AI는 어떻게 '잘' 생각하게 되었는가?

LLM Reasoning Models

Let's Verify Step by Step | Scaling Test-Time Compute | DeepSeek-R1

20200661 고현민 | 20240371 지은수

PART 1

Introduction

오늘의 질문

"ChatGPT에게 수학 문제를 풀어달라고 했더니 틀렸다. 왜일까?"

Q1	AI가 틀릴 때, 어디서 틀리는지 알 수 있을까?	→ PRM
Q2	AI에게 "더 오래 생각할 시간"을 주면 더 잘 풀까?	→ Test-Time Scale
Q3	인간 없이 AI 스스로 추론을 발전시킬 수 있을까?	→ DeepSeek-R1

발표 흐름

01 논문 1

Let's Verify Step by Step

좋은 선생님 만들기

"과정을 봐야 잘 가르친다"

→

02 논문 2

Scaling Test-Time Compute Optimally

시험장 전략

"문제 난이도에 맞게 생각하라"

→

03 논문 3

DeepSeek-R1

스스로 깨닫는 AI

"인간 없이도 추론이 자란다"

AI를 더 잘 생각하게 만드는 여정 : 외부 피드백 설계 → 추론 시간 활용 → 내부 자가 진화

LLM이란?

"LLM = 인터넷 전체를 읽은 뒤 인간처럼 말하는 법을 배운 시스템"

01
자연어 이해
인간의 언어를 문맥과 함께 파악

02
맥락 파악
긴 대화와 문서에서 흐름 유지

03
텍스트 생성
주어진 입력에 맞는 답변 생성

입력 텍스트 → Transformer 처리 → 출력 텍스트

추론(Reasoning)이 왜 어려운가?

핵심 문제

환각 (Hallucination)

AI가 모르는 걸 아는 척 지어내는 현상. 다단계 추론에서 치명적

오류 누적

1단계가 틀리면 이후 모든 단계가 무너짐.

오류 누적 예시

Q: 27 × 38 = ?

1단계: 20×38 = 740 ❌ (실제: 760)

2단계: 7×38 = 266 ✓

최종: 740+266 = 1006 ❌ (실제: 1026)

→ 1단계 오류 하나가 전체를 망침

그래서 AI의 '추론 과정'을 어떻게 감독하고 개선할지가 핵심 연구 과제가 됨

오늘 논문 세 편의 포지셔닝

	PRM	Test-Time Scale	DeepSeek-R1
접근법	더 나은 피드백 설계	추론 시간 연산 최적화	강화학습으로 자가 진화
핵심 질문	과정을 보면 더 잘 가르칠 수 있을까?	생각할 시간을 주면 더 잘 풀까?	혼자서도 추론이 발전할 수 있을까?
비유	과정 채점하는 선생님	시험 시간 전략	자습으로 깨닫는 학생

시간 순으로도, 기술적으로도 서로 위에 쌓아 올린 연구들

PART 2

Let's Verify Step by Step

"선생님이 채점하는 방법이 바뀌면 학생이 달라진다"

Let's Verify Step by Step

저자	데이터셋	공개 데이터
OpenAI 연구팀 (Lightman et al., 2023)	MATH (고등학교~경시대회 수준)	PRM800K (80만 개 단계별 레이블)

핵심 질문

"AI를 훈련할 때, 최종 답만 볼까? 아니면 풀이 과정도 볼까?"

왜 이 논문이 중요한가?

이후 모든 추론 AI (OpenAI o1, DeepSeek-R1 등)의 설계 철학에 영향을 준 논문

ORM vs PRM — 직관적 비유

ORM | 결과 감독 보상 모델

답만 보는 선생님

수학 시험에서 최종 답만 보고
⭕/❌ 표시하는 선생님

자동화 쉬움

잘못된 과정도 답이 맞으면 보상

VS

PRM | 과정 감독 보상 모델

과정을 보는 선생님

풀이 과정 한 줄 한 줄 읽고
어디서 틀렸는지 짚어주는 선생님

정밀한 피드백

인간 레이블링 필요 — 비용 발생

ORM의 치명적 문제 — False Positive

False Positive(거짓 양성): 틀린 추론 과정을 거쳐 운 좋게 정답에 도달한 풀이 → ORM은 이걸 "좋은 풀이"로 학습

예시 비교

잘못된 풀이	거짓 양성 풀이
2/4 + 1/3 = 3/7	1/2 + (- 1/2) = 0/4 = 0
→ 답: 3/7 (오답)	→ 답: 0 (우연히 정답)
ORM 평가: ❌ (답이 틀렸기 때문)	ORM 평가: ✓ (답이 맞았기 때문)

결론: 답만 보면 AI가 요령 피우는 법을 배운다

PRM의 작동 방식

레이블과 확률의 관계:

인간이 각 단계에 Positive/Negative 레이블을 붙인 데이터로 PRM(신경망)을 학습시키면,
PRM은 새로운 단계를 보고 "이 단계가 맞을 확률"을 0~1 숫자로 예측하게 됩니다.
레이블은 학습 재료, 확률은 학습된 모델의 출력입니다.

