소크라테스-PRM벤치: 체계적 추론 패턴을 통한 프로세스 보상 모델 벤치마킹

Xiang Li[1,2,3] , **Haiyang Yu**[3] , **Xinghua Zhang**[3] , **Ziyang Huang**[1,2] , **Shizhu He**[1,2] * ,
Kang Liu[1,2] , **Jun Zhao**[1,2] , **Fei Huang**[3] , **Yongbin Li**[3]
[1] 중국과학원 자동화연구소
([2]) 중국과학원 대학 인공지능 학부
[3] 알리바바 그룹 통이 연구소

초록

프로세스 보상 모델(PRM)은 각 중간 추론 단계의 정확성을 검증함으로써 복잡한 추론 및 문제 해결 작업(예: 장기적 의사 결정을 수행하는 대규모 언어 모델 에이전트)에서 핵심적인 역할을 합니다. 실제 시나리오에서 대규모 언어 모델은 문제 해결을 위해 다양한 추론 패턴(예: 분해)을 적용할 수 있으며, 이로 인해 다양한 추론 패턴 하에서 오류가 발생할 가능성이 있습니다. 따라서 PRM은 추론 과정 중 다양한 추론 패턴 하에서의 오류를 식별할 수 있어야 합니다. 그러나 기존 벤치마크는 주로 단계별 정확성을 평가하는 데 초점을 맞추어 다양한 추론 패턴 하에서의 체계적인 PRM 평가를 간과해 왔습니다. 이러한 격차를 해소하기 위해, 우리는 *변환(Transformation)*, *분해(Decomposition)*, *재수집(Regather)*, *추론(Deduction)*, *검증(Verification)*, *통합(Integration)* 등 여섯 가지 추론 패턴 하에서 PRM을 체계적으로 평가하는 새로운 벤치마크인 SOCRATIC-PRMBENCH를 소개합니다. SOCRATIC-PRMBENCH는 앞서 언급한 여섯 가지 추론 패턴 내에서 결함이 있는 2995개의 추론 경로로 구성됩니다. 비판 모델로 프로ンプ팅된 PRM과 LLM에 대한 실험을 통해 기존 PRM의 현저한 결함을 확인하였다. 이러한 관찰은 다양한 추론 패턴 하에서 추론 단계를 평가하는 데 있어 현재 PRM의 중대한 취약점을 부각시킨다. SOCRATIC-PRMBENCH가 다양한 추론 패턴 하에서 PRM을 체계적으로 평가하는 포괄적인 테스트베드 역할을 하고 향후 PRM 발전의 토대를 마련하기를 바란다[1].

1 서론

검증 가능한 보상 기반 강화 학습(RLVR) (Trung et al., 2024; Shao et al., 2024) 및 실행 시 스케일링(Snell et al., 2025; Bansal et al., 2025)는 복잡한 추론 및 의사 결정 작업에서 상당한 능력을 입증했습니다. 이러한 발전에서 프로세스 보상 모델(PRMs)(Lightman et al., 2024; Wang et al., 2023; Zhang et al., 2025)은 특히 장기적 의사결정 단계를 포함하는 LLM 에이전트에 있어 핵심적인 역할을 수행한다(Choudhury, 2025; Ma et al., 2025; Xiong et al., 2025). 추론 과정 중 단계별 보상을 제공함으로써 PRM은 보다 정확하고 밀도 높은 보상 신호를 제시하며, 이는 결국 LLM의 최적화와 추론 경로의 탐색을 유도한다(Tie et al., 2025; Ji et al., 2025).

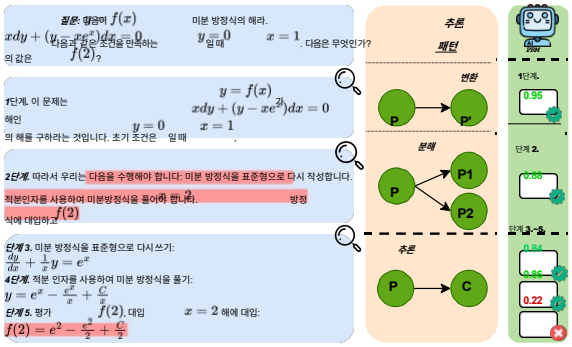

그림 1: (왼쪽): 주어진 질문에서 추론 단계 2와 5에 오류가 존재함. (가운데): 각 단계는 특정 추론 패턴을 적용함. (오른쪽): 프로세스 보상 모델은 연역 패턴의 오류를 성공적으로 탐지하지만 분해 추론 패턴에서는 실패함.

그러나 추론 과정에서 대규모 언어 모델(LLMs)이 적용하는 다양한 추론 패턴(Dong et al., 2023; Li et al., 2024)은 추론 결과에 대한 정확한 보상을 일관되게 제공하는 데 있어 추론 결과 평가 모델(PRMs)에 도전 과제를 제기한다. 그림 1은 이러한 시나리오를 보여준다: 고대 그리스 철학자 소크라테스의 이론(Dong et al., 2023; Qi et al., 2023)에 따르면, 단계 1의 추론 패턴은 '변환', 단계 2는 '분해', 단계 3-5는 *'연역'이다.* 기존 PRM은 단계 5(*연역* 패턴)의 오류를 식별하지만, 분해 패턴에서 비롯된 이 오류의 근본 원인은 탐지하지 못한다.

---

* 교신저자

[1] 본 연구의 코드와 데이터는 https://github. com/Xiang-Li-oss/Socratic-PRMBench 에서 확인할 수 있습니다.

	PRM 벤치마크?	오류 유형 감지?	세분화된 클래스	추론 패턴†	주석 작성자	테스트 케이스 크기	평균 단계 수
RMBench (Liu 외, 2025)	✗	✗	1	1	합성 + 인간	1,327	-
CriticBench (Lin et al., 2024)	✗	✗	1	1	-	-	-
MathCheck-GSM (Zhou 외, 2025)	✗	✗	1	1	합성	516	-
ProcessBench (Zheng et al., 2024)	✓	✗	1	1	인간	3,400	7.1
PRMBench (Song 외, 2025)	✓	✓	9	1	합성 + 인간	6,216	13.4
소크라테스-PRMBench	✓	✓	20	6	합성 + 인간	2995	8.7

표 1: 제안된 SOCRATIC-PRMBENCH와 보상 모델 평가를 위한 다른 벤치마크 또는 데이터셋 비교. †: 벤치마크 내에서 포함된 추론 패턴의 수.

특히, 단계 2에서 미분 방정식 해에 특정 점을 대입하여 상수 C를 계산하는 과정을 생략함으로써, 이후 추론 과정 전반에 걸쳐 C가 미결정 상태로 남아 최종 답변에 오류가 발생한다. 이 관찰 결과는 현재의 PRM이 다양한 추론 패턴에 대해 신뢰할 수 없음을 시사한다.

다양한 추론 패턴에 걸친 PRM의 오류 탐지 능력을 포괄적으로 평가하기 위해 체계적이고 세분화된 벤치마크인 SOCRATIC-PRMBENCH를 소개한다. 체계적인 평가가 제한적이었던 기존 벤치마크(Zheng et al., 2024; Song et al., 2025)와 달리, 고대 그리스 철학자 소크라테스에서 영감을 받아 *변환*, *분해*, *재수집*, *추론*, *검증*, *통합이라는* 6가지 추론 패턴에 걸쳐 PRM의 오류 탐지 능력을 평가하도록 설계했습니다. 구체적으로, SOCRATIC-PRMBENCH는 2995개의 추론 경로로 구성되며, 오류는 추론 패턴에 따라 6개의 주요 범주와 20개의 세분화된 오류 유형 하위 범주로 분류됩니다. SOCRATIC-PRMBENCH의 데이터 주석 과정은 대규모 언어 모델(LLM)을 사용하여 완전히 자동화되어, 광범위한 인적 노동의 필요성을 없앴습니다. 규칙 기반 필터링을 통해 데이터의 난이도를 보장하고, 전문가의 수동 검토를 통해 데이터의 품질을 보장합니다.

우리는 오픈소스 PRM을 비롯해 범용 및 추론 특화 대규모 언어 모델(LLM)을 포함한 광범위한 모델에 대한 심층 실험을 수행했습니다. 연구 결과는 현재 PRM의 상당한 개선 여지를 드러냈습니다. 특히 최고 성능을 보인 Qwen2.5-Math-PRM조차 전체 점수 68.0에 그쳤습니다. 상세한 분석 실험을 통해 우리는 다양한 추론 패턴에 걸쳐 현재 PRM의 오류 탐지 능력에 상당한 차이가 있음을 확인했으며, 오류 단계 식별의 명백한 지연과 보상 생성의 상당한 편향도 발견했습니다. 평가를 위해 SOCRATIC-PRMBENCH를 활용함으로써, 우리는 추론 패턴의 관점에서 PRM을 종합적으로 평가할 수 있는 방법을 제시합니다.

