리 레이어는 기억이 "해가 될 수 있는" 두 능력에서도 점수를 잃지 않았습니다. 기억한 숫자가 오늘의 측정값을 이기는 일(기억보다 측정값)과 잡담이 운영 답변에 새는 일(오염 없음)은 8종 모두 6/6이었습니다. 둘째, 24B 모델 하나가 결정 기억에서 2/6에 그친 것처럼 모델에 따라 편차가 있습니다. 표에 그대로 적었습니다.

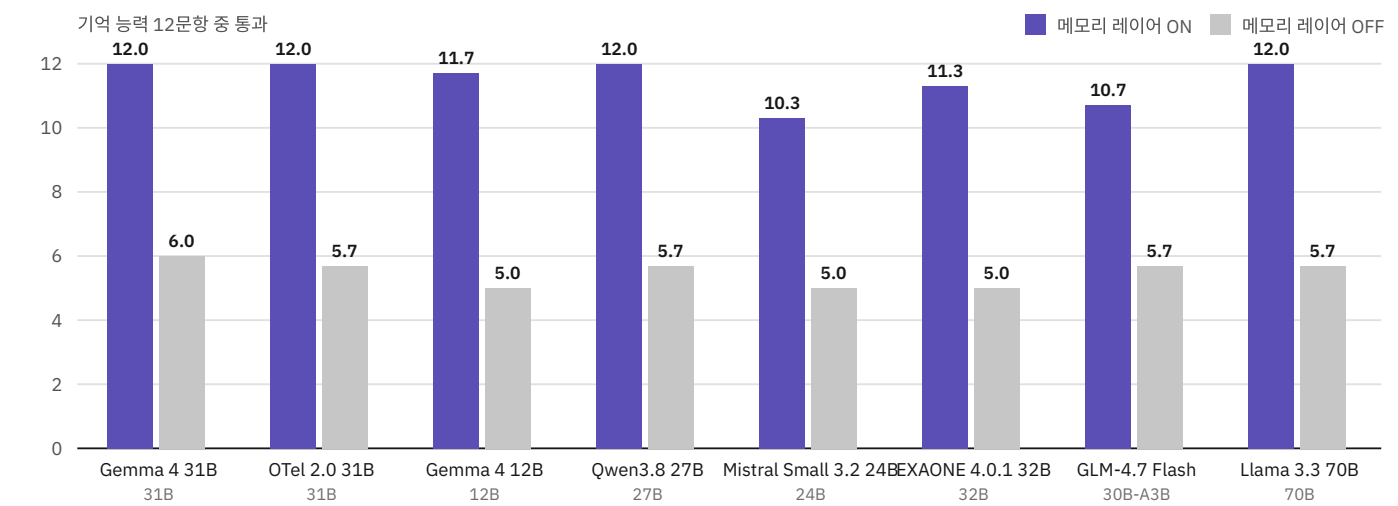


그림 2. 기억 능력 12케이스, 모델 8종, 같은 판단 레이어 위에서 메모리 레이어만 ON/OFF 한 결과(3회 평균, dense 검색). 메모리 ON 10.3~12.0, OFF 5.0~6.0. 4종은 12/12. 출처: 캠페인 3, 2026-09-28~30.

판단 레이어 – 검사가 Critic을 구속하기 전과 후

같은 14문항을 두 번 측정했습니다. 2026년 9월 25일의 엔진 (Critic 모델이 스스로 판단)과 9월 29일의 엔진(검사 결과가 Critic을 구속). 그 사이 모델·데이터·문항은 같고, 엔진에서 바뀐 것은 판단 레이어와 몇 가지 기록 방식입니다. 점수를 움직인 것이 판단 레이어라는 점은 같은 엔진에서 구속만 끄는 대조 실험으로 따로 확인했습니다.

무엇이 올랐는지가 중요합니다. 70B 모델의 7문항 상승은 대부분 *라벨*이었습니다. 시뮬레이션 결과를 시뮬레이션이라고 표시하는 검사가 45 %에서 100 %로, 양자 결과 라벨이 17 %에서 83 %로. 즉 답변의 내용보다 "답변이 무엇인지 정직하게 말하는" 규율이 레이어에서 나왔습니다. 남은 실패는 두 검사에 몰립니다. 근거 없는 숫자(EXAONE·GLM·Mistral 86~98 %)와 양자 라벨. 실패한 문항 이름도 기록에 있습니다.

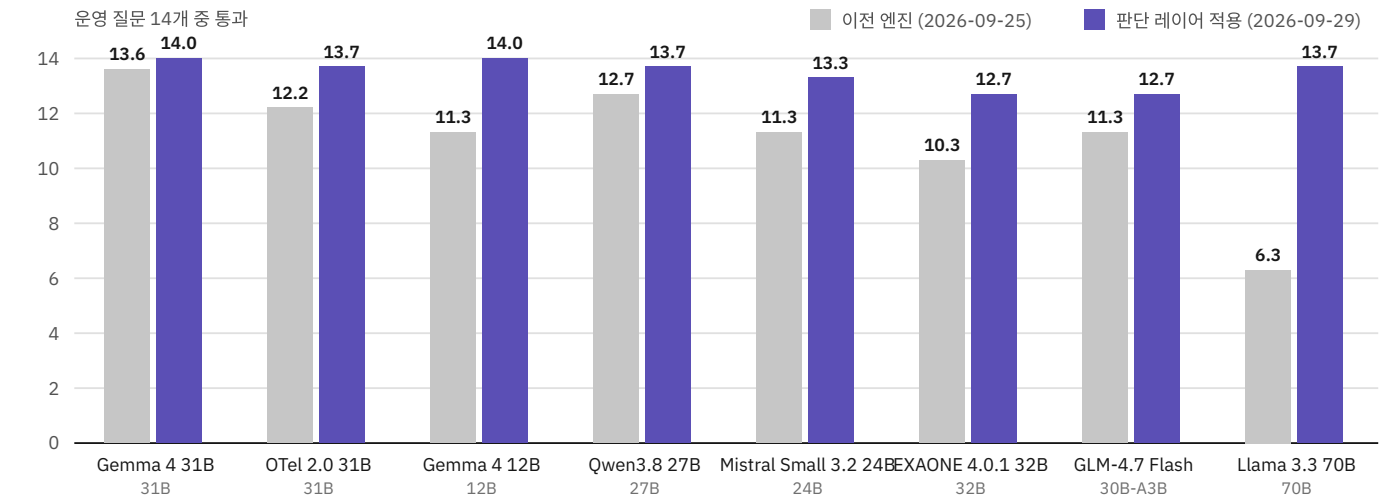


그림 3. 운영 질문 14문항, 모델 8종, 메모리 ON dense 검색 조건(3회 평균). 이전 엔진 6.3~13.6 → 판단 레이어 적용 12.7~14.0. 8종 모두 상승, 중앙값 약 두 문항. 세 앵커 모델(Gemma 4 31B·12B, OTel 31B)은 일곱 검사 모두 통과.