① 인간이 붙이는 레이블 (학습 데이터)

Positive ✓	정확하고 합리적인 단계
Negative ✗	부정확하거나 불합리한 단계
Neutral ?	모호한 단계 (대체로 Positive 처리)

② 학습된 PRM의 확률 출력 → 점수 계산

전체 풀이 점수 = 각 단계 확률의 곱

예: 0.9 × 0.8 × 0.3 = 0.216

→ 3단계에서 오류 가능성이 높음
→ 해석: "모든 단계가 다 옳을 확률"

단계별 평가 흐름 (첫 Negative 단계 이후는 채점 불필요)



데이터 수집 — PRM800K

75,000
개 풀이

800,000
개 단계별 레이블

12,000
개 MATH 문제

능동 학습 (Active Learning) 전략 — 어떤 풀이를 먼저 레이블링할지 선택하는 방법

무작위 레이블링 (기존 방식)

— 전체 풀이 중 무작위로 선택해 레이블 부탁

— PRM이 이미 잘 아는 쉬운 사례에 레이블 낭비

— 약점 개선이 느림

능동 학습 (이 논문의 방식)

— 현재 PRM이 높은 점수를 줬지만 실제론 틀린 풀이를 골라서 레이블 요청

— PRM이 속고 있는 케이스만 집중 공략

— 같은 데이터량으로 더 빠른 성능 향상

효과: 동일한 성능 달성에 필요한 데이터량 2.6배 감소 — PRM의 약점만 골라 가르치기 때문

실험 결과 — 숫자로 보는 차이

	ORM	PRM	Majority Voting
% Solved (Best-of-1860)	72.4	**78.2**	69.6

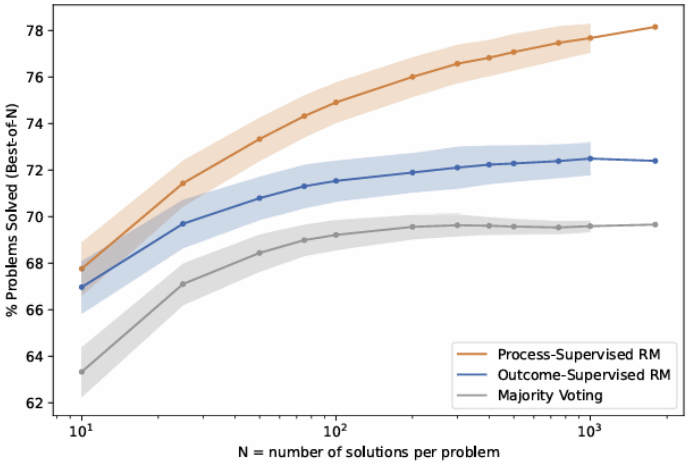


Figure 3: A comparison of outcome-supervised and process-supervised reward models, evaluated by their ability to search over many test solutions. Majority voting is shown as a strong baseline. For $N \leq 1000$, we visualize the variance across many subsamples of the 1860 solutions we generated in total per problem.

78.2% PRM 최고 성능 (MATH 테스트셋)

+5.8%p PRM vs ORM 성능 차이

2.6x 능동 학습 데이터 효율

N(샘플 수)이 커질수록 PRM의 우위가 더 벌어짐 — PRM은 "많이 생각할수록" 더 빛을 발함

OOD 일반화 — 처음 보는 문제에서도 통한다!

실험 설정: 훈련 데이터 수집 이후 출제된 AP 물리 · AP 미적분 · AP 화학 · AMC10/12 시험 문제 234개

	ORM	PRM	Majority Vote	# Problems
AP Calculus	68.9%	**86.7%**	80.0%	45
AP Chemistry	68.9%	**80.0%**	71.7%	60
AP Physics	77.8%	**86.7%**	82.2%	45
AMC10/12	49.1%	**53.2%**	32.8%	84
Aggregate	63.8%	**72.9%**	61.3%	234

Table 1: We measure out-of-distribution generalization using recent STEM tests. We evaluate the outcome-supervised RM, the process-supervised RM, and majority voting using 100 test samples per problem.

핵심 결론

PRM: 72.9%

ORM: 63.8%

Majority Vote: 61.3%

과정을 학습한 PRM은
새로운 유형에도 강함

PRM의 강력한 성능이 분포 변화에도 유지됨 — Process Learning의 일반화 능력 증명

AI 정렬(Alignment) 이야기

AI 정렬 (Alignment)

AI의 행동이 인간의
의도·가치관과 일치하도록 만드는 것

정렬 세금 (Alignment Tax)

AI를 안전하게 만들수록
성능이 떨어진다는 기존 통념

음의 정렬 세금

안전하게 만들었더니 오히려
성능도 올라가는 반전 케이스
(PRM이 바로 여기 해당)

일반적인 통념

"더 안전하게 만들면 성능이 떨어진다"
안전성 ↑ → 성능 ↓ (트레이드오프)

BUT
PRM은?

