2026 KAIST AI FUTURE CHALLENGE · 최우수상

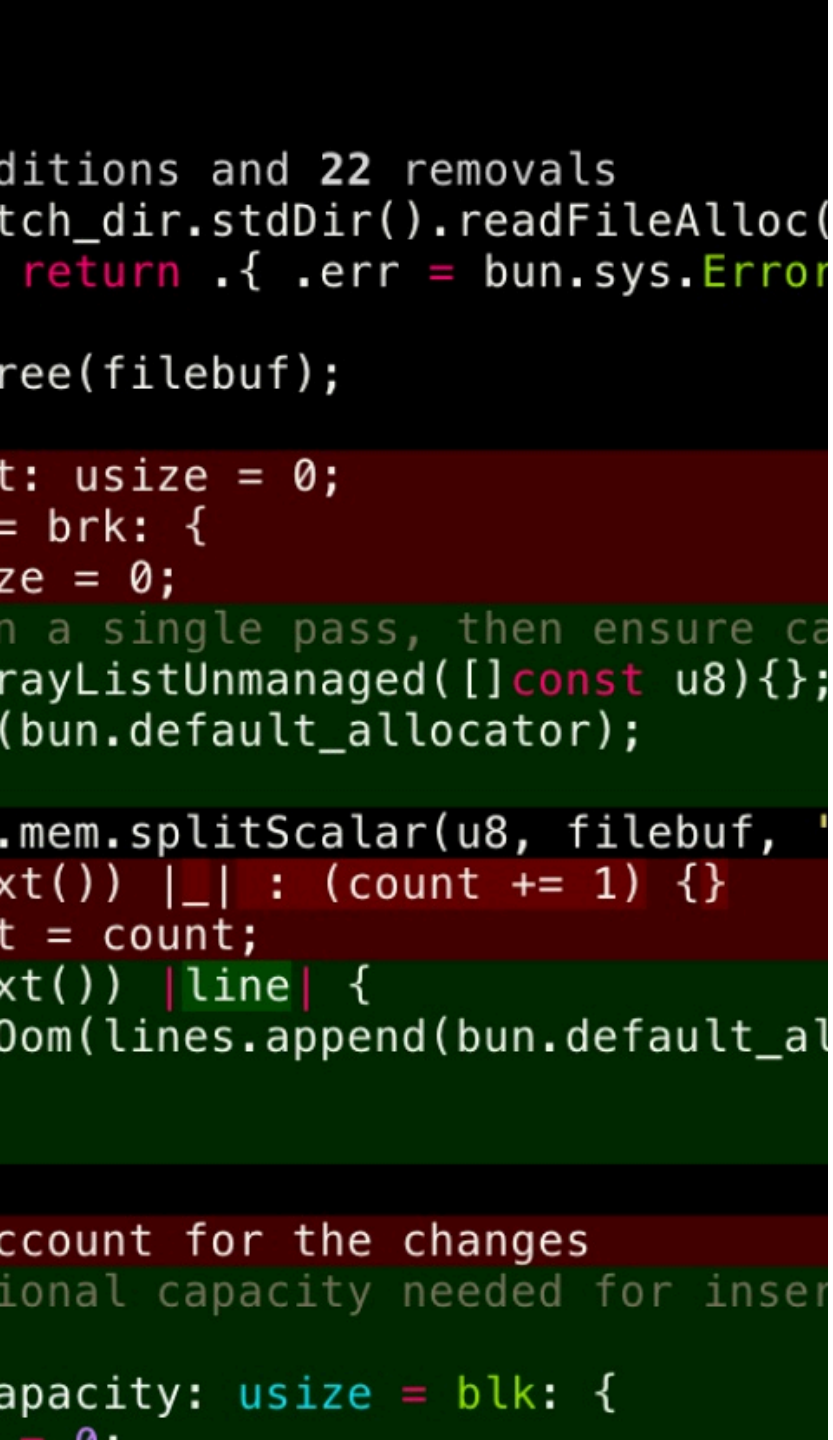
인지면역 AI

도울수록 사용자의 사고력이 자라도록 만드는 AI 설계 철학

발표자 오도열

Cognitive Immune AI

An AI design philosophy where the more it helps, the more the user thinks.