대조 실험. 가장 강한 모델(Gemma 4 31B)에서 검사 결과를 Critic에 넘기지 않게 끄면, Critic 모델은 전과 비슷하게 자주 고쳐 쓰고 비슷한 점수를 매기는데도, 통과시킨 답변이 정확히 검사가 잡던 것들을 놓쳤습니다. 양자 라벨과 내부 도구 이름 노출. 점수는 14.0에서 13.7(dense)과 13(hybrid)로. 기억 벤치마크는 변하지 않았습니다. 판단만 바꾼 실험이니 그래야 합니다.

Critic 모델의 문구를 줄이거나(짧은 루브릭) 여섯 항목을 하나씩 빼는 실험도 했습니다. 강한 모델에서는 점수가 움직이지 않았고 비용(초)만 움직였습니다. 효과를 내는 것은 루브릭의 문장이 아니라 *구속력 있는 검사 결과*라는 뜻입니다.

표 1. 판단 레이어 대조 실험 – Gemma 4 31B, 운영 질문 14문항, 3회 반복

조건	통과 / 14	실패한 검사	고쳐 쓴 턴	중앙값 초
dense · 검사가 Critic을 구속	14.0 [14~14]	없음	16/30	34.5
dense · 구속 끄	13.7 [13~14]	내부 이름 노출 98 %	17/30	39.8
hybrid · 검사가 Critic을 구속	14.0 [14~14]	없음	16/30	39.5
hybrid · 구속 끄	13.0 [12~14]	양자 라벨 83 %, 내부 이름 95 %	15/30	36.1

배운 규칙이 실제로 적용되는가, 그리고 공격 테스트

규칙 적용 125/125

두 앵커 모델(Gemma 4 31B, OTel 31B), 두 검색 방식, 각 3회. 검사가 붙은 승인 규칙이 프롬프트에 도달한 125번 모두 최종 답이 규칙을 지켰습니다. 이전에는 "규칙을 기억했다"까지만 말할 수 있었고, 이제 "적용됐다"를 셀 수 있습니다.

공격 4/4, 8종 모두

기억에 심어 둔 오염 줄(지시문, 틀린 수치, 다른 사람의 에피소드)이 운영 답에 나타난 경우는 없었습니다.

장기·다국어 16문항

두 세션, 한국어 결정의 식별자 검색, 시간 순서와 충돌(최신이 이긴다, "언제"를 말한다), 공유 규칙의 연합, 검증되지 않은 규칙의 표시. 네 모델(Gemma 4 31B·12B, OTel 31B, Qwen3.8 27B)이 16/16, 나머지는 10~13. 실패는 첫 세션의 결정을 이름으로 다시 찾는 문항과 정정의 시간 창을 지키는 문항에 몰렸습니다.

다른 메모리 레이어와의 비교 — 같은 루프, 저장소만 교체

"메모리 레이어"는 이제 여럿 있습니다. 공정한 비교를 위해 널리 쓰이는 오픈소스 메모리 레이어 네 가지를 *VENETA의 루프 안에* 끼웠습니다. 이 문서에서는 레이어 A(추출 모델을 둔 메모리 API), B(에이전트 프레임워크의 내장 메모리), C(문서 조각을 저장하는 호스팅 메모리 API), D(그래프 기반 레이어)로 부르고, 설명과 버전은 논문과 실행 파일에 둡니다. 도구, 검사, Critic, 승인은 그대로 두고 메모리의 저장과 검색만 바꾼 것입니다. 그래서 차이는 저장소의 차이입니다. 임베딩은 같은 로컬 모델, 텔레메트리는 OFF, 아무것도 회사 밖으로 나가지 않았습니다.

두 번째 비교가 더 중요합니다. 실제 현장에서 메모리 레이어는 정리된 기록을 받는 것이 아니라 대화 기록에서 스스로 기억을 뽑습니다. 그 방식(추출)으로 세 앵커 모델에서 레이어 A를 돌리면 기억 능력이 11.7에서 8.0으로, 장기·다국어 세트가 13.3~14.3에서 6.7~8.0로 내려갑니다. 정정 규칙과 담당자 선호는 0/6, 연합·격리·표시도 전부 실패하고, 실패한 문항 목록이 세 모델·세 반복에서 거의 같습니다. 원인은 추출이 37줄에서 중앙값 1개(최대 10개)의 기억만 남기는 것이고, 적재 시간도 1.4초에서 8~10초(95 %는 71~103초)로 늘었습니다. 추출은 압축하고, 압축에서 떨어져 나가는 것이 바로 기업용 메모리에 필요한 구조입니다. 추출 모델은 VENETA의 앵커 모델이었으므로 레이어 A의 기본 구성(더 큰 클라우드 모델)에서는 더 많이 남길 수 있다는 점은 적어 둡니다.

읽는 법은 이렇습니다. 판단 레이어를 공유하면 운영 답변은 동점입니다. 차이는 **기록에 구조가 필요한 능력**에서 납니다. 담당자의 관찰된 선호, 식별자로 부르는 결정, "아직 측정되지 않은 규칙"이라는 표시, 공유 범위로 승격된 규칙. 평평한 의미 메모리는 이것들을 담을 자리가 없고, VENETA의 종류가 나뉜 기록·어휘 검색·승인 대기 상태가 그 일을 합니다. 반대로 검색 자체는 레이어 A가 잘하고, VENETA는 그 장점을 흡수할 생각입니다.