PRM의 반전 — 음의 정렬 세금

인간이 승인한 추론 과정 직접 보상
→ 더 안전 + 성능도 ORM보다 높음
→ 안전성 ↑ 성능도 ↑ (트레이드오프 없음)

ORM이 안전하지 않은 이유 — Reward Hacking:

결과만 보상하면 AI가 채점 허점을 이용해 실제 능력 없이 점수만 높이는 편법을 학습함. PRM은 과정 자체를 보상하므로 허점이 없음.

PRM — 수학을 넘어 현실 세계로

의료 AI

Med-PRM (2025) — 의료 진단 단계 검증

- ✓ 임상 가이드라인 기준 단계별 검증
- ✓ 5개 의료 QA 벤치마크 최대 +13.5%p
- ✓ 약물 금기·감별진단 오류 단계별 포착

금융 추론

Fin-PRM (2025) — 금융 추론 감시

- ✓ 금융 지식 + 규제 검토 통합 PRM
- ✓ SFT +12.9% / RL +5.2%
- ✓ 펀드 전략·리스크 분석 중간 오류 추적

코딩 에이전트

SWE (2025) — 에이전트 궤도 교정

- ✓ 코드 생성·수정 시 PRM이 실시간 비평가
- ✓ GUI·SW 엔지니어링 에이전트 적용
- ✓ 중간 단계 오류 발견 → 즉시 궤도 수정

수학 풀이 과정을 감독하던 PRM이, 이제는 의료·금융·코딩 에이전트의 추론 과정을 단계마다 감시하고 있다

출처: Med-PRM (Yun et al., EMNLP 2025) · MedPRMBench (Wu et al., 2025)

논문 요약

① 수학 추론에서 PRM > ORM 실증 — 최고 성능 PRM 78.2%

② 대규모 PRM이 소규모 모델의 인간 감독을 신뢰할 수 있게 대체 가능

③ 능동 학습으로 데이터 효율 2.6배 ↑

④ PRM800K 전체 공개 — 관련 연구 진입 장벽 대폭 낮춤

좋은 검증자(PRM)를 만들었다. → 이것을 활용해 AI가 '추론 시간'을 더 잘 쓰게 만들 수 있을까? → Test-Time Scale

PART 3

Scaling LLM Test-Time Compute Optimally

"시험장에서 시간을 어떻게 쓰느냐가 성적을 결정한다"

Scaling LLM Test-Time Compute Optimally

저자	사용 모델	벤치마크
Google DeepMind (Snell et al., 2024)	PaLM 2-S* (Codey) 현재 LLM 능력 대표	MATH (고등학교~경시대회)

"AI에게 더 오래 생각할 시간을 주면 진짜로 더 잘 풀까? 어떻게 줘야 가장 효율적일까?"

→ PRM이 이 논문에서 핵심 검증 도구로 등장

→ 단순히 연산을 더 쓰는 게 아니라 → 문제 난이도에 따라 전략을 적응적으로 배분

→ 이 논문의 아이디어가 OpenAI o1, DeepSeek-R1의 직접적인 설계 원리가 됨

직관적 동기 — 인간도 그렇게 한다

"여러분은 쉬운 문제와 어려운 문제에 같은 시간을 쓰나요?"

인간의 전략

— 쉬운 문제 → 빠르게 풀고 넘김

— 어려운 문제 → 여러 방법 시도, 검토 반복

기존 AI 방식

— 모든 문제에 동일한 방식으로 한 번 답 생성

— 비효율, 어려운 문제 실패율 높음

이 논문의 제안

— 문제 난이도에 따라 생각하는 방식을 바꿔야 한다

— Compute-Optimal Strategy

핵심 주장: 문제 난이도에 따라 '생각하는 방식'을 바꿔야 한다

두 가지 전략 프레임

전략 A — Verifier(검증자) 활용

여러 개의 답을 병렬로 생성한 뒤
PRM이 가장 좋은 것을 선택

비유:

시험에서 여러 풀이를 써보고
제일 자신 있는 것을 제출

"많이 낚아서 좋은 것 고르기"

VS

전략 B — Revision(자기수정)

이전 답을 보고
순차적으로 고쳐나감

비유:

초안을 쓰고 퇴고를 반복해서
최종 답 제출

"한 답을 점점 다듬기"

"이 논문은 두 방향을 모두 실험해 비교한다"

Revision — 실제 자가수정의 과정

If $f(x) = \frac{3x-2}{x-2}$, what is the value of $f(-2) +f(-1)+f(0)$? Express your answer as a common fraction.