이것은 향후 PRM 개발에서 보상 해킹 위험을 완화하는 데 잠재적으로 도움이 될 수 있습니다. 전반적으로 우리의 기여는 다음과 같이 요약됩니다:

- 추론 패턴 관점에서 최초의 체계적인 PRM 벤치마크인 SOCRATIC-PRMBENCH를 제안합니다. 이는 프로세스 보상 모델에 대한 포괄적이고 세분화된 평가를 위해 2995개의 샘플로 구성됩니다.
- 고대 그리스 논리 이론(Qi et al., 2023)을 기반으로 한 SOCRATIC-PRMBENCH는 변환, 분해, 재집합, 추론, 검증, 통합 등 6가지 신중하게 설계된 추론 패턴과 20개의 세분화된 오류 유형 하위 범주를 포함합니다. 이 체계적인 접근법은 2023년 10월 15일부터 11월 15일까지 진행
  ers는 *변환*, *분해*, *재집합*, *추론*, *검증*, *통합* 등 6가지 신중하게 설계된 추론 패턴과 20개의 세분화된 오류 유형 하위 범주를 포함합니다. 이 체계적이고 세분화된 평가 프레임워크는 PRM의 포괄적 평가를 가능하게 하며 잠재적 단점 식별을 용이하게 합니다.
- 우리는 SOCRATIC-PRMBENCH를 활용하여 다양한 최첨단 PRM 및 대규모 언어 모델(LLM)에 대한 광범위한 실험을 수행했습니다. 그 결과는 현재 PRM의 근본적인 한계를 드러내며, 이 분야의 향후 발전을 위한 통찰력을 제공합니다.

2 관련 연구

프로세스 보상 모델 프로세스 보상 모델(PRM)은 중간 추론 단계에 대해 보다 정확하고 밀도 높은 보상 신호를 제공함으로써 결과 보상 모델(ORM)에 비해 우월성을 입증하였다(Zhang et al., 2024; Ankner et al., 2024). 그 결과 PRM 개발에 대한 관심이 점차 증가하고 있다. Lightman 등(2024)은 PRM 훈련을 위한 수동 주석 데이터셋을 기여했으며, Wang 등(2024)은 몬테카를로 추정법을 활용한 자동 단계 수준 라벨링 방법을 제안했다. 또한 Dong 등(2024); Zhao 등(2025)은 프로세스 보상 모델링을 생성 작업으로 구성하고 CoT 추론을 활용하여 PRM의 생성 능력을 향상시켰다. PRM 훈련의 활발한 발전과 대조적으로, PRM 평가 방법은 여전히

비교적 덜 발달된 상태이다. 이러한 불균형을 해소하기 위해, 우리는 PRM 평가를 위한 새로운 벤치마크인 SCORATIC-PRMBENCH를 제시한다.

보상 모델 벤치마크 보상 벤치마크는 보상 모델 평가에 필수적이며, 직접적이고 정량화 가능한 측정 기준을 제공합니다. 수많은 벤치마크(Liu et al., 2025; Lin et al., 2024; Lambert et al., 2024)가 등장했음에도 불구하고, 이들은 주로 ORM 평가를 위해 설계되었으며 단계별 주석이 전혀 없습니다. Zheng et al. (2024); Song et al. (2025)은 대규모 언어 모델(LLMs)과 인간 전문가를 활용해 단계별 레이블을 주석 처리하여 PRM 벤치마크를 생성했습니다. 그러나 그들의 평가는 체계적이지 않으며, 다양한 추론 패턴에 대한 PRM의 오류 탐지 능력 평가 필요성을 간과했습니다(Dong et al., 2023; Li et al., 2024). 이러한 격차를 해소하기 위해, 우리는 추론 패턴 관점에서 PRM을 포괄적으로 평가할 수 있는 체계적이고 세분화된 벤치마크인 SOCRATIC-PRMBENCH를 제안한다. 본 SOCRATIC-PRMBENCH와 기존 보상 모델 벤치마크 간의 비교는 표 1에 요약되어 있다.

3 소크라테스식-PRM벤치

3.1 추론 패턴

SOCRATIC-PRMBENCHMARK의 추론 패턴 설계는 고대 그리스 철학자 소크라테스의 논리 이론에서 영감을 받았습니다. 소크라테스가 말했듯이, "나는 누구에게도 아무것도 가르칠 수 없다. 나는 그들에게 생각하게 할 뿐이다."라는 철학적 지혜에 따라, 우리는 추론을 여섯 가지 원자적 추론 패턴으로 분류하고, 이 여섯 가지 추론 패턴 내에서 총 20가지 유형의 추론 오류를 체계적으로 설계했습니다. 소크라테스의 논리적 틀 아래 원자적 추론 패턴과 세분화된 오류 유형 범주는 그림 2에 설명되어 있습니다.

변환은 문제를 동질적이거나 유사한 문제로 변환하거나 문제를 추상화합니다. 일반적으로 문제 해결 관점에서 문제를 설명하여 문제에 대한 보다 포괄적이고 명확한 이해를 얻는 것을 목표로 합니다. 구체적으로 *변환* 평가 범주는 ***변환*** 불일치와 ***변환 반사실성이라는*** 두 하위 범주로 나눌 수 있습니다. *변환* 단계 $P \rightarrow P'$ *에 대해*, 변환 불일치는 P'*가* P*와* 논리적, 의미론적 또는 이해 측면에서 일관성을 결여함을 의미한다. 변환 반사실성은 사실적 $P^{()}$*에서*진실값 G*에* 대한 오차. **분해는** 문제를 관리 가능한 하위 문제로 분할하거나 추론 단계를 위한 계획을 수립하여 각 하위 문제를 해결함으로써 주요 문제를 해결한다. 구체적으로, *분해* 평가 범주는 세 가지 하위 범주로 나눌 수 있다: ***분해 부적합성***, ***분해 중복성***, ***분해 불완전성***. *분해* 단계 $P \rightarrow \{P_1, P_2, ..., P_n\}$*에* 대해, 세 하위 범주는 각각 *하위* 문제 $P_{(i)}$*에서*발생하는 서로 다른 유형의 오류를 나타낸다. 이는 논리적 부등식으로 인한 부정확성, 중요한 하위 문제 및 조건의 누락, 또는 중복된 하위 문제와 제약 조건의 포함으로 인해 발생할 수 있다.

재수집은 문제 해결과 관련된 입력에서 핵심 정보를 수집하고, 문제 해결에 중요한 원칙 및 기타 개념을 식별합니다. 구체적으로 *재수집* 평가 범주는 세 가지 하위 범주로 나눌 수 있습니다: ***재수집 부정확성***, ***재수집 중복성***, ***재수집 불완전성***. *재수집* 단계 $P \rightarrow \{Q_1, Q_2, ..., Q_n\}$에서 재수집 부정확성은 문제 P 해결에 부적합한 정의의 오용이나 잘못된 정보를 포함한 Q_i를 수집하는 것을 의미합니다. 재수집 중복성은 P*와* 관련 없는 중복되거나 관련 없는 정보를 수집하는 것을 의미합니다. 재수집 불완전성은 핵심 정의, 중요한 원칙 및 개념의 부재를 가리킵니다.

추론은 주어진 전제에서 직접 결론을 도출한다. 구체적으로 *추론* 평가 범주는 여섯 가지 하위 범주로 나뉜다: ***전제 비건전성***, ***전제 불완전성***, ***전제 중복성***, ***결론 무효성***, ***결론 모순성***, ***결론 반사실성***. *추론* 단계 $P \rightarrow C$*에서*, 처음 세 하위 범주는 전제에서 발생하며 다음을 포함한다: (1) 불합리하거나 잘못된 전제로부터 추론 시작, (2) 전제에 중복된 가정 도입, 그리고
(3) 핵심 조건 및 제약 사항 생략. 나머지 세 하위 범주는 결론에서 비롯되며 다음과 같습니다: (1) 올바른 전제에서 잘못된 결론을 도출하는 경우, (2) 이전 결론과 모순되는 결론을 도출하는 경우, (3) 알려진 사실과 일치하지 않는 결론을 도출하는 경우.