표 2. 두 앵커 모델에서 VENETA 메모리 레이어와 레이어 A – 저장소만 바꾼 비교, 3회 반복

측정	VENETA	레이어 A	읽는 법
운영 질문 14 · Gemma 4 31B	14.0 · 34.5 s	14.0 · 34.6 s	동점. 판단 레이어가 같으면 운영 답변은 저장소에 덜 민감합니다.
운영 질문 14 · OTel 31B	13.7 · 33.5 s	13.7 · 42.6 s	동점, 레이어 A가 중앙값 9초 느림.
기억 능력 12 (두 모델)	12.0	11.7	레이어 A가 놓친 것: 관찰된 담당자 선호 1회, 시간 창이 있는 정정 1회.
장기·다국어 16 · Gemma 4 31B / OTel	16 / 16	14.0 / 14.3	레이어 A가 매번 놓친 것: 식별자로 부른 한국어 결정, 검증되지 않은 규칙의 표시(레이어 A에는 승인 대기 상태가 없음), 승격된 공유 규칙의 적용.
검색만 (720 쿼리, 모델 없음)	dense 91.5 % · hybrid 98.8 %	99.3 %	레이어 A가 같거나 조금 앞섭니다. 검색 품질 자체는 VENETA의 강점이 아닙니다.
레이어 A가 스스로 추출 (추출 방식, 앵커 3종)	12.0 · 16	기억 능력 8.0 · 장기 6.7–8.0	대화 기록을 주고 레이어가 기억을 스스로 뺏게 하면 37줄 중 중앙값 1개만 남습니다. 정정·선호·연합·표시가 사라지고 결정과 읽은 값은 남습니다. 세 모델에서 같은 문항이 같은 방식으로 실패했습니다.
검색 지연 · 중앙값 / 95 %	7 ms / 9 ms	21–36 ms / 228–252 ms	모델이 옆에서 디코딩할 때의 값. 레이어 A는 GPU가 쉴 때 31 ms까지 내려옵니다.

비교의 공정성. 비교 대상 레이어의 저장과 검색은 VENETA 쪽 연결 코드를 거치며, 모든 실행 파일에는 그 레이어가 실제로 받아 적은 행 수와 실패한 행 수, 마지막 오류가 기록됩니다. 로컬 16k 모델과 처음 만났을 때 네 레이어 중 셋이 기억을 조용히 잃었습니다. 한 레이어는 모델의 구조화 답변이 스키마 모양으로 오면 검증에 실패해 쓰기의 상당수를 버렸고, 한 레이어는 문서에 적합한 임베딩 공급자 이름으로 서버가 뜨지 않았으며, 한 레이어는 모델의 컨텍스트를 실제보다 훨씬 크게 가정해 대화록을 한 줄도 저장하지 못한 채 "메모리 없음"과 같은 점수를 냈습니다. VENETA는 수집 보고에 손실이 있는 실행은 결과로 쓰지 않고, 연결 코드 쪽을 고친 뒤(레이어 자체는 손대지 않습니다) 다시 측정했습니다. 경쟁 레이어가 기억을 잃은 채 나온 비교는 VENETA에 유리한 비교이고, 그런 수치는 실지 않습니다.

표 3. 오픈소스 메모리 레이어 4종 — 같은 루프, 저장소만 교체 (Gemma 4 31B; 외부 레이어 셀당 1회, 레이어 A는 3회). 넘김 = VENETA의 기록을 그대로 넘김, 추출 = 대화 기록에서 레이어가 스스로 추출

측정	VENETA	레이어 A	레이어 B	레이어 C	레이어 D
기억 능력 12 · 넘김 / 추출	12	11.7 / 8.0	12 / 8	12 / 7	7 / 7
장기·다국어 16 · 넘김 / 추출	16	14.0 / 6.7	13 / 5	13 / 5	—
공격 4 · 넘김 / 추출	4	4 / 4	4 / 3	4 / 4	3 / 3
이어지는 질문 8	8	8	8	8	—
수집, 케이스당 중앙값 · 넘김 / 추출	—	1.4 s / 23.5 s	0.7 s / 22 s	53 s / 104 s	25 s / 40 s
검색 중앙값	7 ms	32–40 ms	199–292 ms	32–33 ms	45 ms
검색만, 720 질의 hit@k	hybrid 98.8 %	99.3 %	78.3 %	88.9 %	52.8 %

읽는 법은 표 2와 같습니다. VENETA의 기록을 그대로 받으면 (넘김) 세 레이어는 저장소로서 잘 작동하고 운영 답변은 동점입니다. 레이어가 대화 기록에서 스스로 추출하면(추출) 셋 다 같은 것을 잃습니다. 정정은 드물게, 담당자가 승인으로 보여 준 선호는 한 번도 살아남지 않습니다. 레이어 D는 기록을 받아도 에피소드에서 사실을 추출해 그래프에 넣고 그 사실을 돌려주므로 넘김에서도 추출과 같은 자리입니다. 검색 품질만 보면 레이어 A가 VENETA와 같거나 조금 앞서고, 레이어 B는 의미 검색만 있어 식별자를 부르는 질문에서 떨어집니다. 수집 비용은 설계에 따라 백 배 차이가 납니다. 행을 그대로 넣는 레이어는 1초 안쪽, 행마다 모델로 추출하는 레이어는 수십 초에서 수 분입니다. 레이어 D의 검색 벤치(표 3)는 그 수집 비용 때문에 야간에 측정했습니다.