Attempt 1:

```
Let's evaluate $f(-2)$.
We have $f(-2)=\frac{3(-2)-2}{-2-2}=-\frac{4}{-4}=\frac{1}{1}=1$.
So $f(-2)=1$.
Let's evaluate $f(-1)$.
We have $f(-1)=\frac{3(-1)-2}{-1-2}=-\frac{5}{-3}=\frac{5}{3}$.
So $f(-1)=\frac{5}{3}$.
Let's evaluate $f(0)$.
We have $f(0)=\frac{3(0)-2}{0-2}=\frac{-2}{-2}=1$.
So $f(0)=1$.
So we want to compute
$1+\frac{5}{3}+1=\frac{3}{3}+\frac{5}{3}=\frac{3+5}{3}=\frac{8}{3}$.
So the answer is $\boxed{\frac{8}{3}}$.
#### \frac{8}{3}
```

Attempt 2:

```
Let's first find $f(-2)$.
$f(-2) = \frac{3(-2)-2}{-2-2} = \frac{-6-2}{-4} = \frac{-8}{-4} = 2$.
Now let's find $f(-1)$.
$f(-1) = \frac{3(-1)-2}{-1-2} = \frac{-3-2}{-3} = \frac{-5}{-3} = \frac{5}{3}$.
Now let's find $f(0)$.
$f(0) = \frac{3(0)-2}{0-2} = \frac{-2}{-2} = 1$.
So we have $f(-2) +f(-1)+f(0) = 2+\frac{5}{3}+1 =
\frac{7}{3}+\frac{5}{3}+\frac{3}{3} = \frac{7+5+3}{3} = \frac{15}{3} = 5$.
So the answer is $\boxed{5}$.
#### 5
```

Attempt 3:

```
Let's plug in $-2$, $-1$, and $0$ for $x$ in $f(x)$.
We get $f(-2) = \frac{3(-2)-2}{-2-2} = \frac{-8}{-4} = \frac{2}{1} = 2, f(-1) =
\frac{3(-1)-2}{-1-2} = \frac{-5}{-3} = \frac{5}{3}, f(0) = \frac{3(0)-2}{0-2} =
\frac{-2}{-2} = 1$.
So we have $2+\frac{5}{3}+1=\frac{6}{3}+\frac{5}{3}+\frac{3}{3}=\frac{14}{3}$.
So the answer is $\boxed{\frac{14}{3}}$.
#### \frac{14}{3}
```

Figure 17 | Revision model example 1. The model calculates the sum at the end incorrectly on the first two attempts, but on the third attempt it succeeds and gets the answer correct.

전략 B — Revision(자기수정)

이전 답을 보고
순차적으로 고쳐나감

비유:

초안을 쓰고 퇴고를 반복해서
최종 답 제출

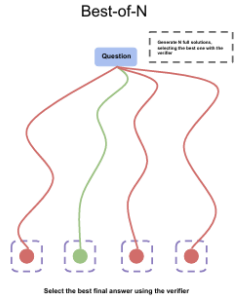
"한 답을 점점 다듬기"

Verifier 기반 탐색 3종 비교

① Best-of-N (가장 단순)

N개의 답을 독립적으로 생성
PRM이 최고 점수 답 선택

비유: 시험지 N장 각자 풀고
가장 잘 쓴 것 제출

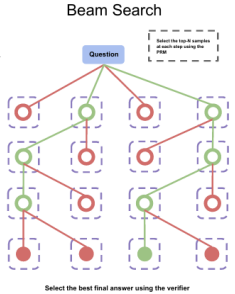


효율적

② Beam Search (단계별 탐색)

각 단계마다 N개 후보 생성
PRM 점수 기준 상위만 유지
나머지 버림

비유: 오디션에서 매 라운드마다
탈락자를 걸러내며 진행

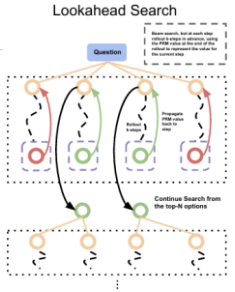


중간 난이도에
효과적

③ Lookahead Search (미래 시뮬레이션)

각 단계에서 k단계 앞을
미리 시뮬레이션한 뒤
현재 결정의 가치 추정

비유: 체스에서 몇 수 뒤를
머릿속으로 미리 계산



비용 > 효과
→ 비추천

Lookahead Search는 '해봤는데 비용 대비 효과 없음'을 보여주기 위한 비교 실험. 실제로는 Best-of-N / Beam Search만 사용.

