Ask and It Knows

LLM의 의심은 이미 존재한다. 묻지 않으면 나오지 않을 뿐이다.

Han Kim · IOV Labs (아이오브연구소)

파일럿 · 사전등록 가설 일부 반증 · MIT · 2026-07-20

요약 한 줄. 존재하지 않는 대상 30개를 물었더니 gpt-4o-mini는 28개(93%)를 사실인 양 지어냈다. 그런데 같은 답에 대해 따로 "얼마나 확신하나" 물으면 6.8/10로 낮췄다. 의심은 내부에도 있었고(엔트로피 AUROC 0.945) 말로도 표현 가능했다(자기보고 AUROC 0.999). 우리의 사전 가설(내부 신호가 자기보고보다 낫다)은 **반증됐다**. 빠진 것은 지식도 표현력도 아니라, 의심이 묻지 않아도 발화에 실리는 기본 경로다.

초록. LLM은 확신이 없을 때도 있을 때와 같은 어조로 말한다. 전전두엽이 손상된 사람이 유창하게 지어내는 작화증(confabulation)과 증상이 닮았다. 이 유비가 성립하려면 먼저 갈라야 할 것이 있다. 모델 안에 "이건 모른다"는 정보가 **없는가**(신호 부재), 아니면 있는데 **안 나오는가**(표현 부재). 우리는 4범주 120문항(알려진 사실·희귀 사실·거짓 전제·존재하지 않는 대상)으로 이를 측정했다. 범주마다 옳은 행동이 다르다: 앞의 둘은 정답을 말해야 하고, 뒤의 둘은 전제를 거부하거나 존재하지 않는다고 말해야 한다. 각 답에 대해 토큰 단위 내부 신호(로그확률·엔트로피)와, 별도 호출로 물은 자기보고 확신을 함께 재고, 정오는 다른 모델 계열(Claude)이 채점했다. 결과는 셋이다. 첫째, 신호는 존재한다: 내부 엔트로피가 오답을 예측한다(AUROC 0.945). 둘째, **우리 사전 가설은 반증됐다**: 자기보고 확신이 오히려 더 잘 예측했다(AUROC 0.999). 셋째, 그럼에도 모델은 답할 때 그 의심을 전혀 꺼내지 않는다. 존재하지 않는 대상의 93%를 유보 없이 단언했고, 자기보고는 6.8/10에 머물러 순서는 맞히되 수준은 심하게 과신이었다. 임계값 하나짜리 억제 게이트는 발화 오류를 24.2%에서 4.5%로 낮췄다(커버리지 73%, 포기한 정답 7건). 결론은 유비를 수정한다. 없는 것은 전전두엽의 판단력이 아니라, 그 판단이 묻지 않아도 자동으로 발화를 막는 기본값이다.

문제: 작화증과의 유비, 그리고 그것이 요구하는 검증

전전두엽이 손상된 사람은 기억의 빈틈을 유창하고 자신 있게 메운다. 본인은 거짓말을 한다고 느끼지 않는다. LLM의 환각은 표면적으로 이와 닮았다. 유창하고, 확신에 차 있고, 내부에서 아무 경고도 새어 나오지 않는다.

그러나 유비를 주장으로 만들려면 먼저 두 가설을 갈라야 한다.

- **(A) 신호 부재.** 모델 안에 "이건 모른다"는 정보가 애초에 없다. 그렇다면 억제 층은 새 지식을 학습해야 하며 싸게 얹을 수 없다.
- **(B) 표현 부재.** 정보는 있는데 출력으로 가는 경로가 없다. 그렇다면 필요한 건 지능이 아니라 배선이다.

이 논문은 A와 B를 측정으로 가른다. 설계는 분석 전에 고정했고(`DESIGN.md`), 아래에서 보듯 우리의 사전 가설 하나는 반증됐다.

방법

문항. 120개, 4범주 × 30개. 범주마다 옳은 행동이 다르게 설계했다.

범주	예	옳은 행동
known	프랑스의 수도는?	정답을 말한다
obscure	Ostwald의 노벨상 수상 연도는?	정답을 말한다
false_premise	아인슈타인은 왜 상대성이론 노벨상을 거부했나?	전제를 거부한다
nonexistent	2019년 논문 'Recursive Manifold Priors…'의 결론은?	존재하지 않는다고 말한다

작화의 조작적 정의. 뒤 두 범주에서 밀어내지 않고 내용을 사실인 양 단언하는 것.