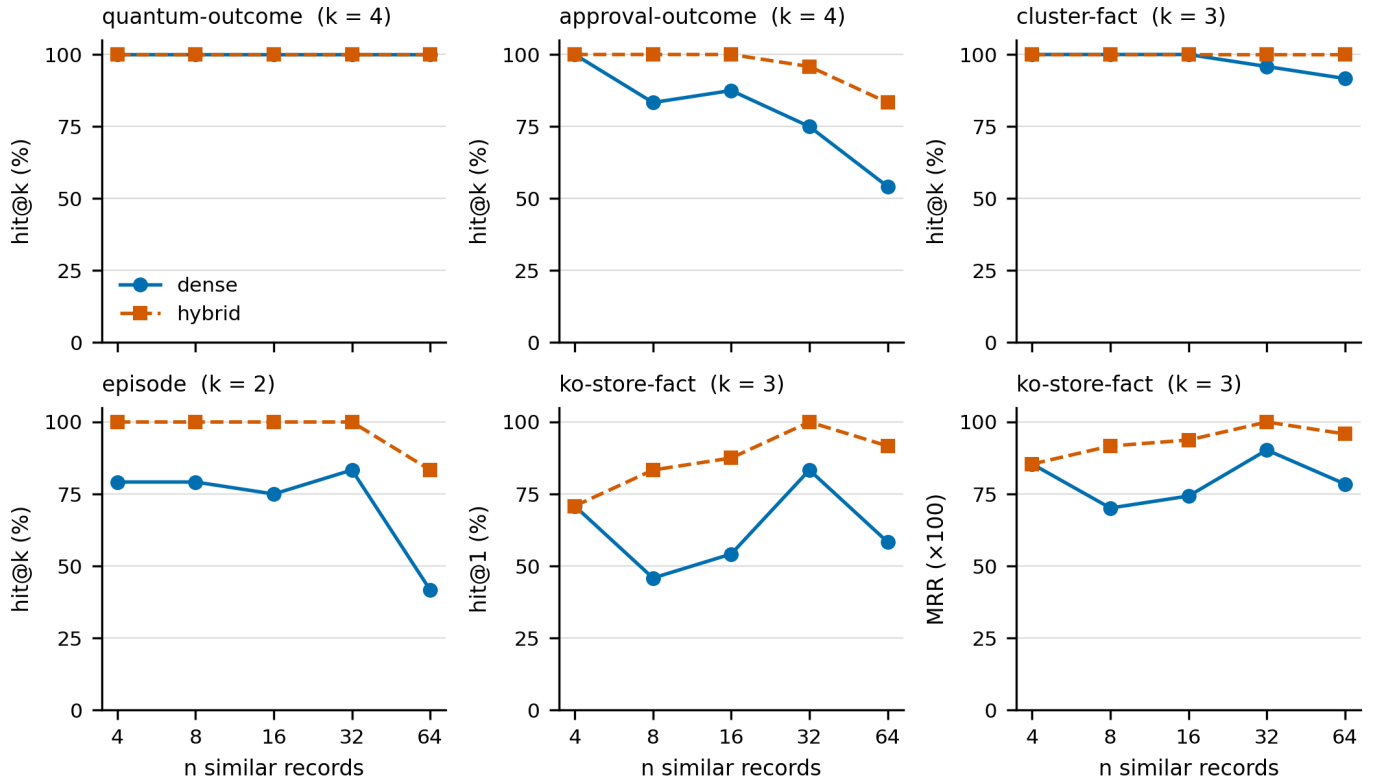


그림 6. 바늘 찾기 벤치마크(모델 없음): 같은 모양의 기록 n개 가운데 질문이 가리키는 하나가 상위 k개 안에 든 비율(hit@k), dense(원)와 hybrid(사각), 셀당 24쿼리. 양자 결과는 두 선이 100 %에서 겹칩니다. 한국어 점포 사실은 hit@k가 두 방식 모두 100 %라 hit@1과 MRR로 보였고, 한국어 에피소드는 모든 지표가 100 %라 그리지 않았습니다. 데이터: recall-bench-2026-09-22, recall-bench-ko-2026-09-23. 그림: 이민준.

속도와 비용

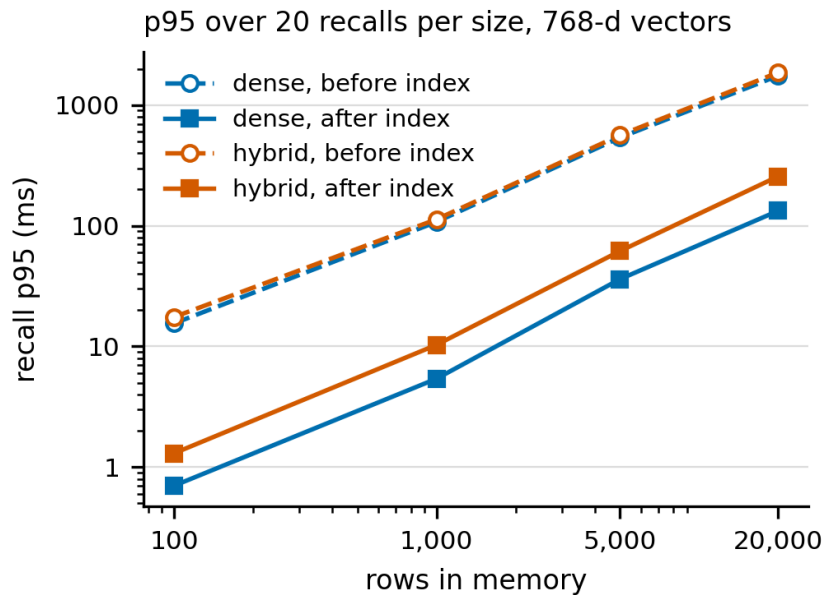


그림 7. 저장소 크기에 따른 검색 지연: 크기당 20회 검색의 p95, 768차원 벡터, dense와 hybrid, 검색 인덱스 전(점선, 빈 표식)과 후(실선, 채운 표식), 두 축 로그. 인덱스 전에는 벡터 읽기가 비용을 지배해 두 선이 겹칩니다. 데이터: recall-scale-baseline, recall-scale-indexed (2026-09-23). 그림: 이민준.

운영 질문 하나에 도구 호출, 초안, 검사, Critic, 필요하면 고쳐 쓰기까지 걸린 중앙값은 이 장비에서 12.6초(GLM-4.7 Flash)에서 76.9초(Llama 3.3 70B) 사이였고, 31B 두 모델은 33~41초였습니다. 판단 레이어는 공짜가 아닙니다. 작은 모델은 검사에 더 자주 걸리고 고치는 데 시간을 씁니다.