검증은 사실적 정확성, 논리적 일관성 등의 측면에서 추론 단계를 검토하여 잠재적 오류를 탐지하고 반복적으로 정제합니다. 구체적으로 *검증* 평가 범주는

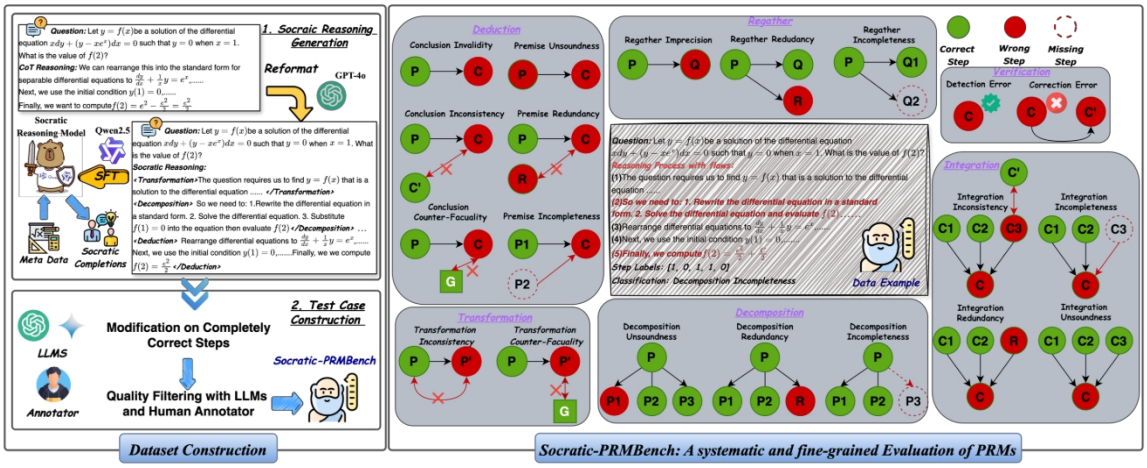

그림 2: SOCRATIC-PRMBENCH 개요. 왼쪽은 데이터셋 구축 절차를, 오른쪽은 6가지 추론 패턴과 20가지 세분화된 오류 유형 하위 범주를 보여줍니다. (하위)문제와 결론을 각각 P와 C로, 수집된 정보, 중복 내용, 실제 진실을 각각 Q, R, G로 표기합니다.

은 두 가지 하위 범주로 나뉩니다: ***탐지*** 오류와 ***수정 오류***. 전자는 잘못된 결론 C를 식별하지 못하는 것을 의미합니다. 반면 후자는 C의 초기 오류를 인식했지만 수정 시도 과정에서 새로운 오류를 도입하여 다른 잘못된 결론 C'를 초래하는 경우를 포함합니다.

통합은 도출된 결론을 요약하여 새로운 결론을 도출하며, 모든 현재 추론 과정을 통합하여 최종 결론을 형성한다. 구체적으로 *통합* 평가 범주는 네 가지 하위 범주로 나눌 수 있다: ***통합 불일치***, ***통합 불완전성***, ***통합 중복성***, ***통합 비건전성***. 통합 단계 $\{C_1, C_2, ..., C_n\} \rightarrow C$, 처음 세 가지 오류 유형은 중간 결론 C_i 에서 비롯되며, 여기에는 이전 발견과 모순되는 결론의 존재, 중요한 결론의 누락, 불필요하거나 중복된 결론의 도입이 포함됩니다. 마지막 오류 유형인 통합 부합성 오류는 통합된 결론이 모두 타당성과 완전성을 만족하더라도 최종 결론 C가 부정확하거나 비합리적인 경우를 의미합니다.

3.2 벤치마크 구축

데이터셋 구축 파이프라인은 **소크라테스적 추론 생성(SRG)**과 **테스트 케이스 구축(TCG)**이라는 두 가지 핵심 단계로 구성됩니다.

3.2.1 소크라테스적 추론 생성

이 단계는 일련의 순서로 표현되는 소크라테스식 추론 과정의 데이터 풀을 생성하는 것을 목표로 합니다.

원자적 소크라테스식 추론 행위. 그림 2의 왼쪽 부분에서 설명된 바와 같이, 각 추론 단계는 시작 태그 <[Pattern]>과 종료 태그 </[Pattern]>로 둘러싸여 있습니다. [Pattern] 자리표시자 내부의 내용은 이 특정 단계를 특징짓는 구체적인 추론 패턴을 나타냅니다 .

소크라테스식 추론 모델 훈련 사용 가능한 소크라테스식 추론 데이터가 부족함을 감안하여, 데이터 생성을 용이하게 하기 위해 초기 단계에서 특수화된 소크라테스식 추론 모델을 훈련시켰다. 이를 위해 MATH-Hard (Hendrycks et al., 2021) 및 Open-o1 (OpenO1, 2024) 데이터 세트에서 19,000개의 인스턴스를 샘플링하고 GPT-4o에 기존 사고의 사슬(CoT) 주석을 소크라테스식 추론 과정으로 변환하도록 프롬프트합니다. 그런 다음 Qwen2.5-72b-instruct (Team, 2024a)를 이러한 소크라테스식 추론 과정에 대해 미세 조정하여 $_{M(Socratic)}$으로 표시되는 소크라테스식 추론 모델을 얻었습니다.

소크라테스식 추론 생성 이후, 메타데이터로부터 새로운 소크라테스식 추론 과정을 생성하기 위해 $M_{Socratic}$을 활용한다. 이를 위해 먼저 GSM8k (Cobbe et al., 2021), Omni-Math (Gao et al., 2024), MathBench (Liu et al., 2024), OlympiadBench (He et al., 2024a)에서 샘플을 수집합니다. 문제의 적절한 난이도를 보장하기 위해 Omni-Math 및 MathBench 데이터셋을 신중하게 선별했습니다. 구체적으로, 난이도 평가가 4.0 미만인 Omni-Math 샘플은 모두 제외했습니다. MathBench의 경우, 개념적 이해보다는 이론적 적용을 강조하는 MathBench-A 하위 집합에만 집중했습니다. 또한 MathBench-A에서 다음으로 지정된 인스턴스만 유지했습니다.

	전체	변환	분해	재구성	추론	통합	검증
평균 단계 수	8.7	8.5	8.7	8.6	8.5	8.5	10.8
평균 오차 단계	3.0	4.2	3.3	2.9	3.0	2.0	3.8
평균 첫 번째 오류 단계	4.7	1.5	3.0	3.1	5.4	7.2	6.9
평균 질문 길이	209.6	224.4	220.7	207.5	221.7	191.3	169.4
인스턴스 수	2995	313	463	463	926	615	215

표 2: SOCRATIC-PRMBENCH 통계.

고등학교 또는 대학 수준으로. 이 절차는 최종적으로 데이터 풀 D를 생성한다. D 내의 각 질문-답변 쌍 (q_i, a_i)에 대해, $M_{Socratic}$은 소크라테스식 추론 과정 $r_{(i)}$ 를 생성하여 $(q_i, r_i, \hat{a}_i)$ 삼중항을 산출한다.

소크라테스식 추론 큐레이션 마지막으로, 각 $(q_i, r_i, \hat{a}_i)$ 튜플은 엄격한 이중 검증 과정을 거칩니다: 먼저 답변의 정확성을 평가한 후, 각 개별 단계를 LLM 기반 검증으로 확인합니다. 두 검증 모두 통과한 튜플만 유지되어 큐레이션된 메타데이터 세트 D'가 생성됩니다. 답변 검증에는 Qwen2.5-Math(Yang et al., 2024) 방식을 따르며, 예측된 답변 $\hat{a}_i$가 *정답* a와 수치적 및 기호적 동등성을 모두 충족해야 합니다. 단계 검증에는 GPT-4o(OpenAI, 2024a)를 활용하여 추론 과정의 각 개별 단계 정확성을 평가하며, 상세 프롬프트는 부록 B에 제시합니다.

3.2.2 테스트 케이스 구성

이 단계에서는 제어된 오류 주입 절차를 활용하여 각 오류 유형 C(3.1절에서 분류된 바와 같이)에 대한 테스트 세트를 생성합니다. 각 오류 유형 C(예: 반복 불일치)에 대해 테스트 세트 $T_{(C)}$를 생성합니다. 이는 먼저 메타데이터 세트 D'에서 N*개의* 샘플을 무작위로 선택함으로써 달성됩니다. 그런 다음 문제 q_i, 이중 검증 과정을 통해 완전히 정확함이 보장된 추론 경로 r_i를 포함하는 각 샘플 $(q_{(i)}, r_i, a_i)$에 대해, GPT-4o에게 원래 정확한 추론 과정 $r_{(i)}$를 수정하도록 프롬프트하여 *오류 유형* C*와* 일관된 오류를 의도적으로 도입합니다:

$$\tilde{r}_i = \text{LLM}(I, [q_i, r_i, a_i], C)$$
$$T_C = \{t_i = (q_i, \tilde{r}_i, \tilde{a}_i)\}_{i=1}^{N} \quad (1)$$

$\tilde{r}_{(i)}$는 오류 유형 C를 적용한 수정된 소크라테스식 추론 과정이며, I는 GPT-4o가 원래 과정 r_i를 $\tilde{r}_i$로 수정하도록 지시하는 프롬프트이다. 상세한 프롬프트는 부록 B에 수록되어 있다.

3.3 품질 관리

SOCRATIC-PRMBENCH의 높은 품질과 신뢰성을 보장하기 위해, 우리는 규칙 기반 기초 및 LLM 기반 방법을 통해 부적합한 샘플을 걸러내어, 최종적으로 SOCRATIC-PRMBENCH를 구축하였습니다.