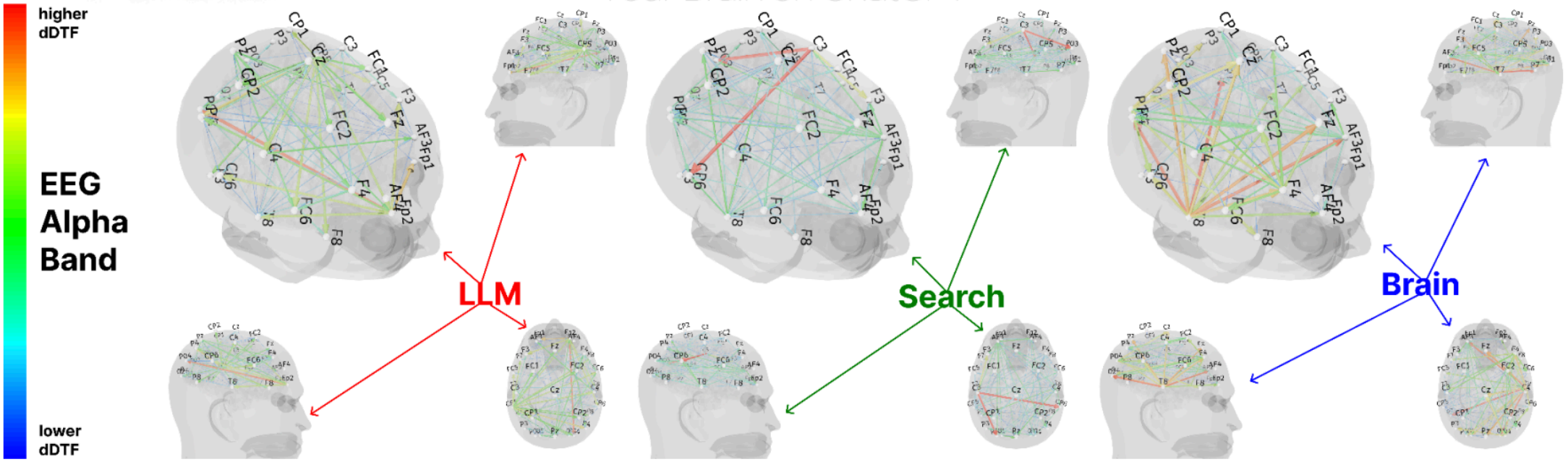


개발하며 AI에 지나치게 의존하는 저 자신을 본 순간, 이 아이디어가 시작됐습니다.

답은 받았습니다. 하지만 그 코드의 논리는 머리에 남지 않았습니다.

LLM의 영향, 뇌까지 도달한 첫 증거

> **Your Brain on ChatGPT: Accumulation of Cognitive Debt when Using an AI Assistant for Essay Writing Task**
>
> Nataliya Kos'myna et al. — MIT Media Lab (2025, arXiv:2506.08872)



Alpha 대역에서 측정한 **뇌 영역 간 정보 흐름의 강도.** LLM 그룹은 이 흐름이 가장 약했다 **(= 머리를 가장 적게 쓰며 글을 썼다).**

계산기와는 다르다

도구가 가져간 인지 기능의 범위

	가져간 것	사용자에게 남은 것
🧮 **계산기**	암산	문제 정의, 판단, 추론
🔍 **검색엔진**	기억	문제 정의, 판단, 추론
🤖 **오늘의 AI**	**암산 + 기억 + 문제 정의 + 비판적 판단 + 창의적 추론**	…? 🤔

학습 심리학의 두 가지 이론적 토대

01

Bjork (1994)

바람직한 어려움

Desirable Difficulty

너무 쉬우면 흘려보내고, 너무 어려우면 포기한다.
그 사이의 **적당한 도전**이 학습과 기억을 강화한다.

02

McGuire (1964)

예방접종 이론

Inoculation (비유적 차용)

약한 항원에 미리 노출되면 저항력이 자라듯,
작은 인지적 도전이 사고력을 단련한다.