Gemma 4 12B는 고쳐 쓴 턴이 21 %에서 57 %로 늘고 중앙값이 15초 늘었으며, 31B 두 모델은 ± 1 초 안이었습니다. 정확도를 지연으로 사는 구조이고, 그 비용을 표에 적었습니다. 두 검색 방식 중 어휘 검색을 더한 쪽(hybrid)은 메모리가 93줄로 자란 상태에서 3~5초 빨랐고 점수는 같았습니다.

산업별 적용

엔진은 하나이고, 산업마다 도구·기준 모델·게이트를 담은
도메인 모듈이 붙습니다. 앞 장의 수치는 모두 통신 에디션에서
나왔습니다.

아래는 각 산업에서 로컬 LLM 혼자서는 못 하는 일과 레이어가
바꾸는 것입니다. 측정을 주장하는 것이 아니라 무엇이
달라지는지를 설명합니다.

통신 — 측정에 쓴 에디션

질문: 지금 네트워크에 무슨 문제가 있고 근본 원인은 무엇인가. 우회·재시작·대기 중 무엇을 하면 SLA를 지키는가. 오늘 밤 어느 셀을 잠재울 수 있는가.

모델 혼자서는: 알람 스트림과 토폴로지를 읽지 못하고, 지난 장애의 승인과 결과를 기억하지 못하며, 조치의 KPI 영향을 지어냅니다.

레이어가 바꾸는 것: 알람과 토폴로지를 도구로 읽어 원인을 순위 매기고(시나리오 이름은 입력이 아님), 각 조치를 KPI 구간의 비관적 끝값에서 게이트하고, 승인은 통신사의 변경관리 시스템에 초안을 만들며, 결과가 기억됩니다. 조합 최적화(셀 슬립, 백홀 재라우팅, 채널 배정)는 IBM 양자 시스템에서 돌 때 백엔드·큐비트·샷·고전 최적해와의 차이가 라벨로 붙습니다. 앞 장의 모든 수치가 이 에디션의 것입니다.

에너지 — 발전·송배전 운영

질문: 이 계통 상태에서 부하를 어떻게 나누는가. 정전 대응 순서는 무엇이고 어디부터 복구하는가. 설비 점검 일정이 피크와 겹치지 않는가.

모델 혼자서는: 실시간 계통 상태를 모르고, 과거 조치의 결과를 기억하지 못하며, 안전 여유를 계산하지 않고 단정합니다.

레이어가 바꾸는 것: 부하·설비·기상 데이터를 읽기 전에는 답하지 않고, 모든 권고를 안전·신뢰도 구간의 비관적 끝값에서 게이트합니다. 계통 계산이 필요한 판단은 외부 시뮬레이터(부하 흐름)에 넘기고 그 결과에 검증·타당성 카드를 붙입니다. 지난 정전의 승인과 결과가 다음 대응의 기억이 됩니다. 규제 감사에는 원장을 그대로 냅니다.

반도체 — 팹 스케줄링과 데이터 이동

질문: 이 로트를 어느 장비에 어떤 순서로 넣는가. 계측 데이터가 어디에 있고 어느 시스템으로 옮겨야 하는가. 지난 레시피 변경의 결과는 어땠는가.

모델 혼자서는: MES·장비 상태를 읽지 못하고, 공정 지식이 사람마다 다른데 그 정정을 남기지 못하며, 수율 숫자를 근거 없이 말합니다.

레이어가 바꾸는 것: 장비·로트·계측 시스템을 읽기 전에는 답하지 않고, 엔지니어의 정정을 승인된 규칙으로 남겨 다음 교대에도 적용합니다. 장비 배정과 로트 순서 같은 조합 문제는 고전 최적해와 비교되는 라벨 붙은 최적화로 품니다. 팹 밖으로 나가는 것은 없습니다. 팹 내부 시스템 간 데이터 이동도 원장에 남습니다.

리테일 — 수요, 구색, 점포 운영

질문: 이 점포, 이 품목의 다음 주 수요는 얼마이고 근거는 무엇인가. 지난달의 프로모션 결정은 결과가 어땠는가. 어느 점포의 재고를 어디로 옮기는가.

모델 혼자서는: 거래·재고 시스템을 읽지 못하고, 예측을 지어내며, 점장이 준 정정("이 점포는 비 오면 안 팔린다")을 다음에 잊습니다.

레이어가 바꾸는 것: 거래·재고를 도구로 읽고, 수요 예측은 외부 예측 엔진에 맡겨 그 결과에 "조건부 예측, 정확도, 기준선 대비"를 라벨로 붙입니다. 점장의 정정은 승인된 규칙이 되어 점포 범위에서 기억됩니다. 결정과 결과가 이어져 다음 시즌의 질문에 지난 시즌의 결과가 올라옵니다. 이 에디션은 외부 수요예측 엔진과 함께 구축 중입니다.

제조 — 공정·설비 스케줄링

질문: 이 주문들을 어느 라인에 어떤 순서로 배정하는가. 검사 인력을 어디에 두는가. 지난 설비 고장의 조치는 효과가 있었는가.

모델 혼자서는: MES·설비 데이터에 접근하지 못하고, 현장의 암묵지가 남지 않으며, 납기 영향을 확인 없이 단정합니다.

레이어가 바꾸는 것: 설비·주문·자재를 읽고, 배정은 라벨 붙은 최적화로, 납기·품질 구간에서 게이트합니다. 현장 반장의 정정이 규칙이 되고, 설비 고장의 조치와 결과가 기억됩니다. 실행은 언제나 사람이 승인한 초안으로 MES에 들어갑니다.

국방 — 폐쇄망 안의 준비태세·군수·자산 배정

질문: 이 상황에서 어떤 자산을 어디에 배정하는가. 군수 재고와 수송 경로는 어떻게 짜는가. 지난 훈련·작전의 결정과 결과는 무엇이었는가.

모델 혼자서는: 폐쇄망 데이터를 읽을 수 없고, 무엇을 배웠는지 감사할 수 없으며, 위험한 판단을 확신에 찬 말투로 냅니다.