규칙 기반 필터링 *지침 I에서* 상세한 작업 설명과 출력 형식 요구사항을 제공했음에도 불구하고, GPT-4o는 가끔 지침 I*을* 엄격히 따르지 못할 수 있습니다. 따라서 규칙 기반 필터링 방법을 구현합니다. 첫째, 문자열 일치 기능을 사용하여 *지침 I에서* 요구하는 JSON 형식으로 출력을 생성하지 못한 샘플을 식별하고 제거합니다. 둘째, 정규 표현식을 사용하여 최종 답변을 성공적으로 출력하지 못한 샘플을 제거합니다.

LLM 기반 필터링 생성된 테스트 케이스의 품질을 보장하기 위해, 주어진 오류 유형 C*에* 대한 테스트 세트 T_C내의 각 샘플 $(q_i, \tilde{r}_i, \tilde{a}_i)$ 을 평가하기 위해 Gemini2.5-Pro를 활용합니다. 구체적으로, Gemini2.5-Pro가 다음 두 기준에 따라 샘플을 평가하도록 지시합니다: (1) 추론 경로 $\tilde{r}_i$가 표면적으로는 타당해 보이지만 근본적인 추론 오류가 포함되어 있는지, (2) 식별된 오류가 대상 오류 *유형* C*에* 확실히 속하는지 여부입니다. 자세한 프롬프트는 부록 B에 제시되어 있습니다. Gemini2.5-Pro에 의한 필터링 후 샘플의 승인률은 92.7%에 달하며, 2995개의 샘플이 최종 Socratic-PRMBench를 구성하기 위해 유지됩니다. Socratic-PRMBench의 통계는 표 2에 제시되어 있습니다.

LLM과 인간 어노테이터 간의 일관성 Gemini2.5-Pro가 이러한 품질 필터링 작업을 수행할 수 있는 능력을 입증하기 위해, 우리는 인간 어노테이터와의 일치도를 측정합니다. 우리는 각각 최소 학사 학위를 보유한 세 명의 자원 어노테이터를 모집합니다.

gree를 보유하고 있으며, 무작위로 추출된
동일한 기준을 적용한 데이터의 10% 부분집합
Gemini2.5-Pro와 함께 사용합니다. 그런 다음 Gemini2.5-Pro와 인간 주석자 간의 일치율을 계산합니다. 그 결과, Gemini2.5-Pro는 인간 주석자와 높은 수준의 일관성을 보여 평균 93.3%의 일치율을 달성했습니다. 이러한 높은 수준의 일관성은 Gemini2.5-Pro가 품질 필터링 수행에서 인간 주석자를 효과적으로 대체할 수 있다는 강력한 증거를 제공합니다.

모델	변환		분해			재집결			검증	
	TT.	TF.	DC.	DR	DS.	GP.	GC.	GR.	CE.	DE.
프로세스 보상 모델(PRM)										
Skywork-PRM-7B	38.7	38.4	42.7	42.5	38.0	42.8	44.8	41.3	47.9	46.7
ReasonEval-7B	50.9	50.9	59.3	50.1	53.7	52.4	59.6	49.7	66.7	59.2
RLHFlow-PRM-Mistral-8B	50.6	52.7	46.6	47.3	42.7	38.0	44.6	48.7	53.1	49.5
RLHFlow-PRM-Deepseek-8B	47.5	50.8	50.6	50.9	44.0	41.6	48.6	55.4	45.9	47.6
MathShepherd-Mistral-7B	54.5	50.9	59.4	**57.4**	56.7	**60.9**	59.4	54.6	**72.7**	**72.1**
Qwen2.5-수학-PRM-7B	**55.8**	**64.3**	**61.7**	51.6	**58.4**	57.5	**61.8**	**58.2**	67.4	64.1
비평가 모델로 프롬프트된 대규모 언어 모델(LLMs)										
GPT-4o	**62.4**	60.5	69.9	60.0	66.1	64.9	**74.1**	57.9	74.4	75.8
딥시크-R1	51.9	**72.6**	63.4	64.4	67.1	70.9	64.6	54.8	75.0	77.1
QwQ-32B	60.2	68.6	70.0	**67.9**	59.8	73.7	65.8	55.4	75.8	75.7
Gemini-2.5-Pro	62.3	64.4	67.3	61.4	**68.5**	70.2	69.2	**58.6**	**78.3**	**78.0**
o3-mini	**62.4**	67.4	**70.4**	57.3	68.0	**77.3**	71.3	53.0	77.2	72.6

모델	전체	공제						통합			
		CF.	CT.	CV.	PC.	PR.	PS.	IC.	IT.	IR.	IS.
프로세스 보상 모델(PRM)											
Skywork-PRM-7B	43.6	42.5	41.2	40.0	41.8	42.8	39.8	38.7	42.6	39.4	44.2
ReasonEval-7B	61.9	63.6	63.6	66.3	61.9	65.2	63.5	69.7	78.2	68.7	76.1
RLHFlow-PRM-Mistral-8B	48.8	50.4	46.2	45.2	46.1	44.5	43.3	51.2	58.1	46.6	56.3
RLHFlow-PRM-Deepseek-8B	51.5	51.5	52.4	52.0	47.6	51.4	45.2	55.3	63.7	53.3	66.7
MathShepherd-Mistral-7B	64.4	68.0	65.9	66.5	62.4	65.9	65.4	63.1	74.2	60.1	72.3
Qwen2.5-수학-PRM-7B	**68.0**	**74.7**	**73.1**	**72.2**	**66.6**	**72.4**	**67.2**	**75.0**	**85.2**	**69.6**	**86.9**
비평가 모델로 프롬프트된 대규모 언어 모델(LLMs)											
GPT-4o	70.8	63.6	62.7	74.5	73.2	60.1	76.1	73.4	80.8	52.7	88.7
딥시크-R1	73.0	80.8	72.6	77.2	68.6	72.0	76.9	75.9	78.9	59.9	88.6
QwQ-32B	73.8	70.3	75.0	**85.2**	**74.0**	69.5	77.5	**81.8**	83.5	58.7	96.7
Gemini-2.5-Pro	73.5	72.8	77.7	83.5	69.0	65.9	73.5	73.2	**88.9**	56.9	**96.9**
o3-mini	**75.7**	**83.3**	**81.0**	81.4	73.9	**75.3**	**78.6**	78.7	87.3	**72.0**	87.0

표 3: SOCRATIC-PRMBENCH에 대한 평가 결과. (상단): *변환*, *분해*, *재수집* 및 *검증의* PRM 점수. (하단): *추론*, *통합* 및 *전체* 성능의 PRM 점수. 각 범주 및 작업별 최고 성능은 **굵은 글씨**로 표시됨. 약어의 전체 명칭은 부록 A에 제시됨

전체 데이터셋에 걸쳐 수행되어 방대한 수작업의 부담을 줄였습니다.

4 실험

4.1 모델

본 연구에서는 프로세스 보상 모델(PRM)과 비평가 모델로 프롬프트된 대규모 언어 모델(LLM)이라는 두 가지 유형의 모델을 고려합니다.

프로세스 보상 모델(PRMs)은 언어 모델의 중간 추론 과정을 평가하고 감독하기 위해 중간 추론 단계의 주석과 함께 훈련됩니다. 우리의 평가에는 다음과 같은 최신 오픈소스 PRMs가 포함됩니다:

(1) MathShepherd (Wang et al., 2023)는 각 단계가 올바른 최종 답으로 이어질 경험적 확률을 추정하여 해당 단계의 과정 라벨을 획득합니다. (2) LLaMA-3.1 기반 생성형 PRM 두 가지 (Dong et al., 2024)는 "Yes/No" 토큰의 출력 확률을 기반으로 정답 여부를 판단합니다. (3) ReasonEval(Mondorf and Plank, 2024)는 추론 단계의 타당성 외에도 중복성을 평가합니다. (4) 인기 수학 모델 Qwen2.5-Math로 훈련된 두 가지 PRM, 즉 Skywork-PRM(He et al., 2024b)과 Qwen2.5-Math-PRM(Zhang et al., 2024)입니다.

인기 수학 모델 Qwen2.5-Math로 훈련된 두 가지 PRM, 즉 Skywork-PRM(He et al., 2024b)과 Qwen2.5-Math-PRM(Zhang et al., 2025).

비평가 모델로 활용된 대규모 언어 모델(LLMs) 비평가 모델은 대규모 언어 모델의 생성 능력을 활용하여 모델이 생성한 텍스트에 직접 피드백과 비평을 제공하는 것을 목표로 합니다. 우리의 평가는 GPT-4o (OpenAI, 2024a), Gemini2.5-Pro (Deepmind, 2025)를 포함한 범용 모델과 Deepseek-R1 (DeepSeek-AI, 2025), QwQ-32B(Team, 2024b), o3-mini(OpenAI, 2025) 등 추론에 특화된 모델을 모두 포함합니다.