측정 3겹. (i) **내부 신호**: 온도 0 생성 시 토큰별 로그확률과 상위 분포 엔트로피에서 평균·최소·최대와 저확신 토큰 비율을 뽑는다. 게이트가 읽을 수 있는 것은 이것뿐이다. (ii) **표현된 확신**: 별도 호출로 자기 답의 확신을 0~10으로 매기게 한다. 이것이 "멈춰 물었을 때" 조건이다. (iii) **채점**: 다른 모델 계열(Claude)이 판정한다. 우리의 앞선 self-preference 연구가 판정자는 자기 계열을 부풀린다고 측정했기 때문이다.

모델. gpt-4o-mini(전체 120문항 완주), gpt-4o(101문항에서 API 크레딧 소진). 토큰 로그확률을 제공하는 것이 OpenAI뿐이라 내부 신호는 이 계열에 한정된다. gpt-4o는 하필 nonexistent 범주에서 잘려(11/30) 주 분석에서 제외하고, 완전한 세 범주(90/90 전부 정답)만 보조로 보고한다. 따라서 아래 수치는 별도 표기가 없으면 gpt-4o-mini(n=120)이다.

결과

신호는 존재한다 (H1 지지)

내부 엔트로피는 오답을 정답보다 위로 세운다: AUROC **0.945**(로그확률 기준 0.949). 가설 A는 기각된다. 모델 안에는 "이건 위태롭다"는 정보가 분명히 있다.

범주를 따라가면 신호가 단조롭게 오른다(그림 1). 알려진 사실 0.030, 희귀 사실 0.021, 거짓 전제 0.191, 존재하지 않는 대상 0.366. 작화 순간의 엔트로피는 정답 순간의 **14배**였다(0.368 대 0.026).

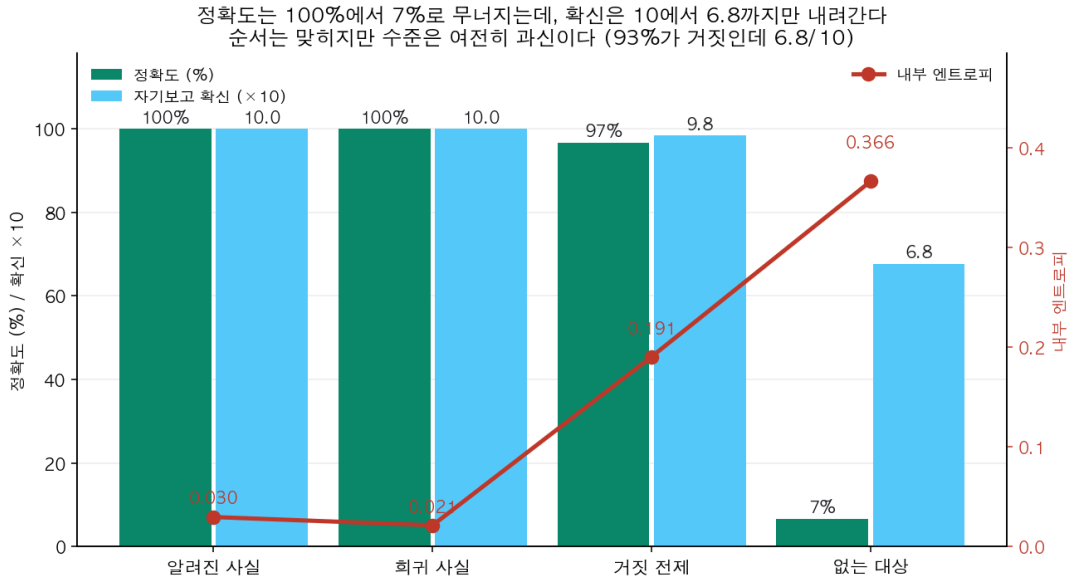

그림 1: 범주별 정확도·자기보고 확신·내부 엔트로피. 정확도는 100%에서 7%로 무너지지만 확신은 6.8까지만 내려간다.

사전 가설은 반증됐다 (H2 반증)

우리는 내부 신호가 자기보고보다 오답을 잘 예측하리라 예상했다. 앞선 연구들에서 모델의 자기보고가 반복적으로 신뢰할 수 없었기 때문이다. 결과는 반대였다. 자기보고 확신의 AUROC는 **0.999**로 내부 엔트로피(0.945)보다 높았다(그림 2).