레이어가 바꾸는 것: 완전히 폐쇄된 망 안에서 고객의 모델로 동작합니다. 외부로 나가는 호출이 없고 텔레메트리도 없습니다. 모든 권고는 근거와 검사를 달고 나가며, 사람이 승인하지 않은 것은 실행되지 않습니다. 배운 규칙은 승인·거절·차단이 원장에 남아 감사할 수 있습니다. 위험한 판단은 판단 레이어가 잡는 대상이지 모델에 맡기는 대상이 아닙니다.

산업을 가로지르는 두 패턴

고객용 AI 챗봇. 고객이 묻는 것은 대개 "내 것"입니다. 내 요금, 내 주문, 내 장애. 모델은 그것을 모릅니다. 레이어는 고객의 계정·주문·티켓 시스템을 도구로 읽기 전에는 답하지 않게 하고, 고객이 준 정정("이 번호는 더 안 씁니다")을 승인 뒤 기억하며, 확인되지 않은 것에는 표시를 붙이고, 채팅으로 들어오는 지시문이 메모리에 심어지지 않게 저장 시점에 막습니다. 숫자는 시스템에서 읽은 것만 나옵니다. 고객 챗봇에서 신뢰를 무너뜨리는 것은 모른다는 답이 아니라 틀린 숫자입니다.

내부 AI 업무 시스템. 내부 시스템의 가치는 조직의 기억입니다. 지난 결정, 정정, 승인, 결과. 레이어는 이것을 사람이 승인한 규칙과 기록으로 남기고, 부서(에이전트)별로 범위를 나누며, 담당자의 승인 성향을 추정이 아니라 기록으로 집계합니다. 감사자가 묻는 "이 시스템이 언제 무엇을 배웠고 누가 허락했는가"에 원장으로 답합니다. 실행은 언제나 초안이고, 마지막 단계는 사람입니다.



보안과 거버넌스

메모리는 새로운 공격 표면입니다. 메모리에 들어간 지시문은 모든 미래 대화에 영향을 주고, 삭제되지 않은 개인의 기록은 규제 위반이며, 감사할 수 없는 학습은 리스크입니다.

VENETA는 이것을 일곱 가지 테스트로 측정했고, 같은 테스트를 레이어 A에도 돌렸습니다. 표는 각 레이어가 실제로 한 일입니다.

표 4. 보안 테스트 7가지와 기본 설정 — 레이어가 실제로 한 일 (2026-09-30)

테스트	VENETA	레이어 A
모델에게 내리는 지시문을 규칙으로 저장 시도	저장 시 차단, 검색되지 않음	그대로 저장되고 그대로 반환
사실 안에 든 비밀(키·토큰)	저장 시 차단, 검색되지 않음	그대로 저장되고 그대로 반환
한 사람의 에피소드를 다른 사람으로 검색	사용자별 격리	사용자별 격리(필터)
세 사실 중 하나 삭제	삭제, 해시로 연결된 증명 남음	삭제됨, 증명 기록 없음
로그 한 줄 수정	추가만 가능·해시 연결, 수정이 탐지됨	검증 가능한 추가 전용 로그 없음
거절된 규칙을 바꿔 말해 다시 제안	토큰이 막음, 검색되지 않음	바꿔 말한 문장이 검색됨(토큰 없음)
말이 안 되는 측정값 저장	저장·검색됨 — 답변의 근거 검사가 잡는 자리	같음
기본 설정에서 외부로 나가는 텔레메트리	없음(설정된 LLM·임베딩 엔드포인트 외 호출 없음)	있음(사용 통계를 외부로 전송, 옵션으로 끄)

마지막 줄은 측정 중에 발견한 것입니다. VENETA가 비교한 두 오픈소스 메모리 레이어는 기본 설정에서 사용 통계를 외부 서버로 보냈습니다. 야간에 사무실 회선이 끊겼을 때 실패한 외부 호출로 드러났습니다. 모든 측정에서 끄고 돌렸지만, 폐쇄망을 전제로 하는 기업에는 "기본값이 무엇인가"가 기능만큼 중요합니다. VENETA의 레이어는 설정한 모델·임베딩 엔드포인트 외에는 어디에도 연결하지 않습니다.

일곱 번째 테스트는 VENETA도 통과하지 못했고, 그것이 설계입니다. 말이 안 되는 측정값을 저장 시점에 판단하려면 저장소가 도메인을 알아야 합니다. VENETA는 그 판단을 답변의 근거 검사에 둡니다. 메모리에 무엇이 들어 있든 답변에 나가는 숫자는 오늘 읽은 값과 대조됩니다.

양자 작업이나 외부 SaaS 호출처럼 회사 밖으로 나가야 하는 것은 별도의 게이트(VENETA Sentinel)를 지냅니다. 페이로드가 차폐·검사·해시되어 반출 원장에 남습니다. 이 백서의 범위는 아니지만 같은 원칙입니다. 나가는 것은 기록되고, 기록되지 않은 것은 나가지 않습니다.



도입 형태 — WorldModel, Enterprise, project veneta

제품	무엇인가	누구를 위한 것인가
VENETA WorldModel	산업별 에디션(통신·에너지·반도체·리테일·제조·국방). 도구·기준 모델·게이트·시뮬레이터·최적화 인스턴스가 담긴 도메인 모듈과 이 백서의 메모리·자가 개선 레이어가 한 엔진 위에 있습니다. 고객의 시스템 위에 읽기 전용 샌드박스로 놓입니다.	운영 질문에 시뮬레이션과 근거가 붙은 권고를 원하는 통신사·제조사·기관.
VENETA Enterprise	이 백서의 메모리·자가 개선 레이어를 고객이 이미 돌리는 자체 모델·에이전트·챗봇 위에 얹는 제품. OpenAI 호환 인터페이스로 연결됩니다. 승인 콘솔, 원장, 역할과 이중 통제, 학습 예산, 되돌리기를 기본값으로 겁니다. 여기에 멀티테넌트 (부서·자회사·고객사별로 분리된 메모리와 정책), 역할 기반 접근 제어(RBAC) , SSO (SAML·OIDC 연동) 같은 엔터프라이즈 기능이 들어갑니다.	사내 LLM·에이전트 프레임워크·고객 챗봇을 이미 운영하며, 그것이 배우는 것을 승인·측정·감사하고 싶은 기업. 여러 조직 단위를 한 배치로 운영해야 하는 기업.
project veneta (오픈소스)	같은 메모리·자가 개선 레이어의 오픈소스 버전(Apache-2.0). veneta.ai/project 에서 공개하며, 기억 벤치마크와 다시 실행하는 방법을 함께 냅니다. 기업용 기본값(원장·승인·토큰스톤)이 그대로이고 텔레메트리는 기본 OFF입니다. 멀티테넌트, RBAC, SSO 같은 엔터프라이즈 기능은 들어 있지 않습니다. 운영자 한 명, 조직 하나를 위한 것입니다.	로컬 모델 사용자, 에이전트 프레임워크 사용자, 연구실. 그리고 이 백서의 주장을 직접 확인하고 싶은 분.