난이도별 최적 전략 — 핵심 발견

문제 난이도	연산 예산 적을 때	연산 예산 클 때
쉬운 문제 (1~2단계)	Best-of-N	Best-of-N
중간 문제 (3~4단계)	Beam Search	Best-of-N
어려운 문제 (5단계)	⚠ 효과 제한적 (한계)	⚠ 효과 제한적 (한계)

⚠ 효과 제한적 (한계) 의 의미: 가장 어려운 문제(5단계)는 Best-of-N, Beam Search 어떤 방법을 써도, 연산을 아무리 늘려도 정확도가 유의미하게 오르지 않음. 추론 시간을 늘리는 것만으로는 해결 안 되는 문제들이 있다는 이 논문의 솔직한 한계 인정.

쉬운 문제에서 Beam Search가 역효과인 이유: PRM 점수를 과도하게 최적화 → Over-optimization 발생 (엉뚱한 특징을 강화)

Compute-Optimal 핵심 성과: 난이도 기반 최적 전략 적용 → Best-of-N 대비 4배 적은 연산으로 동등한 성능

Revision 모델 — 스스로 고치는 AI

기존 방식의 문제점

일반 LLM에게 "네 답을 고쳐봐"라고 프롬프트만 줘도 → 효과 거의 없음 → 진짜로 수정하는 법을 별도로 학습시켜야 함

훈련 데이터 생성 방법 (4단계)

1. 각 문제에 대해 64개의 답 병렬 생성
2. 문자 수준 편집 거리 기준으로 오답 ↔ 정답 페어링
3. "오답 → 오답 → ... → 정답" 형태의 수정 궤적 구성
4. 이 궤적으로 SFT (Supervised Fine-Tuning) 수행

편집 거리란? 두 텍스트 사이의 글자 수준 차이량. ex) '답: 7/12'와 '답: 5/12'는 숫자 하나만 다르므로 편집 거리가 짧음 → 페어링 대상.
'답: 7/12'와 완전히 다른 풀이 방식의 정답은 편집 거리가 김 → 페어링 제외.

비슷한 것끼리 짝 지어야 '새로 시도하기'가 아닌 진짜 '수정하기'를 학습할 수 있음

Sequential vs Parallel — 상보적 관계

Sequential (순차 수정) — Local Refinement

- 이전 답을 보고 다음 답 생성
- 국소적 개선에 강함
- 방향이 맞는 답을 정교하게 다듬기



Parallel (병렬 생성) — Global Search

- 독립적으로 N개 동시 생성
- 다양한 접근법 탐색에 강함
- 완전히 다른 풀이 전략 발견 가능



Sequential의 38% 문제란?

이전 수정 단계에서 올바른 답을 생성했음에도, 다음 수정 시 그것을 틀렸다고 잘못 판단하고 엉뚱하게 바꿔버리는 경우가 38% → 단독 사용이 불안정한 이유!

전반적으로 Sequential이 Parallel보다 약간 우수하지만 단독 사용 시 38% 역방향 오류 — 불안정

쉬운 문제: Sequential 100% / 어려운 문제: Sequential + Parallel 혼합이 최적 → 두 방법은 상호 보완적

Compute-Optimal의 핵심 성과

4×
적은 연산

Verifier 기반
난이도별 최적 전략 적용 시
Best-of-N 대비 4배 적은 연산으로 동등한 성능
(64개 생성 → 16개만으로 가능)

4×
더 적은 연산

Revision 기반
난이도별 최적 비율 적용 시
Best-of-N 대비 4배 적은 연산으로 더 높은 성능
(256개 생성 → 64개만으로 가능)

14×
큰 모델 대체

작은 모델 + Test-Time
작은 모델에 Compute-Optimal
Test-Time Scaling 적용 시
14배 큰 모델의 성능 능가 (쉬운~중간 문제)

작은 모델 + 똑똑한 추론 전략 = 큰 모델보다 효율적

Test-Time Compute — 이론에서 제품으로

IOI 2024

국제 정보 올림피아드 금메달 (OpenAI, 2025)

- ✓ o1-ioi: 1만 개 후보 → 클러스터링 + 재순위
- ✓ Best-of-N + Verifier 전략의 실전 적용
- ✓ o3: 휴리스틱 없이 50회 제출만으로 금메달

OpenAI 제품

o3-mini '생각 강도' 3단계 선택

- ✓ API: reasoning_effort = low / medium / high
- ✓ low = 최고속, high = 최대 컴퓨트로 최선의 답
- ✓ 난이도별 전략 차등화 원리의 직접 구현

ARC-AGI

인간 기준선 돌파 (o3, 2024)

- ✓ o3: ARC-AGI 88% (인간 기준선 85% 초과)
- ✓ o1은 동일 조건에서 32% 그침
- ✓ 고득점 버전은 문제당 약 $1,000 컴퓨트 사용
- ✓ '더 생각할수록 더 잘 푼다'의 극적 증명

논문의 Best-of-N / Verifier 전략이 실제 국제 올림피아드 금메달과 상용 AI 제품 설계에 직접 반영되었다

출처: OpenAI, Competitive Programming with Large Reasoning Models (arXiv 2502.06807, 2025)