→ **도전이 곧 면역이다.** 좋은 AI라면 우리를 약간 불편하게 만들 줄 알아야 한다.

기존 도구들의 공통된 한계

도구	정체	자동 작동	범용 작업
Khanmigo	답 대신 단계별 질문을 던지는 학습 AI	❌ 학습 모드만	❌ 교육 한정
Duolingo	게임 방식으로 외국어를 가르치는 앱	❌ 학습 모드만	❌ 언어 한정
Cursor Plan Mode	코드 작성 전 계획부터 세우게 하는 모드	❌ 수동으로 켜야	✅
Claude thinking	답하기 전 추론 과정을 보여주는 모드	❌ 수동으로 켜야	✅
인지면역 AI	**본 제안**	✅ **평소에 자동**	✅ **모든 작업**

해법 1 · 사용자 맞춤 강도

사용자 상태를 읽고, 강도를 맞춘다 — 네 가지 신호

01 **AI 부르기 전 직접 쓴 시간**
얼마나 스스로 시도했는가

02 **받은 답 그대로 쓰는 비율**
AI가 준 답을 수정 없이 받아쓰는가

03 **같은 힌트 반복 요청**
같은 종류의 도움을 계속 청하는가

04 **질문이 어려워지는 속도**
시간이 갈수록 더 어려운 질문을 던지는가

→ RLHF 보상에 **'인지 참여' 항**을 더해, 힌트와 반론을 배합

→ 응급·접근성 상황은 **곧바로 답하는 모드**로 자동 전환

→ 네 신호는 **서로 교차 검증 + 결과물 품질**로 신뢰성 확보

해법 2 · Nudge 기반 자율성

"자동 작동"의 실체 = 도전 모드를 기본값으로 두는 것

도구	기본값		실제 사용
Cursor / Claude	일반 모드	→	대부분 일반 모드 (켜기 귀찮음)
인지면역 AI	**도전 모드**	→	**대부분 도전 모드 + 한 번이면 일반 모드로**

차이는 단 하나 — **기본값**.
그 하나가 강제 없이 자동 작동을 만든다.

행동경제학의 **기본값 효과** (Default Effect) — Thaler, *Nudge* (2008)

해법 3 · 인지독립률 (CII, Cognitive Independence Index)

네 가지 신호를 하나의 점수로

01 AI 부르기 전 시간
8.2s

02 그대로 쓰는 비율
31%

03 같은 힌트 반복
0.4

04 질문 어려워지는 속도
+18%

→

$$CII = 0.30 \cdot \log(\text{편집시간}) + 0.25 \cdot (1 - \text{수락률}) + 0.25 \cdot (1 - \text{힌트반복}) + 0.20 \cdot \Delta\text{복잡도}$$

* 가중치는 추후 실험을 통해 개선

→

CII

72 / 100

▲ 이번 주 +5

"AI 사용의 BMI"

사용량이 아니라, **사용의 질**을 잰다

어떻게 실현하는가

새 모델은 필요 없다. 기존 LLM 위에 한 층의 미들웨어를 얹으면 된다.

구성 요소		구현 방식
⏱️ 📋	**그대로 받는 신호** (01 편집 시간, 02 수락률)	IDE / 브라우저에서 이벤트로 바로 수집
💡 📈	**해석이 필요한 신호** (03 반복 힌트, 04 질문 난이도)	별도 분석 — 비슷한 질문끼리 묶거나, 저비용 LLM으로 평가
🧠	**CII 계산**	4개 신호 가중평균을 사용자 화면에 실시간 표시
🎯	**인지 참여 보상**	모델 재훈련 없이 프롬프트만으로 먼저 검증 → 효과 확인되면 RLHF에 반영
🚀	**배포**	VS Code Extension · Chrome Extension · LLM Wrapper

검증 경로: 사용자를 두 그룹으로 나눠 A/B 테스트 → 동일 과제에서 **CII 변화량 비교**

AI를
인간 대체 도구가 아니라
인간 성장의 환경으로.

때로는 좋은 도구의 기준이,
사용자를 얼마나 잘 대신해주는가가 아니라
사용자를 얼마나 성장시키는가일지도 모릅니다.

감사합니다.