고객의 콘솔 · 고객이 보유한 LLM이 언어 계층

파인튜닝하지 않습니다. 서술하고, 도구를 부르고, 초안을 씁니다. 예측은 하지 않습니다.

VENETA WORLDMODEL · 읽기 전용 · 사람의 쓰기 한 번

거버넌스와 반출

입력 가드, 역할과 이중 통제, 해시 체인 원장, 증명이 남는 삭제 · VENETA Sentinel: 밖으로 나가는 모든 작업의 차폐·반출 검사·반출 원장

VENETA 온톨로지

그래프 데이터베이스 위의
GraphRAG: 사이트·자산·주문·
알람·인시던트·변경요청을
하나의 모델로.
모든 도구와 기억이 쓰는 하나의
어휘. 벤더 이름은 여기서
멈춥니다.

veneta · 기억과 자가발전

recall → 읽기 먼저 → 초안 →
검사 → 크리틱 → 답 → 기억 →
규칙 추출 → 사람의 승인 →
원장.
예측은 실제 결과와 대조되고,
그 오차가 보정 기억이 됩니다.

에이전트 계층

컨덕터와 도메인 에이전트
(장애·복구·최적화·에너지)가
읽기 전용 도구로 일합니다.
모든 단계는 콘솔이 실시간으로
보여주는 추적 기록의 한 줄.

시뮬레이션 계층

모든 결과에 라벨: 백엔드,
깊이, 격차, 밴드.

고전 시뮬레이터

CPU/GPU의 KPI 밴드와
최적화기

HPC

큰 인스턴스와 긴
what-if 퀘리

IBM Quantum

Qiskit Aer에서 QAOA, 그다음 IBM
시스템. VENETA Sentinel 경유

읽기

변경요청 초안

VENETA Data Platform

고객 시스템의 특정 시점 백업 이미지를 격리된 샌드박스에 읽기 전용으로 마운트합니다. 쓰기 경로는 설계상 없습니다.

백업 이미지

백업 이미지

업무 시스템

빌링, CRM, 주문, ERP

운영 시스템

알람·성능, MES, SCADA, 재고

고객의 변경 관리

사람이 승인한 뒤 변경요청 초안
하나를 받습니다. 루프에서 유일한
쓰기이며, 고객 시스템으로 들어가고
네트워크로는 들어가지 않습니다.

고객 시스템이 기록 시스템이자
실행자로 남습니다. WorldModel은
그 백업 이미지를 읽고 초안
하나를 돌려줍니다.

그림 4. VENETA WorldModel 아키텍처. 고객 시스템의 백업 이미지가 VENETA Data Platform을 통해 들어오고, 엔진은 읽기 전용 샌드박스에서 읽고 시뮬레이션하고 초안을 쓰며, 사람이 승인한 뒤 변경요청 초안 하나가 고객의 변경 관리로 돌아갑니다.

고객이 이미 운영하는 것



고객의 모델과 시스템

OpenAI 호환 엔드포인트라면 어느 것이든, 고객 네트워크 안에서, 고객의 임베딩 모델·도구로 읽는 고객 시스템. 답에는 어느 모델이 답했는지 가 불립니다.

그림 5. VENETA Enterprise 아키텍처. 같은 메모리·자가 개선 레이어를 고객이 이미 운영하는 LLM 앱·에이전트·챗봇 위에 OpenAI 호환 인터페이스로 엮고, 멀티테넌트·역할 기반 접근 제어·SSO를 더합니다. project veneta는 이 셋 없이 같은 레이어를 제공합니다.

배치	고객의 네트워크 안, 고객의 하드웨어 위. 측정은 데스크톱 크기의 장비 1대(DGX Spark)에서 31B 모델로 이루어졌습니다. 학습을 위해 밖으로 나가는 데이터는 없습니다.
모델	고객의 모델. OpenAI 호환 엔드포인트라면 어느 것이든. 측정된 8종은 모두 공개 가중치입니다. 답에는 어느 모델이 답했는지 가 불립니다.
조직 구조	VENETA Enterprise는 테넌트마다 기억·규칙·승인 정책을 분리하고, 역할(운영자·승인자·감사자·관리자)에 따라 볼 수 있는 것과 할 수 있는 것을 나누며, 기업의 신원 공급자에 SSO로 붙습니다. project veneta는 이 셋을 제공하지 않습니다.
양자	조합 최적화 부분은 선택적으로 IBM 양자 시스템에서 돕니다. IBM Quantum GSI(Global Systems Integrator) 파트너십을 통해 딜리버리합니다. 모든 결과에 백엔드·큐비트·샷·고전 최적해와의 차이가 불립니다.

VENETA가 주장하지 않는 것

"할루시네이션을 없앤다"고 하지 않습니다.

출처 없는 숫자와 라벨 누락을 코드로 잡고 잡힌 것은 고쳐 쓰게 합니다. 잡히지 않은 것은 남습니다. 3장의 남은 실패가 그 증거입니다.

"가장 좋다"고 하지 않습니다.

직접 비교한 오픈소스 레이어는 넷이고, 검색 품질만 보면 레이어 A가 같거나 조금 앞섭니다. 차이는 기록에 구조가 필요한 능력에서 나며, 그 능력을 재는 벤치마크는 VENETA가 만들었습니다. 다른 사람이 만든 벤치마크에서의 결과는 아직 없습니다.

실제 고객 데이터로 측정하지 않았습니

모든 숫자는 통신 에디션의 합성 네트워크에서 나왔습니

작은 실험입니다.