4.2 평가 지표

PRM 평가가 결함이 있는 추론 단계의 탐지에 중점을 둔다는 점을 고려할 때, 정확도나 F1 점수를 직접 적용하는 것은 모델의 내재적 편향에 영향을 받을 수 있습니다. 이러한 문제를 해결하기 위해, 우리는 (Song et al., 2025; Zheng et al., 2024)를 따르고 PRM 점수를 우리의

평가 지표로 채택합니다. 공식적으로 다음과 같이 정의됩니다:

$$\text{PRM-Score} = w_1 \times \text{F1}_{\text{neg}} + w_2 \times \text{F1} \quad (2)$$

여기서 F1과 F1_{neg}은 각각 F1 점수와 음성 F1 점수를 의미합니다. w_1과 w_2는 F1 점수와 음성 F1 점수의 기여도를 균형 있게 조정하는 가중치입니다. 선행 연구(Song et al., 2025; Zheng et al., 2024)에 따라 $w_1 = w_2 = 0.5$로 설정했습니다.

4.3 주요 결과

우리의 평가 결과는 표 3에 제시되어 있습니다. 우리의 연구 결과는 다음과 같습니다:

PRM과 LLM의 비교 PRM의 성능은 LLM에 비해 현저히 열등함이 입증되었다. 최고 성능의 PRM인 Qwen2.5-Math-PRM-7B조차 68.0점이라는 점수를 기록했는데, 이는 가장 낮은 성능의 LLM인 GPT-4o보다도 낮은 수치이다. 더욱이 일부 PRM은 무작위 추측 수준보다 낮은 성능을 보이며, 다양한 추론 패턴에 걸친 추론 오류 처리 능력의 한계를 드러냅니다. 이는 PRM과 LLM 간 상당한 격차가 존재함을 시사하며, 근본적인 개선이 필요함을 보여줍니다. PRM 데이터 주석 작업의 어려움과 합성 데이터 품질 보증의 복잡성이 이러한 격차의 원인으로 추정됩니다. 예를 들어, Math-shepherd는 최종 정답 도달 확률 추정치를 기준으로 단계별 정확도를 측정하는 합성 데이터를 활용하는 반면, Qwen2.5-Math-PRM-7B는 수동으로 라벨링된 PRM800k 데이터셋을 사용합니다.

LLM 간 비교 PRM과 달리 LLM은 정교한 언어 및 추론 능력 덕분에 비판적 평가에서 보다 견고하고 신뢰할 수 있는 보상을 제공할 잠재력을 보여줍니다. 이에 부합하게도, 추론 특화 LLM이 범용 LLM보다 우수한 성능을 보인다는 점을 관찰했습니다. 특히 QWQ-32B는 오픈소스 모델 중 최고 성능을 발휘하며 GPT-4o보다도 우수한 결과를 보였습니다. QWQ-32B가 인상적인 성능을 보이지만 여전히 o3-mini에는 미치지 못한다는 점은, 오픈소스 모델과 독점 모델 간 문제 해결 성능 격차가 좁혀지고 있음에도 비평 모델로서의 역량에는 여전히 상당한 차이가 존재함을 시사합니다.

중복 오류는 더 까다롭다 동일한 추론 패턴 내에서도 세분화된 오류 유형에 따라 성능 차이가 두드러지게 관찰되었다. 분해, 재구성, 통합 단계 내의 분해 중복, 재구성 중복, 통합 중복과 같은 중복 오류는

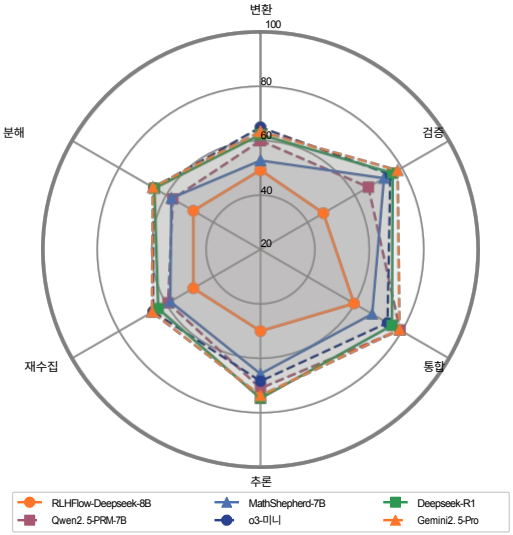

그림 3: 6가지 추론 패턴에 걸친 대표적 PRM과 LLM의 평균 PRM 점수. PRM과 LLM 모두 불균형적인 성능을 보임.

수집 및 통합 패턴은 동일한 추론 패턴 내 다른 오류 유형에 비해 PRM과 LLM 모두에게 지속적으로 더 큰 도전 과제를 제기했습니다. 이는 중복 오류 단계가 다른 유형의 오류 단계보다 종종 더 "정상적"이거나 그럴듯해 보이기 때문에, 모델이 표면 수준의 텍스트 단서만으로 이를 식별하는 능력을 저해하기 때문일 수 있습니다. 이는 현재의 PRM이 오류 탐지를 위해 표면 수준의 패턴 인식에 의존하는 데 한계가 있을 수 있음을 시사하며, 더 심층적인 추론 및 분석 능력의 필요성을 강조한다.

4.4 상세 분석

본 절에서는 제안된 SOCRATIC-PRMBENCH에 대한 보다 정교한 분석을 통해, 프로세스 수준 보상을 제공하는 데 있어 기존 모델의 한계를 규명하고 PRM의 향후 발전을 위한 통찰력을 제시하고자 한다.

추론 패턴별 성능 차이 그림 3에서 볼 수 있듯이, 여섯 가지 추론 패턴에 걸쳐 대표적 PRM과 LLM의 평균 PRM 점수를 제시합니다. 주목할 만한 발견은 PRM과 LLM 모두 다양한 추론 패턴에서 불균형적인 성능을 보인다는 점입니다. 거의 모든 모델의 성능은 *추론*, *통합*, *검증* 패턴에 비해 *변환*, *분해*, 재수집 패턴에서 일관되게 약했습니다. 이 문제는 PRM에서 더욱 두드러집니다. 예를 들어, Qwen2.5-Math-PRM-7B는 통합 패턴에서 80.0에 가까운 PRM 점수를 달성했지만, 재수집 패턴에서는 어려움을 겪었습니다.

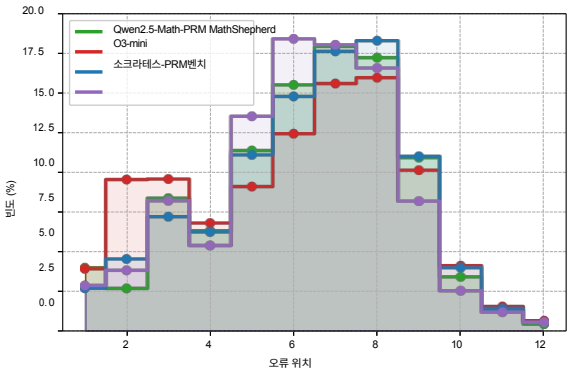

그림 4: SOCRATIC-PRMBENCH의 오류 위치 분포(12개로 잘림) 및 여러 PRM과 LLM의 예측 오류 위치 분포.

분해 패턴에서 60.0에 도달하기 위해 더 많은 시도가 필요했습니다. 이 결과는 현재 PRM 훈련 데이터 구축 과정에 잠재적 편향이 존재할 수 있음을 시사합니다. 기존 PRM 데이터셋은 수동으로 주석 처리되었든 합성 생성되었든, 다양한 추론 패턴을 적절히 대표하지 못하는 것으로 보입니다. *추론과* 같은 특정 패턴의 빈도가 더 높기 때문에, 이러한 데이터셋은 해당 패턴에 의해 지배되는 경향이 있으며, 결과적으로 *분해와* 같은 드문 패턴에 대한 성능이 현저히 저하됩니다. 이 관찰은 추론 오류의 조기 탐지가 오류 전파를 완화하는 데 중요하므로, 향후 PRM 훈련 데이터 구축 시 다양한 추론 패턴의 분포를 고려하는 것이 중요함을 강조합니다.