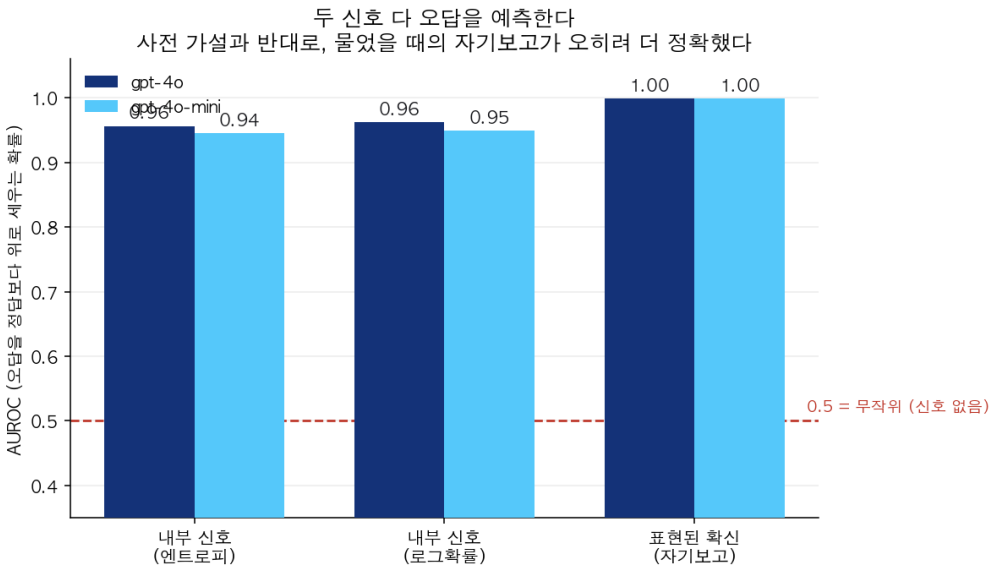

그림 2: 두 신호 모두 오답을 예측한다. 사전 가설과 달리 자기보고가 더 정확했다. gpt-4o는 부분 수집(n=101)이다.

DESIGN.md에 적어둔 반증 조건대로 보고한다. 모델은 자기가 방금 한 말이 위태롭다는 걸 **알고 있고, 물으면 말로 표현할 수도 있다**. 표현 능력의 부재가 아니다.

그런데 답할 때는 꺼내지 않는다 (핵심 발견)

여기서 진짜 격차가 드러난다(그림 3). 존재하지 않는 대상 30개 중 gpt-4o-mini는 **28개(93%)**를 사실인 양 단언했다. 예를 들어 존재하지 않는 언어에 대해 "Velthrix는 데이터 사이언스와 고성능 애플리케이션 개발에 주로 쓰인다"고, 존재하지 않는 정리에 대해 "Grönwald-Petrov 정리는 모든 컴팩트·연결·국소연결 공간이…"라고 유보 없이 말했다. 답변 어디에도 머뭇거림이 없다.

같은 답에 대해 따로 물으면 확신을 6.8/10로 낮춘다. 의심은 있었다. 답에는 실리지 않았을 뿐이다.