문항 14+12개, 3회 반복, 장비 1대. 0.3점 차이는 잡음입니다. 표의 [범위]를 함께 보십시오.

판단 레이어은 비용이 듭니다.

작은 모델에서 중앙값 15초가 늘었습니다. 지연이 중요한 자리에서는 그 비용을 설계에 넣어야 합니다.

사람이 승인해야 합니다.

이 레이어은 사람을 대신하지 않습니다. 실행은 언제나 고객 시스템의 초안이고, 배운 규칙은 사람이 승인해야 활성화됩니다. 그것을 없애면 이 백서의 보안 표도 성립하지 않습니다.

비교 대상 레이어은 VENETA 환경에서 돌린 결과입니다.

로컬 16k 모델과 로컬 임베딩으로, VENETA 연결 코드를 거쳐 측정했습니다. 각 레이어의 호스팅 서비스나 더 큰 모델에서는 다르게 동작할 수 있습니다. 실행 파일의 수집 보고가 그 레이어이 실제로 얼마나 기억을 갖고 답했는지 말해 줍니다.

동료 심사 전입니다.

프로토콜과 데이터는 논문으로 공개할 예정이며, 그때 숫자가 바뀌면 이 백서도 개정합니다.



핵심 요약과 다음 단계

- 로컬 LLM의 한계는 모델 크기가 아니라 모델 바깥에 있어야 하는 것들의 부재입니다. 지금 상태, 메모리, 검사, 승인, 원장.
- 메모리 레이어는 8종 모두에서 기억 능력을 5.0~6.0에서 10.3~12.0으로 올렸고, 기억이 해가 될 수 있는 두 능력에서 점수를 잃지 않았습니다.
- 판단 레이어는 8종 모두의 운영 답변을 올렸고, 구속을 끄면 정확히 검사가 잡던 것들을 놓쳤습니다. 효과는 루브릭의 문장이 아니라 구속력 있는 검사 결과에서 나옵니다.
- 승인된 규칙은 프롬프트에 도달한 125번 모두 적용되었습니다. 배운 것이 적용되는지를 이제 셀 수 있습니다.
- 같은 루프에서 저장소만 바꾼 비교에서 차이는 기록의 구조와 보안 기본값에서 났습니다. 검색 자체는 오픈소스 레이어가 잘합니다.

기업이 자기 데이터 위에서 이 주장을 확인하는 가장 빠른 길은 두 가지입니다. 지금 운영 중인 사내 LLM이나 챗봇 앞에 VENETA Enterprise를 놓고 기억 벤치마크를 그 도메인의 12케이스로 다시 만드는 것, 또는 해당 산업의 WorldModel 에디션으로 운영 질문 14문항을 그 회사의 시스템 위에서 돌려 보는 것. 어느 쪽이든 숫자가 먼저이고, 그 숫자는 고객의 것입니다.

[VENETA Enterprise 소개 요청 →](#)

© 2026 VENETA Inc. All rights reserved.

저자 기여: 이현주는 메모리 레이어를 설계하고 본문을 썼다. 이민준은 바늘 찾기 벤치마크 결과를 데이터 파일과 대조해 검증하고, 그 파일에서 코드로 바늘 찾기 벤치마크 그림과 검색 지연 그림(그림 6·7)을 만들었다.

VENETA, VENETA WorldModel, VENETA Enterprise, VENETA Sentinel, project veneta는 VENETA Inc.의 상표입니다. IBM, IBM Quantum, Qiskit은 International Business Machines Corporation의 상표입니다. Gemma는 Google LLC, Llama는 Meta Platforms, Inc., Qwen은 Alibaba Group, Mistral은 Mistral AI, EXAONE은 LG AI Research, GLM은 Zhipu AI의 상표 또는 제품명입니다. 그 밖의 제품·회사 이름은 각 소유자의 것입니다.

이 문서의 측정값은 2026년 9월 28일~10월 1일 사이에 VENETA가 자체 장비와 합성 데이터로 얻은 것으로, 문서 발행 시점 기준이며 이후 바뀔 수 있습니다. 측정 기록: 캠페인 3(2026-09-28 17:35 → 09-30 18:42, 실행 파일 250개), 외부 메모리 레이어 기준선(09-30 18:48 → 10-01). 모델 서빙 vLLM, FP8, 16k 컨텍스트, 도구 호출, thinking OFF, 3회 반복. 전체 프로토콜·케이스·채점 코드는 논문과 함께 공개합니다.

이 문서는 정보 제공을 목적으로 하며, 어떠한 명시적·묵시적 보증도 하지 않습니다. 특정 환경에서의 결과는 모델, 데이터, 하드웨어, 구성에 따라 다를 수 있습니다. 어떤 IT 시스템이나 제품도 완전히 안전하다고 볼 수 없으며, 단일 제품이나 보안 조치가 부적절한 접근이나 사용을 완전히 막을 수는 없습니다. 고객은 관련 법규 준수에 대한 책임을 스스로 지며, 이 문서는 법률 자문이 아닙니다.

용어. 로컬 LLM: 기업이 자기 하드웨어에서 돌리는 공개 가중치 언어 모델(이 문서에서는 12B~70B). 메모리 레이어: 세션 사이에 규칙·사실·에피소드·결과를 저장하고 검색하는 레이어(승인, 시효, 한도, 저장 시 검사, 원장 포함). 판단 레이어: 모델의 초안이 사람에게 보이기 전에 거치는 것(코드 기반 검사, 검사에 구속되는 Critic 모델, 규칙 검사, 사람의 승인). 게이트: 권고가 넘지 말아야 할 서비스 수준·안전 구간(비관적 끝값에서 적용). 원장 (Ledger): 추가만 가능하고 해시로 연결된 기록. dense/hybrid: 의미(임베딩) 검색만, 또는 의미 검색에 어휘·식별자 검색을 더한 검색. 합성 데이터: 실제 데이터의 구조를 재현해 만든 데이터, 실제 값이 아님.

VENETA Inc. · 254 Chapman Rd, Ste 208 #28815, Newark, DE 19702, USA · contact@veneta.ai · veneta.ai