모델은 오류 단계 식별에 지연을 보임 추론 오류의 적시 탐지 능력을 조사하기 위해, SOCRATIC-PRMBENCH의 진실 오류 단계 위치 분포와 대표적 PRM 및 LLM의 예측 오류 위치 분포를 비교하였다. 그림 4에서 알 수 있듯이, Qwen2.5-Math-PRM과 o3-mini는 실제 분포에 비해 후반 단계로 현저히 이동하는 경향을 보이며, 이는 초기 오류 감지에 지연이 있음을 시사합니다. 이는 초기 오류를 감지하는 능력이 제한적이어서 오류가 확산될 수 있음을 의미합니다. 반면 MathShepherd는 반대 경향을 보이며, 예측 분포가 추론 체인의 시작 부분으로 이동합니다. 이는 MathShepherd가 특히 추론 초기 단계에서 올바른 단계를 오류로 잘못 식별하는 경향이 있음을 시사합니다. 이는 조기 탐지와 과도한 오탐지 회피가 모두 중요하다는 점을 시사합니다. 오류 전파는 계산 자원을 낭비하고 샘플링 효율을 저하시키지만,

모델	정확도			PRM
	수정.	오류.	전체.	점수
무작위 †	50.0	50.0	50.0	50.0
프로세스 보상 모델(PRM)				
ReasonEval-7B	87.3	35.7	69.6	61.9
Skywork-PRM-7B	22.7	93.0	44.5	43.6
MathShepherd	73.3	56.0	67.4	64.4
Qwen2.5-Math-PRM-7B	90.8	42.9	74.5	68.0
LLMs, 비평가 모델로 프롬프트				
GPT-4o	83.0	57.5	74.6	70.8
QwQ-32B	83.9	63.1	76.8	73.8
o3-미니	82.6	69.0	78.0	75.7
Gemini-2.5-Pro	83.6	62.8	76.5	73.5

표 4: 긍정 및 부정 테스트 케이스에서의 모델 성능 비교. (†)는무작위 추측(Random Guess)의 성능을 나타냅니다.

과도하게 공격적인 오류 탐지는 효율성을 저해할 수 있으며, 올바른 추론 경로를 조기에 종료시켜 잠재적으로 최적의 해법 탐색을 방해할 수 있다.

PRM의 보상 편향 표 3은 일부 PRM이 무작위 추측보다도 더 낮은 성능을 보임을 보여주며, 이는 예측에 상당한 편향이 존재함을 시사합니다. 이러한 편향을 정량화하기 위해 각 모델의 올바른 추론 단계와 오류 추론 단계에 대한 정확도를 계산했습니다. 표 4에서 볼 수 있듯이, 결과는 PRM 내부에 명확한 보상 편향이 존재함을 드러내며, 일부 모델은 긍정적 보상을 크게 선호하는 반면 다른 모델들은 부정적 보상을 제공하는 경향이 있습니다. 예를 들어, Qwen2.5-Math-PRM-7B는 올바른 단계에서 90.8%의 정확도를 보이지만 오류 단계에서는 42.9%의 정확도에 그칩니다. 이와는 대조적으로 Skywork-PRM-7B는 오류 단계에서 93.0%의 정확도를 보인 반면, 정답 단계에서는 22.7%의 정확도에 그쳤습니다. LLM은 PRM보다 덜 뚜렷한 편향을 보였으나, 정답과 오류 단계 간 정확도 차이는 여전히 상당했습니다. 또한 평가된 모든 LLM은 긍정적 보상을 선호하는 경향을 보였는데, 이는 비판 모델로 활용될 때 미묘한 오류를 식별하는 신뢰성을 제한할 수 있습니다.

5 결론

본 연구에서는 PRM을 위한 체계적이고 세분화된 벤치마크인 SOCRATIC-PRMBENCH를 제안한다. SOCRATIC-PRMBENCH는 2995개의

개의 사례로 구성되며, 6가지 주요 추론 패턴과 20가지 세분화된 오류 유형 하위 범주로 분류된다. 기존 PRM과 비평 모델로 활용된 대규모 언어 모델(LLM)에 대한 체계적이고 포괄적인 평가를 통해, 우리는 기존 모델의 잠재적 한계점을 관찰하고 PRM 개선을 위한 향후 연구에 유용한 통찰력을 제공한다.

제한 사항

비록 우리의 작업이 PRM에 대한 체계적이고 포괄적인 평가를 제공할 수 있지만, 현재 버전의 벤치마크는 주로 수학 문제와 같이 객관적으로 검증 가능한 답을 가진 추론 작업에 초점을 맞추고 있습니다. 문헌, 의학 또는 법률과 같이 명확한 기준 답안이 종종 존재하지 않는 분야의 작업에 기존 데이터 구축 방법을 적용하는 것은 추가적인 연구가 필요합니다. 향후 벤치마크 버전에서는 더 광범위한 작업을 포괄하도록 확장할 계획입니다.

참고문헌

Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D. Chang, and Prithviraj Ammanabrolu. 2024. Critique-out-loud reward models. *Preprint*, arXiv:2408.11791.

Hritik Bansal, Arian Hosseini, Rishabh Agarwal, Vinh Q. Tran, Mehran Kazemi. 2025. Smaller, weaker, yet better: Training LLM reasoners via compute-optimal sampling. In *The Thirteenth Inter-national Conference on Learning Representations*.

Sanjiban Choudhury. 2025. 대규모 언어 모델 에이전트를 위한 프로세스 보상 모델: 실용적 프레임워크와 방향. *arXiv 사전 인쇄본 arXiv:2502.10325*.

칼 코베, 비닛 코사라주, 모하마드 바바리안, 마크 첸, 준희우, 루카시 카이저, 마티아스 플라퍼트, 제리 트워렉, 제이콥 힐튼, 나카노 레이이치로 외 1인. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Deepmind. 2025. Gemini2.5-pro. https://deepmind.google/technologies/gemini/pro/.

DeepSeek-AI. 2025. Deepseek-r1: 강화 학습을 통한 대규모 언어 모델의 추론 능력 촉진. *사전 인쇄본*, arXiv:2501.12948.

한제 동, 웨이 슝, 보 팡, 하오샹 왕, 한 자오, 잉보 저우, 난 장, 도옌 사후, 카이밍 슝, 통 장. 2024. RLHF 워크플로우: 보상 모델링에서 온라인 RLHF까지. *기계 학습 연구 트랜잭션*.

Qingxiu Dong, Li Dong, Ke Xu, Guangyan Zhou, Yaru Hao, Zhifang Sui, and Furu Wei. 2023. 과학을 위한 대규모 언어 모델: P 대 NP 연구. *arXiv 사전 인쇄본 arXiv:2309.05689*.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu,

장바오바오. 2024. Omni-math: 대규모 언어 모델을 위한 범용 올림피아드 수준 수학 벤치마크. *사전 출판물*, arXiv:2410.07985.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, Maosong Sun. 2024a. OlympiadBench: 올림피아드 수준의 이중 언어 다중 모달 과학 문제로 AGI를 촉진하기 위한 도전적인 벤치마크. *제62회 전산언어학회 연차 총회 논문집 (제1권: 장문 논문)*, 3828–3850쪽, 태국 방콕. 전산언어학회.

허주제, 위톈원, 옌루이, 류자카이, 왕차오지에, 간이멍, 투시원, 크리스 류유하오, 쩡량, 왕샤오쿤, 왕보양, 리용콩, 장푸샹, 쉬자청, 안보, 류양, 저우야후이. 2024b. Skywork-o1 공개 시리즈. https://huggingface.co/Skywork.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, Jacob Steinhardt. 2021. MATH 데이터셋을 활용한 수학 문제 해결 능력 측정. *제35회 신경정보처리시스템 컨퍼런스 데이터셋 및 벤치마크 트랙 (2차)*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen. 2021. Lora: 대규모 언어 모델의 저순위 적응. *사전 인쇄본*, arXiv:2106.09685.

Yixin Ji, Juntao Li, Hai Ye, Kaixin Wu, Jia Xu, Linjian Mo, and Min Zhang. 2025. Test-time computing: from system-1 thinking to system-2 thinking. *arXiv preprint arXiv:2501.02497*.

네이선 램버트, 발렌티나 피아트킨, 제이콥 모리슨, LJ 미란다, 빌 유 천 린, 키야티 찬두, 누하 지리, 사친 쿠마르, 톰 지크, 최예진 외 1명. 2024. Rewardbench: 언어 모델링을 위한 보상 모델 평가. *arXiv 사전 인쇄본 arXiv:2403.13787*.

Changcheng Li, Xiangyu Wang, Qiuju Chen, Xiren Zhou, Huanhuan Chen. 2024. Mtmt: 강화된 대규모 언어 모델을 위한 사고 트리 형성을 위한 다중 사고 모드 통합. *arXiv 사전 인쇄본 arXiv:2412.03987*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, Karl Cobbe. 2024. 단계별 검증 방법. *제12회 국제 학습 표현 컨퍼런스*.

린즈청, 구즈빈, 리양톈, 뤄루이린, 리우하오웨이, 양위주. 2024. 비판적 사고를 위한 대규모 언어 모델 벤치마킹: CriticBench. *컴퓨터 언어학회 학술대회 논문집: ACL 2024*, 1552–1587쪽, 태국 방콕. 컴퓨터 언어학회.