논문 요약

① 무식하게 많이 생각하는 게 아니라, 문제에 맞게 전략을 골라야 한다

② Verifier(선택)와 Revision(수정)은 서로 상호 보완적이다 — 섞어 쓰는 게 최적

③ 작은 모델도 똑똑하게 추론하면 훨씬 큰 모델을 이길 수 있다

상호보완적이라는 말의 의미

Verifier는 다양한 접근법을 넓게 탐색하는 데, Revision은 방향이 맞는 답을 정교하게 다듬는 데 강함
예를 들어 어려운 문제는 먼저 Parallel로 여러 초안을 생성(Verifier)한 뒤, 좋은 것을 Sequential로 다듬는 (Revision) 방식으로 혼합해 쓰면 각각의 단점을 보완할 수 있음

이런 '더 잘 생각하는 능력'을 AI가 처음부터 스스로 터득하게 만들 수는 없을까? 인간의 레이블 없이도? → DeepSeek-R1

PART 4

DeepSeek-R1

"선생님 없이 스스로 깨닫는 AI"

DeepSeek-R1

저자	기반 모델	핵심 알고리즘
DeepSeek AI (중국, 2025)	DeepSeek-V3-Base (오픈소스)	GRPO (Group Relative PO)

"인간 레이블 없이, 순수 강화학습만으로 세계 최고 수준의 추론 AI를 만들 수 있을까?"

→ 기존 상식: "좋은 AI = 방대한 인간 레이블 데이터 필요"

→ DeepSeek-R1: "보상 신호만 있으면 AI가 스스로 추론 전략을 발견한다"

→ MIT 라이선스 오픈소스 공개 — 상용 최고 모델과 동등한 성능을 누구나 무료로

기존 방식의 한계 — 왜 새로운 접근이 필요했나

CoT + SFT 방식의 한계

— 인간이 만든 추론 패턴의 한계에 갇힘

— 인간보다 뛰어난 전략 발견 불가능

— 대규모 레이블 데이터 → 비용·편향 문제

DeepSeek의 가설

— "인간의 추론 예시가 오히려 모델의 잠재력을 가두는 건 아닐까?"

→ 정답/오답 보상만으로 AI가 스스로 최선의 추론 전략을 찾게 하자!

해결책

인간 레이블 없이 순수 강화학습으로 → AI가 스스로 최선의 추론 전략을 발견하도록

GRPO 알고리즘 — 효율적인 강화학습

강화학습 한 줄 요약: 정답을 맞히면 칭찬(보상), 틀리면 벌칙 → 칭찬받는 행동을 더 하게 됨

기존 PPO

— Value Model(가치 모델) 별도 필요

가치 모델 : '앞으로 얼마나 보상받을지' 예측하는 신경망
→ 정책 모델만큼 커서 메모리 2배

— KL 패널티를 토큰별로 따로 계산

KL : 모델이 학습 중 원래 모델과 너무 달라지지 않도록 제어하는 제동 장치

GRPO (개선)

— Value Model 불필요 → 비용 대폭 절감

같은 문제에 여러 답 생성 후 그룹 내 상대 비교로 '좋고 나쁨' 판단

— KL 발산을 손실 함수에 직접 추가 (별도 모델 없이 처리)

더 간단한 구조로 유사하거나 더 높은 성능 달성

Advantage 계산 수식: $A_i = (r_i - \text{그룹 평균 보상}) / \text{그룹 표준편차}$

→ 이 답이 같은 문제의 다른 답들보다 얼마나 더 좋은가? → Value Model 없이도 상대 비교로 품질 판단 가능

보상 설계 — 단순하지만 강력하게

① **정확도 보상 (Accuracy Reward)**

— 수학: 모델의 최종 답을 정답 DB와 문자열 비교 → 일치하면 +1, 불일치하면 0

— 코딩: 생성된 코드를 실제 실행 → 미리 준비된 테스트케이스 통과하면 +1

→ 인간 개입 없이 자동으로 계산 가능 (확인 가능한 도메인에서만 적용)