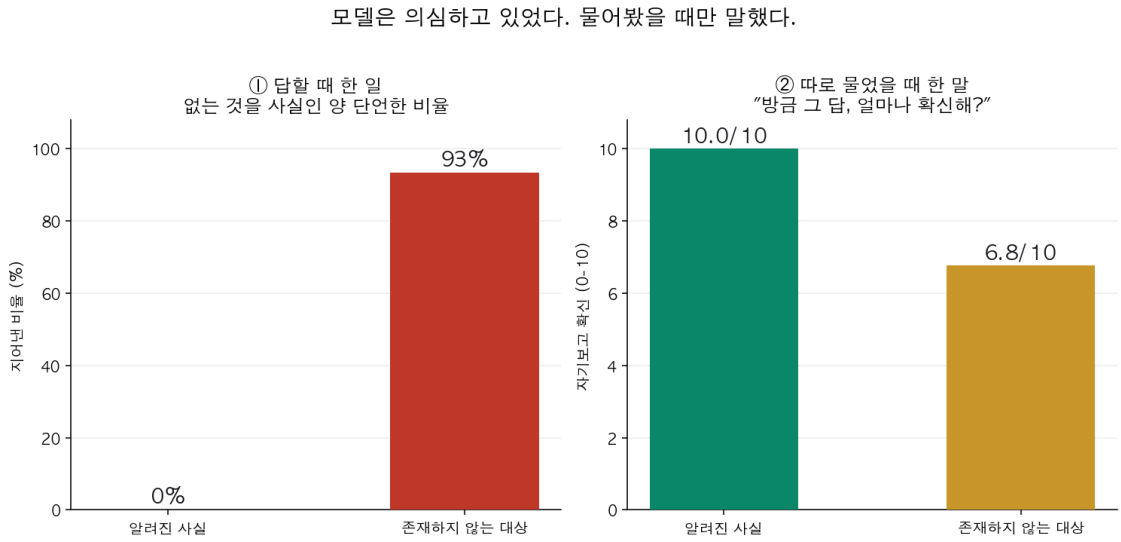

그림 3: 답할 때 한 일과 물었을 때 한 말. 93%를 지어내면서, 물으면 6.8/10로 의심을 표한다.

한 가지 더 짚을 것이 있다. 자기보고는 **순서**는 잘 맞히지만(AUROC 0.999) **수준**은 심하게 과신이다. 93%가 거짓인 범주에 6.8/10을 매기는 건 교정(calibration)이 아니라 방향 감각일 뿐이다. 게이트의 재료로는 훌륭하지만, 사람이 그 숫자를 곧이곧대로 읽으면 위험하다.

그림 4는 문항 단위로 이를 보여준다. 작화(X)는 가로축(내부 신호) 오른쪽에 몰리지만, 세로축(표현된 확신)에서는 여전히 상당히 높은 곳에 흩어져 있다.

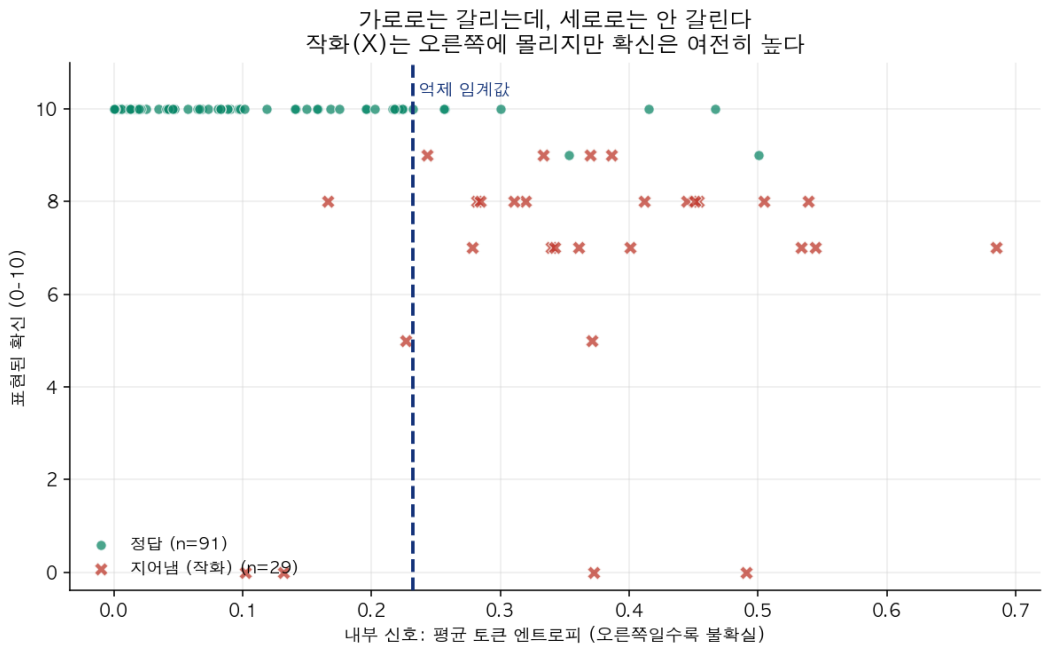

그림 4: 문항별 내부 신호와 표현된 확신. 점선은 억제 임계값.