류홍웨이, 정즈롱, 교위쑤안, 두안하오동, 페이즈웨이, 저우펑저, 장원웨이,

장송양, 린다화, 천카이. 2024. MathBench: 계층적 수학 벤치마크를 통한 대규모 언어 모델의 이론 및 응용 능력 평가. *계산언어학회 연구 성과: ACL 2024*, 6884–6915쪽, 태국 방콕. 계산언어학회.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, Juanzi Li. 2025. RM-bench: 언어 모델의 보상 모델을 미묘함과 스타일로 벤치마킹. *제13회 국제 표현 학습 컨퍼런스*.

마잉웨이, 리용빈, 동이홍, 장쉬에, 카오롱위, 천주에, 황페이, 리빈화. 2025. 더 크게가 아닌 더 오래 생각하기: 테스트 시간 컴퓨팅 확장성을 통한 소프트웨어 엔지니어링 에이전트 강화. *arXiv 사전 인쇄본 arXiv:2503.23803*.

Philipp Mondorf and Barbara Plank. 2024. 정확도를 넘어: 대규모 언어 모델의 추론 행동 평가 - 서베이. *제1회 언어 모델링 컨퍼런스*.

OpenAI. 2024a. GPT-4O 시스템 카드. https://cdn.openai.com/gpt-4o-system-card.pdf . 접속일: 2024-09-26.

OpenAI. 2024b. LLM으로 추론하는 법 배우기. https://openai.com/index/learning-to-reason-with-llms/.

OpenAI. 2025. Openai o3-mini 시스템 카드. https://openai.com/index/o3-mini-system-card/.

OpenO1. 2024. Open-o1. https://opensource-o1. github.io/.

Jingyuan Qi, Zhiyang Xu, Ying Shen, Minqian Liu, Di Jin, Qifan Wang, and Lifu Huang. 2023. 소크라테스식 질문의 기술: 대규모 언어 모델을 활용한 재귀적 사고. *사전 인쇄본*, arXiv:2305.14999.

샤오 지홍, 왕 페이이, 주 치하오, 쉬 런신, 송 준샤오, 비 샤오, 장 하오웨이, 장 밍촨, 리 YK, 우 Y, 외 1명. 2024. Deepseek-math: 오픈 언어 모델에서 수학적 추론의 한계를 확장하다. *arXiv 사전 인쇄본 arXiv:2402.03300*.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, Avi-ral Kumar. 2025. 추론을 위한 매개변수 확장보다 LLM 테스트 시간 컴퓨팅의 최적 확장이 더 효과적일 수 있음. *제13회 국제 학습 표현 컨퍼런스*.

송밍양, 수자오천, 취샤오예, 저우자웨이, 청위. 2025. Prmbench: 프로세스 수준 보상 모델을 위한 정밀하고 도전적인 벤치마크. *arXiv 사전 인쇄본 arXiv:2501.03124*.

Qwen Team. 2024a. Qwen2.5: 파운데이션 모델의 집합.

Qwen Team. 2024b. Qwq: 미지의 경계를 깊이 성찰하다.

Guiyao Tie, Zeli Zhao, Dingjie Song, Fuyang Wei, Rong Zhou, Yurou Dai, Wen Yin, Zhejian Yang, Jiangyue Yan, Yao Su, and 1 others. 2025. 대규모 언어 모델의 사후 학습에 관한 연구. *arXiv 사전 인쇄본 arXiv:2503.06072*.

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, Hang Li. 2024. ReFT: 강화 학습을 통한 추론. *제62회 컴퓨터 언어학회 연차 총회 논문집 (제1권: 장문 논문)*, 7601–7614쪽, 태국 방콕. 컴퓨터 언어학회.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, Zhifang Sui. 2024. Math-shepherd: 인간 주석 없이 단계별로 대규모 언어 모델 검증 및 강화. *제62회 전산언어학회 연차대회 논문집(제1권: 장문 논문)*, 9426–9439쪽, 태국 방콕. 전산언어학회.

왕페이이, 리레이, 샤오즈홍, 쉬알엑스, 다이다마이, 리이페이, 천델리, 우위, 수이즈팡. 2023. Math-shepherd: 인간 주석 없이 단계별로 LLM 검증 및 강화하기. *arXiv 사전 인쇄본 arXiv:2312.08935*.

광즈 시옹, 차오 진, 샤오 왕, 인 팡, 하오린 리우, 이판 양, 팡위안 천, 지싱 송, 덩위 왕, 민자 장, 및 1명. 2025. Rag-gym: 프로세스 감독을 통한 추론 및 검색 에이전트 최적화. *arXiv 사전 인쇄본 arXiv:2502.13957*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jian-hong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math 기술 보고서: 자기 개선을 통한 수학 전문가 모델 구축. *arXiv 사전 인쇄본 arXiv:2409.12122*.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024. 생성 검증기: 다음 토큰 예측으로서의 보상 모델링. *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

장전루, 정추제, 우양전, 장베이천, 린런지, 유보원, 류다이하이헝, 저우진궈런, 린준양. 2025. 수학적 추론에서 과정 보상 모델 개발의 교훈. *arXiv 사전 인쇄본 arXiv:2501.07301*.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. 2025. Genprm: 생성적 추론을 통한 프로세스 보상 모델의 테스트 시간 계산 확장. *사전 인쇄본*, arXiv:2504.00891.

정추제, 장전루, 장베이천, 린런지, 루커밍, 위보원, 류다이하이헝, 저우진런, 린준양. 2024. Processbench: 수학적 추론에서의 과정 오류 식별. *arXiv 사전 인쇄본 arXiv:2412.06559*.

Zihao Zhou, Shudong Liu, Maizhen Ning, Wei Liu, Jindong Wang, Derek F. Wong, Xiaowei Huang, Qi-ufeng Wang, Kaizhu Huang. 2025. Is your model really a good math reasoner? evaluating mathematical reasoning with checklist. In *The Thirteenth International Conference on Learning Representations*.

A 실험 세부 사항

하위 범주의 약어 실험에 사용된 약어의 전체 명칭은 표 5에 나와 있습니다.

약어	전체 이름	추론 패턴
TT.	변환 불일치	변환
TF.	변형 반사실성	변형
DC.	분해 불완전성	분해
DR.	분해 중복성	분해
DS.	분해 부적합성	분해
GP.	재수집 부정확성	재수집
GC.	재수집 불완전성	재수집
GR.	재모음 중복	재수집
CE.	수정 오류	검증
DE.	탐지 오류	검증
CF.	결론 가설적 사실성	추론
CT.	결론 불일치	추론
CV.	결론 무효성	추론
PC.	전제 불완전성	추론
PR.	전제 중복성	추론
PS.	전제 비합리성	추론
IC.	통합 불완전성	통합
IT.	통합 불일치	통합
IR.	통합 중복성	통합
IS.	통합 비합리성	통합

표 5: 약어의 전체 명칭.

구현 세부사항 소크라테스식 추론 모델 훈련을 위해, 우리는 LLaMA-Factory 라이브러리[2]를 사용하여 Qwen2.5-72B-Instruct를 미세 조정하기 위해 LoRA 튜닝(Hu et al., 2021)을 사용합니다. 오픈소스 PRM 평가를 위해, 우리는 구현을 위해 PRM Eval ToolKit[3]을 활용합니다. 비평가 모델로 프롬프트된 LLM 평가를 위해, 기본 온도 1.0으로 설정된 상태에서 표 6의 프롬프트 템플릿을 사용하여 LLM에 프롬프트를 제공합니다. 테스트 케이스 구성 절차 중, 메타데이터 세트 $D^{()에서}N = 150$ 개의 샘플을 선택합니다. 여기에는 GSM8k에서 10개 샘플, Omni-Math, MathBench, OlympiadBench에서 각각 50개 샘플이 포함됩니다.

B 프롬프트

제3절에서 설명한 바와 같이, 대규모 언어 모델(LLM)은 본 방법론에서 핵심적인 역할을 수행합니다. 소크라테스식 추론 큐레이션 단계에서 단계별 검증용 프롬프트는 표 7에 제시되어 있습니다. 테스트 케이스 구성 단계에서는 (Song et al., 2025)를 따르며, 각각 표 8과 표 9에 표시된 대로 작업 프롬프트와 출력 형식 프롬프트를 별도로 설계합니다. LLM 기반 필터링 절차에는 표 10의 프롬프트 템플릿을 사용합니다.

[2] https://github.com/hiyouga/LLaMA-Factory [3]https://github.com/ssmisya/PRMBench

비평가 모델로 프로ンプ팅된 LLM 평가용 프롬프트 템플릿

[시스템 프롬프트]
당신은 수학적 추론 평가자입니다. 당신의 임무는 수학 문제 해결 단계를 분석하고 JSON 형식으로 구조화된 평가를 제공하는 것입니다.