② **형식 보상 (Format Reward)**

— <think>...</think> 태그가 출력에 존재하는지 규칙으로 확인 → 있으면 보상

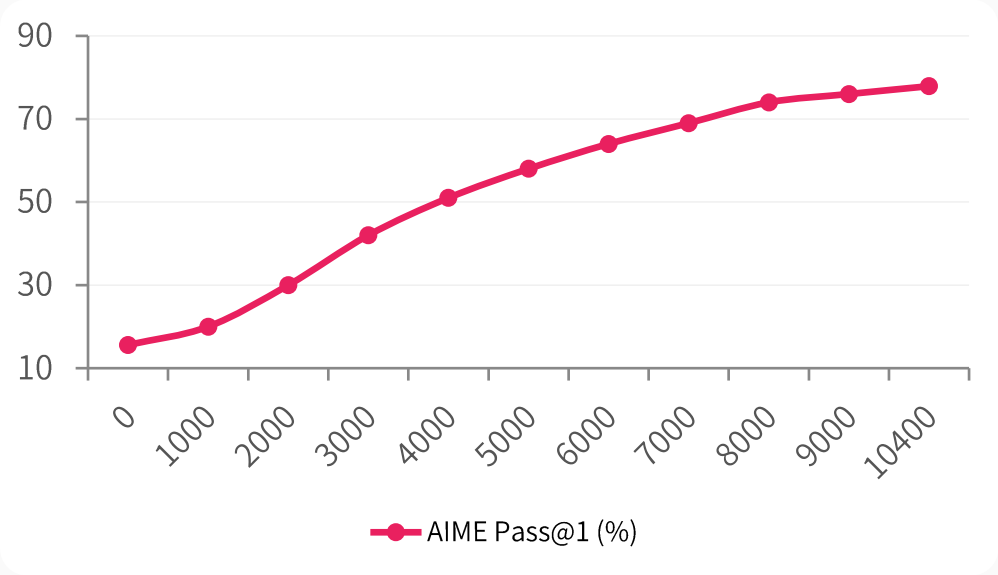
— 추론 과정을 답변과 분리해야 분석·해석이 가능하기 때문에 강제

→ 이 태그 안에서 AI가 자유롭게 사고 과정을 전개하도록 유도

신경망 보상 모델을 배제한 이유 — Reward Hacking 방지:

신경망 보상 모델은 자체적으로 허점이 있어 AI가 점수만 높이는 편법을 학습할 수 있음.
규칙 기반 보상(정답 일치, 태그 존재)은 명확한 기준이라 허점이 없고 진짜 능력만 보상됨.

Aha Moment



15.6% → 77.9%

AIME 2024 Pass@1 성장

"Aha Moment"

"Wait, wait. Wait.
내가 방금 틀렸다. 다시 생각해보면..."

반성적 사고 단어 사용 5~7배 증가

아무도 "이렇게 반성해라"고 가르친 적이 없음 → 정답/오답 보상만으로 AI가 스스로 메타인지를 발전시킴

R1-Zero의 한계 — 완벽하지 않다

언어 혼용	한 문장 안에 중국어·영어가 뒤섞임 (학습 데이터 분포 반영)
가독성 저하	추론 과정이 인간이 읽기 어려운 형태로 생성됨
범용성 부족	수학·코딩 외 작문, 오픈도메인 QA에서 성능 미흡

이것이 문제인 이유

— 추론 능력은 뛰어나지만 실제 사용자에게 배포하기 어려움

— 해석 불가능한 추론 과정은 AI 정렬 관점에서 신뢰하기 어려움

DeepSeek-R1: 4단계 파이프라인

Stage 1	Stage 2	Stage 3	Stage 4
Cold Start SFT	**1차 RL 훈련**	**2차 SFT**	**2차 RL 훈련**
고품질 Long CoT 수천 개 지도 학습	수학·코딩 집중 GRPO + 언어 일관성 보상 추가	추론 데이터 60만 + 일반 데이터 20만 = 80만 개 재학습	추론 + 유용성 + 안전성 보상 통합
목적 ✓ RL 시작 전 '읽을 수 있는 형태로 추론하는 버릇' 심기 ✓ 언어 혼용·가독성 문제 예방	목적 ✓ 추론 능력 극대화 + 언어 혼용 문제 해결	목적 ✓ 수학만 잘하는 AI → 글쓰기 / QA 등 뭐든 잘하는 AI	목적 ✓ 실제 사용자가 믿고 쓸 수 있는 최종 모델 완성

SFT(지도 학습)로 형식을 잡고 → RL로 능력을 키우고 → 다시 SFT로 범용성 추가 → 최종 RL로 안전성 통합

최종 성능 비교 — 세계 최고와 어깨를 나란히

벤치마크	DeepSeek-R1	OpenAI o1	GPT-4o
AIME 2024 (Pass@1)	**79.8% ★**	79.2%	9.3%
MATH-500	**97.3% ★**	96.4%	74.6%
Codeforces (Pct.)	**96.3% ★**	96.6%	23.6%
GPQA Diamond	71.5%	75.7% ★	49.9%
MMLU	90.8%	91.8% ★	87.2%

지식 증류(Distillation) — AI 민주화

지식 증류란? 큰 모델(교사)이 가진 지식을 작은 모델(학생)에게 옮기는 기법. DeepSeek-R1이 생성한 고품질 추론 데이터를 소형 모델에 SFT로 학습시켜, 소형 모델이 대형 모델의 추론 패턴을 흉내내도록 만듦. 소형 모델이 RL을 처음부터 하는 것보다 훨씬 효율적.