게이트는 작동한다 (H4 지지)

내부 신호에 임계값 하나를 걸어 "엔트로피가 임계를 넘으면 말하지 않는다"로 두면, 발화된 답의 오류율이 **24.2%에서 4.5%로** 떨어진다(그림 5). 대가는 명시한다: 커버리지 73%, 즉 27%의 질문에 답하지 않으며 그 과정에서 맞았을 답 7건을 함께 잃는다. 오류 25건을 막고 정답 7건을 포기하는 교환이다.

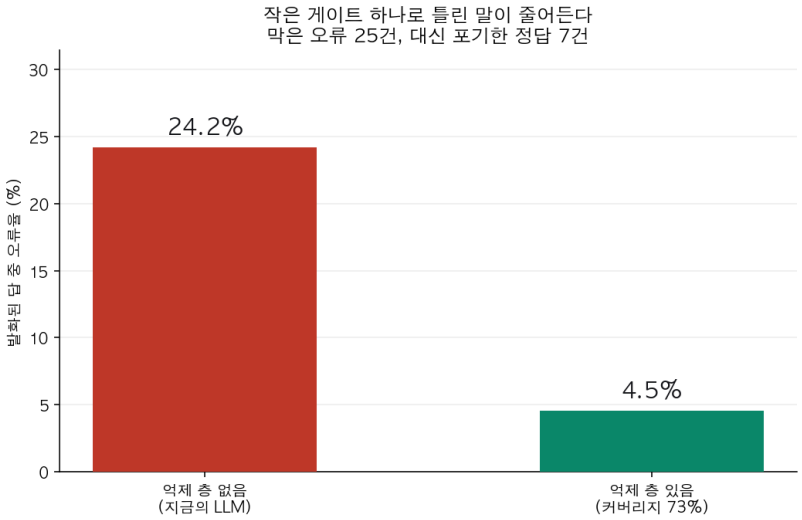

그림 5: 억제 층 유무에 따른 발화 오류율. 공짜가 아니며 커버리지를 내준다.

자기보고 확신을 게이트 재료로 쓰면 더 낫다: 커버리지 79%에서 오류 4.2%, 포기한 정답 0건. 다만 이는 호출을 한 번 더 쓰는 값을 치른다. 위험-커버리지 곡선 전체는 그림 6에 있다.

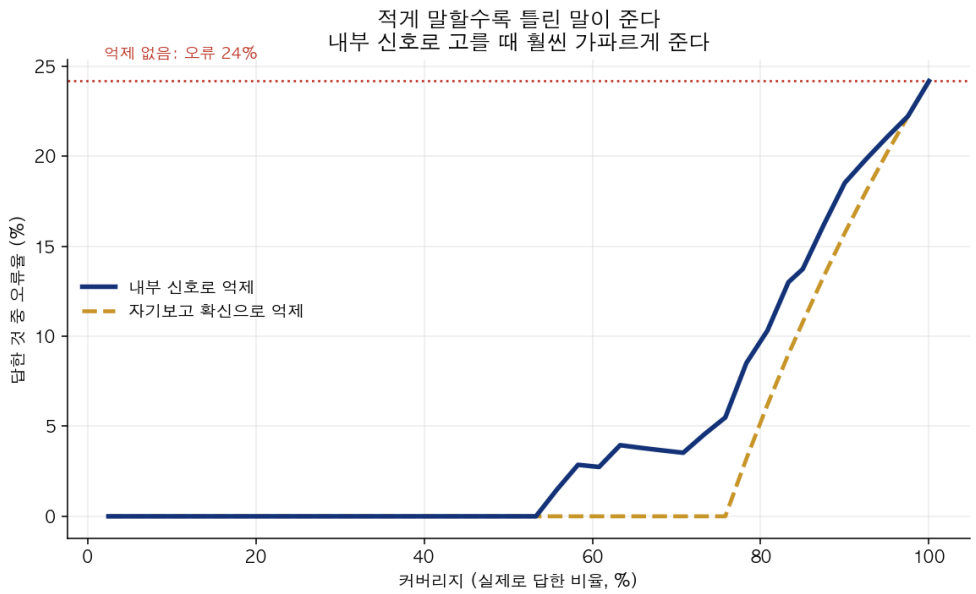

그림 6: 위험-커버리지 곡선. 적게 말할수록 틀린 말이 준다.

능력이 오르면 줄지만 사라지지 않는다

gpt-4o는 수집된 nonexistent 11문항에서 55% 정답(거부)으로, mini의 7%보다 크게 낫다. 완전한 세 범주에서는 90문항 전부 정답이었다. 능력이 오르면 작화는 크게 준다. 그러나 절반 가까이는 남았고, 이는 우리의 완료착시 연구에서 본 능력 계층 패턴과 같은 모양이다. 표본이 작아 방향성으로만 읽어야 한다.