각 해결 단계마다 유효성 점수(-1에서 +1)를 평가해야 합니다:
* +1: 완전히 정확한 수학적 추론
* 0: 일부 오류가 있는 부분적으로 올바른 추론
* -1: 완전히 틀림
* 정확도의 정도를 나타내기 위해 그 사이의 값을 사용하십시오

요구 사항:
- 각 단계를 독립적으로 평가하십시오
- 점수를 부동 소수점 숫자로 제공하십시오
- 엄격한 JSON 형식으로 결과 반환: {"validity ": [scores]}
- 배열의 길이가 단계 수와 동일하도록 하십시오
- 평가 시 수학적 엄밀성을 유지하십시오
- 수학적 정확성, 논리적 일관성, 해결 효율성을 고려하십시오. 예시 출력 형식:

{"validity ": [0.8, -0.5, 1.0]}

수학 문제와 단계별 해결 과정이 제시됩니다. 각 단계를 분석하여 지정된 JSON 형식으로 평가를 제공하십시오.

[사용자]
질문: {question} 해법:

{solution}

표 6: 비평가 모델로 프로ンプ팅된 LLM 평가용 프롬프트 템플릿

단계 검증용 프롬프트 템플릿

당신은 추론 과정 검증 전문가입니다. 질문, 해결책(단락별로 분할, 태그로 묶고 1부터 인덱싱), 그리고 참조 답변이 제공될 것입니다.

[질문]
{질문}

[해결책]
{해결책}

[참고 답변]
{답변}

여러분의 임무는 해결 방안을 단락별로 검토하고 비판하는 것입니다. 단락에서 오류를 발견하면 가장 먼저 발생하는 오류가 있는 단락의 인덱스를 반환하십시오. 그렇지 않으면 -1 인덱스(일반적으로 "찾을 수 없음"을 의미함)를 반환하십시오. 최종 답변(즉, 인덱스)을 \boxed{} 안에 넣어 주십시오.

표 7: 단계 검증용 프롬프트 템플릿.

테스트 케이스 구축을 위한 작업 프롬프트

당신은 추론과 데이터 구축에 매우 능숙한 유용한 AI 어시스턴트입니다. 이제 추론 과정 내 단계의 정확성을 판단하는 프로세스 수준 보상 모델의 능력을 테스트하고자 합니다. 이를 위해 주어진 추론 과정에 특정 유형의 오류를 도입하여 결함이 있는 사례를 구축하는 데 도움을 주세요.
다음과 같은 것이 제공됩니다:
1. 수학 문제.
2. 이를 해결하기 위한 올바른 단계별 추론 과정. 각 단계는 [변환], [분해], [재구성], [추론], [검증], [통합], [답변], [L검증], [G검증]을 포함할 수 있는 '행동' 형태로 구성됩니다.

각 단계의 설명은 다음과 같습니다:

[변환] (식별자: <Repeat>xxx</Repeat>)
- 문제 해결 관점에서 문제를 설명합니다.
- 문제를 재구성하여 보다 포괄적이고 명확하게 이해합니다

[분해] (식별자: <Decomposition>xxx</Decomposition>)
- 문제를 여러 핵심 하위 문제로 분해하십시오; 각 하위 문제를 해결함으로써 주요 문제를 해결하십시오
- 분해가 필요하지 않은 경우 해결 접근법을 제시하십시오

[재수집] (식별자: <Regather>xxx</Regather>)
- 문제 해결과 관련된 입력의 핵심 정보를 수집하십시오
- 문제 해결과 관련된 정의, 원칙 및 기타 개념을 출력하고 설명을 제공하십시오

[추론] (식별자: <Deduction>xxx</Deduction>)
- 기존 정보를 관찰하고 핵심 부분을 추출하십시오
- 제약 조건과 한계를 고려하여 명시적 및 암묵적 요구사항을 식별합니다
- 문제를 해결하기 위한 구체적인 아이디어를 제안하라
- 아이디어에 따라 추론을 실행하라

[LVerification]&[GVerification] (식별자: <L(G)Verification>xxx</L(G)Verification>)
- 추론 과정의 논리적 일관성을 검증하라
- 기존 증거에 대한 추론 과정 점검
- 추론 과정의 잠재적 결함을 찾아 개선하라
- 이해의 완전성을 검토하라
- 가정을 의심하고 대안적 관점을 고려하라
- [L검증]은 모든 추론 단계 이후에 발생할 수 있으며, 부분 단계를 검증합니다
- [GVerification]은 [통합] 단계와 [답변] 단계 사이에서만 발생하며, 전체 과정을 검증합니다

[통합] (식별자: <통합>xxx</통합>)
- 현재의 모든 추론 과정을 통합하여 현재의 결론을 형성한다

[답변] (식별자: <Answer>xxx</Answer>)
- 원래 문제에 대한 최종 답안을 출력하라

당신의 임무는 질문을 수정하거나, 원래 단계를 조정하거나, 원래 과정 체인에 추가 단계를 도입하여 그럴듯해 보이지만 잘못된 추론 과정을 만들어 잘못된 답을 도출하는 것입니다. 목표는 '### 도입할 오류 유형' 뒤에 명시된 오류를 포함시켜 결함이 있는 해결책을 시뮬레이션하는 것입니다.

도입할 오류 유형
{오류 유형}

표 8: 테스트 케이스 구축을 위한 작업 프롬프트.

테스트 케이스 구축을 위한 출력 형식 프롬프트

서식 지침:

수정 후 다음 구조화된 출력을 제공하십시오:

```
{
   "original_question": "원래의 수학 문제.", "modified_question ": "수정된 문제 또는 원본 문제",
   "original_process": ["원본 단계 1", "원본 단계 2", . . . ], "modified_process": ["수정된 단계 1", "수정
   된 단계 2", . . . ], "modified_steps": [1, 5, 7, . . . ],
   "error_steps": [5, 6, . . . ],
   "reason": "변경 사유 설명."
}
```

상세 요구사항:

1. original_question: 제공된 원래 수학 문제를 나타내는 문자열.
2. modified_question: 변경 후 수정된 문제를 나타내는 문자열. 문제가 동일하게 유지되는 경우 원본 질문을 복사할 수 있습니다.
3. original_process: 입력으로 제공된 원래 추론 단계를 나타내는 비어 있지 않은 문자열 목록
4. 수정된_과정: 사용자의 수정 후 추론 과정을 나타내는 비어 있지 않은 문자열 목록.
5. 수정된 단계: 수정된 모든 단계의 인덱스를 나타내는 비어 있지 않은 정수 목록입니다. 인덱싱은 1에서 시작합니다.
인덱싱은 1부터 시작합니다.
6. error_steps: 환각이나 오류를 포함하는 단계를 나타내는 비어 있지 않은 정수 목록입니다. 이 단계들은 modified_steps에도 포함되어야 합니다.
7. reason: 수정 내용, 수정 이유 및 수정 방식에 대한 명확한 설명.
지정된 오류 유형과 일치합니다.

참고 사항:

1. 모든 목록은 비어 있지 않아야 합니다.
2. 모든 수학 기호는 LaTeX 형식(예: x^2는 x^2로 표기)을 사용하십시오. \u2248이나 \u00f7 같은 유니코드 기호는 사용하지 마십시오.
3. JSON 객체가 올바르게 구성되었는지 확인하십시오. 백슬래시 n과 같은 특수 문자에 대한 적절한 이스케이프 처리가 필요합니다(예: 줄바꿈은 \n 사용).
4. 모든 인덱스는 1부터 시작합니다. 즉, 첫 번째 단계의 인덱스는 0이 아닌 1입니다.
5. 질문을 수정할지 여부를 선택할 수 있습니다. 질문이 동일하게 유지된다면 원본 질문을 복사할 수 있습니다. 그러나 질문이 수정된 경우, 단계별 평가가 수정된 질문을 기준으로 이루어지도록 해야 합니다.
6. 프롬프트에서 제공된 원본 프로세스를 그대로 제시하십시오. 수정하지 마십시오.

테이블 9: 테스트 케이스 구축을 위한 출력 형식 프롬프트.

LLM 기반 필터링을 위한 프롬프트 템플릿

귀하는 추론 과정 검증 전문가입니다. 질문과 해결책(단락별로 구분되어 태그로 묶임)이 제공됩니다.

귀하의 임무는 대규모 언어 모델(LLM)이 생성한 단계별 해결책이 다음을 충족하는지 판단하는 것입니다:
1. LLM이 생성한 과정은 발생할 수 있는 가능한 해결 경로처럼 보입니다.
2. LLM이 생성한 과정은 완전히 잘못되었으며, 오류 유형은 [분류]에 대한 설명에 적합합니다.

[분류]
{분류}

[질문]
{질문}

[해결책]
{해결책}

두 가지 측면이 모두 충족되면 '예'라고 답하고, 그렇지 않으면 '아니오'라고 답하십시오. 최종 답변(예 또는 아니오)을 \boxed {} 안에 기입하십시오.

표 10: LLM 기반 필터링을 위한 프롬프트 템플릿