모델	AIME 2024	MATH-500
GPT-4o (비교군)	9.3%	74.6%
R1-Distill-Qwen-1.5B	**28.9%**	**83.9%**
R1-Distill-Qwen-7B	**55.5%**	**92.8%**
R1-Distill-Qwen-32B	**72.6%**	**94.3%**
R1-Distill-Llama-70B	**70.0%**	**94.5%**

15억 파라미터 초소형 모델(1.5B)이 수학에서 GPT-4o를 능가 — 강력한 AI의 능력을 저비용으로 작은 모델에 이식하는 시대

DeepSeek-R1 — 오픈소스가 세계를 흔들다

앱스토어 1위

ChatGPT 제치고 미국 앱스토어 1위

- ✓ 2025.1.20 출시 → 7일 만에 美 iOS 1위
- ✓ 엔비디아 주가 18% 폭락, 시총 ~600조 원 증발
- ✓ MIT 라이선스 무료 오픈소스

$6M의 기적

GPT-4 비용의 6%로 동급 성능

- ✓ 훈련 비용 약 600만 달러
- ✓ GPT-4 추정 1억 달러 — 약 17배 차이
- ✓ 인간 레이블 비용 줄인 GRPO 방식이 핵심
- ✓ AI 비용 패러다임 붕괴

기업 배포

Microsoft Azure·GitHub 기업 API 채택

- ✓ Azure·GitHub에서 오픈소스 추론 엔진으로 상업 배포
- ✓ 안정성·비용 효율성 인정
- ✓ 140개국 앱스토어 상위권 진입

인간 레이블 없이 순수 강화학습으로 만든 오픈소스 AI가 상용 최고 모델과 동등한 성능을 달성하고 산업 지형을 바꿨다

출처: DeepSeek-AI (2025), DeepSeek-R1 Technical Report

PART 5

종합 마무리 — 연결 고리와 빅픽처

세 편의 논문이 어떻게 하나의 큰 흐름이 되는가

3편 논문 연결 고리 — 퍼즐 맞추기

PRM

Let's Verify Step by Step

과정을 채점하는
보상 모델(PRM) 설계
PRM800K 공개

→

Test-Time Scale

Scaling Test-Time Compute

PRM을 검증자로 활용해
→ 최적의 탐색 전략 설계
Compute-Optimal 프레임워크

→

DeepSeek-R1

DeepSeek-R1

GRPO + 규칙 기반 보상
→ Aha Moment
4단계 파이프라인

점점 인간의 개입이 줄어들고, AI가 스스로 더 잘 생각하는 방향으로 진화함!

빅픽처 — 스케일링의 패러다임이 바뀌고 있다

기존 패러다임
(Training-Time Scaling)

— 더 많은 데이터 + 더 큰 모델 = 더 좋은 AI

— 학습할 때 돈을 씀

— GPT-1 → GPT-4 시대

새로운 패러다임
(Test-Time Scaling)

— 모델 크기를 키우는 것에 한계가 보이기 시작

— 추론할 때 더 잘 생각하게 함

— OpenAI o1 / DeepSeek-R1 시대

GPT-1 → GPT-4

파라미터 스케일링 시대

OpenAI o1 / DeepSeek-R1

추론 시간 스케일링 시대

한계점 및 열린 질문

어려운 문제의 벽	Test-Time Compute도 가장 어려운 문제에서는 효과 없음 — 근본적 한계 존재
리워드 해킹	검증 가능한 수학·코딩 외 도메인(창작, 윤리 판단 등)에서 신뢰할 보상 신호 구축 어려움
비수학 도메인	현재 추론 AI들의 강점은 수학·코딩에 집중 — 일반 도메인 일반화는 미완성

열린 질문 — 현재 연구 흐름에서의 잠정적 답

Q. PRM tree-search와 Revision을 결합하면?	*→ 현재 활발히 연구 중. o1·o3는 내부적으로 유사한 결합을 사용한 것으로 추정됨*
Q. AI 추론이 발전하면 언제까지 인간이 검증할 수 있을까?	*→ 슈퍼얼라인먼트(Superalignment)의 핵심 과제. AI가 AI를 감독하는 방향으로 연구 중*
Q. 수학 이외의 분야에서도 이 패러다임이 통할까?	*→ 제한적으로 통함. 검증 가능한 도메인(코딩, 논리)은 확장 중. 창작·윤리 등은 보상 신호 설계 자체가 미해결*

핵심 메시지

01 결과가 아닌 과정을 보상하면 AI가 더 잘 배우고, 더 안전해진다

02 문제 난이도에 맞게 생각하는 방식을 바꾸면 4배 효율이 오른다

03 인간 없이도 AI는 스스로 깨달음을 얻고 세계 최고 수준에 도달할 수 있다

"AI를 더 똑똑하게 만드는 열쇠는, 더 많이 가르치는 것이 아니라 더 잘 생각하게 하는 것이다"

Lightman et al. (2023) · Snell et al. (2024) · DeepSeek-AI (2025)

Thank you