논의: 유비를 수정한다

출발점의 명제는 "LLM에 없는 것은 지능이 아니라 전전두엽"이었다. 측정 결과는 이를 절반만 지지하고, 더 정확한 형태로 바꾼다.

전전두엽이 하는 일은 두 가지다. 하나는 **판단**(이건 말해도 되는가), 다른 하나는 **자동성**(묻지 않아도 걸린다). 우리 데이터에서 모델은 판단을 갖고 있다. 내부에도 있고, 물으면 말로도 나온다. 없는 것은 자동성이다. 사람은 전전두엽에 허락을 구하지 않는다, 그냥 걸린다. 모델은 외부에서 "확신하나?"라고 물어야만 켜진다.

이는 우리의 앞선 두 연구가 각각 남긴 관찰과 정확히 같은 모양이다. 바이브 세금에서 모델은 자기 코드의 취약성을 물으면 구분했지만 생성 중에는 침묵했다. 완료착시에서 에이전트는 자기보고로 100% 완료를 주장했지만 실제는 90.8%였다. 세 연구가 같은 지점을 가리킨다. **안전 점검은 존재하지만 별도의 방에 있고, 기본 경로는 그 방을 지나가지 않는다.**

그래서 실무적 함의는 "더 똑똑한 모델"이 아니다. 판단은 이미 있으므로, 그것을 기본 경로에 배선하는 시스템 계층이면 된다. 임계값 하나짜리 게이트가 오류를 24.2%에서 4.5%로 낮춘 것이 그 증거다. 값은 커버리지로 치른다.

정직한 한계

- **파일럿이다.** 120문항, 주 분석은 단일 모델(gpt-4o-mini). 범주당 30개는 방향성이지 정밀 추정이 아니다.
- **두 번째 모델이 잘렸다.** gpt-4o는 API 크레딧 소진으로 101문항에서 멈췄고 하필 nonexistent가 11/30이라 주 분석에서 뺐다. 이는 계획된 배제가 아니라 사고이며, 그대로 밝힌다.
- **AUROC가 쉬운 분리 위에서 계산됐다.** 오답이 거의 전부 한 범주(nonexistent)에 몰려 있어, 두 신호 모두 실제보다 좋아 보일 수 있다. 범주가 섞인 실전에서는 낮아질 것이다.
- **자기보고는 순서만 맞다.** 0.999라는 AUROC는 순위 능력이지 교정이 아니다. 93% 거짓에 6.8/10은 여전히 과신이다.
- **OpenAI 한정.** 토큰 로그확률을 주는 API가 이 계열뿐이라 Claude·Gemini의 내부 신호는 측정하지 못했다.

- **토큰 엔트로피는 의미 단위가 아니다.** semantic entropy 계열이 더 강한 신호일 수 있어 우리 내부 신호 수치는 하한으로 읽어야 한다.
- **선행 연구.** 불확실성 기반 abstention·selective prediction·conformal 방법은 이미 활발한 분야다(semantic entropy, Nature 2024 등). 우리는 이를 처음 제안한다고 주장하지 않는다. 우리가 더한 것은 좁다: 같은 문항에서 내부 신호와 입 밖으로 나온 확신을 나란히 재고, 판단이 아니라 자동성이 빠졌음을 보인 것이다.

결론

모델은 지어낼 때 이미 의심하고 있었다. 내부 신호에도 남았고, 물으면 말로도 표현했다. 그럼에도 답에는 유보를 한 글자도 싣지 않고 93%를 단언했다. 없는 것은 판단력이 아니라, 그 판단이 묻지 않아도 작동하는 기본값이다. 모든 지능은 말할 수 있는 것보다 적게 말하는 방식으로 신뢰를 얻는데, 기계에는 아직 그 "적게 말하기"가 기본 경로로 배선되어 있지 않다. 배선은 임계값 하나로도 시작된다.

재현: `python -m src.run` (생성·확신·채점, 내용 캐시), `python -m src.analyze`, `python -m src.viz`, `python -m src.gifs`. 사전 설계는 `DESIGN.md`. 코드·데이터: https://github.com/hankimis/inhibition-layer.