

Towards a Fine-Grained Benchmark for Implicit Reasoning in Large Language Models

2025. 10. 23.

양 건

Alice in Wonderland: Simple Tasks Showing Complete Reasoning Breakdown in State-Of-the-Art Large Language Models

Marianna Nezhurina^{1,2,4*} **Lucia Cipolina-Kun**^{1,2,3} **Mehdi Cherti**^{1,2,4} **Jenia Jitsev**^{1,2,4*}

¹LAION ²Juelich Supercomputing Center (JSC), Research Center Juelich (FZJ)

³ School of Electrical and Electronic Engineering, University of Bristol

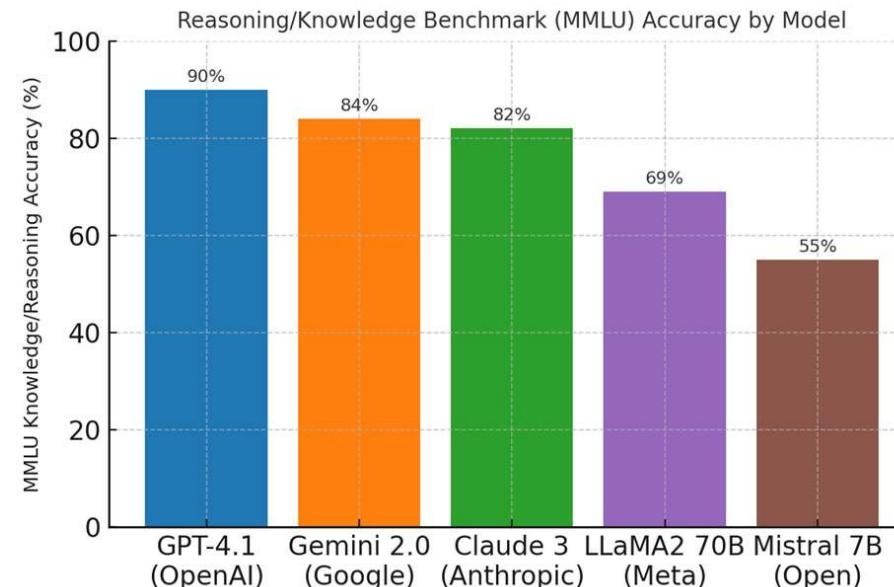
⁴ Open-Ψ (Open-Sci) Collective

*Corresponding authors: {m.nezhurina, j.jitsev}@fz-juelich.de, contact@laion.ai

<NeurIPS SciDL Workshop 2024>

Our Perception of LLMs

- 대규모 언어 모델(LLMs)에 대한 현재 인식
 - 강력한 generalization 능력을 보유 → 다양한 과제/조건에서 few-shot/zero-shot transfer에 robust 함
 - Scaling Laws를 따름 → 사전학습 규모가 커질수록 성능 (generalization 능력)이 향상
- 인식을 뒷받침하는 근거
 - standardized benchmarks (MMLU, HellaSwag, GSM8K 등) = 일반화/추론 능력을 잘 반영한다고 가정
 - 해당 벤치마크들에서의 점수가 모델의 추론 능력을 나타내는 척도 (점수 = 성능)



Are We Overestimating LLMs' Capabilities?

- Rising Doubts: Do Benchmark Scores Reflect True Ability?
 - 논란의 원인 1: Test Data Contamination
 - benchmark 데이터셋이 그 이후 나온 모델들의 학습 데이터로 사용되었다는 지적
→ Model이 '암기'한 답을 말하는 것일 수 있음
 - 논란의 원인 2: Discrepancy between Claimed and Actual Capabilities
 - 벤치마크에서 높은 점수를 받은 모델이 단순한 task 추론에 실패함을 보여주는 선행 연구들 존재
↳ 반론: Prompt engineering, Self-Verification, Majority Voting 등 simple intervention으로 실패 완화 가능
 - Model의 fundamental deficit보다 ill-posed problem formulation에서 비롯되었을 가능성

Research Questions

- RQ 1 – Robustness on Generalization
 - 사람에게는 아주 쉽고 단순한 commonsense 문제와, 해당 문제에서 숫자만 바꾼 variation에서 LLM이 일관되게 정답을 내는가?
- RQ 2 – Source of Failure
 - Task에서의 실패가 LLM의 근본적인 일반화/추론 능력 결함을 보여주는 것인가 아니면 프롬프트 수정 등으로 쉽게 해결될 수 있는 부차적인 문제인가?
- RQ 3 – Validity of current Standard Evaluation Methods
 - 현존하는 standardized benchmarks에서의 고득점이 이런 단순 추론 task에서의 성능을 잘 represent 하는가?

The “Alice in Wonderland” Challenge: A Reality Check

- 복잡한 벤치마크 대신, 단순하고 명확한 문제를 통해 모델의 근본 능력을 검증

Template

“Alice has **N** brothers and she also has **M** sisters. How many sisters does Alice’s brother have?”

Answer = **M** + 1 (Alice 포함 자매 수)

- 문제의 논리적 구조와 난이도는 동일하지만, 숫자만 다른 variation 생성

Variation 1. Alice has 3 brothers and she also has 6 sisters. [Correct answer: 7]

Variation 2. Alice has 2 sisters and she also has 4 brothers. [Correct answer: 3]

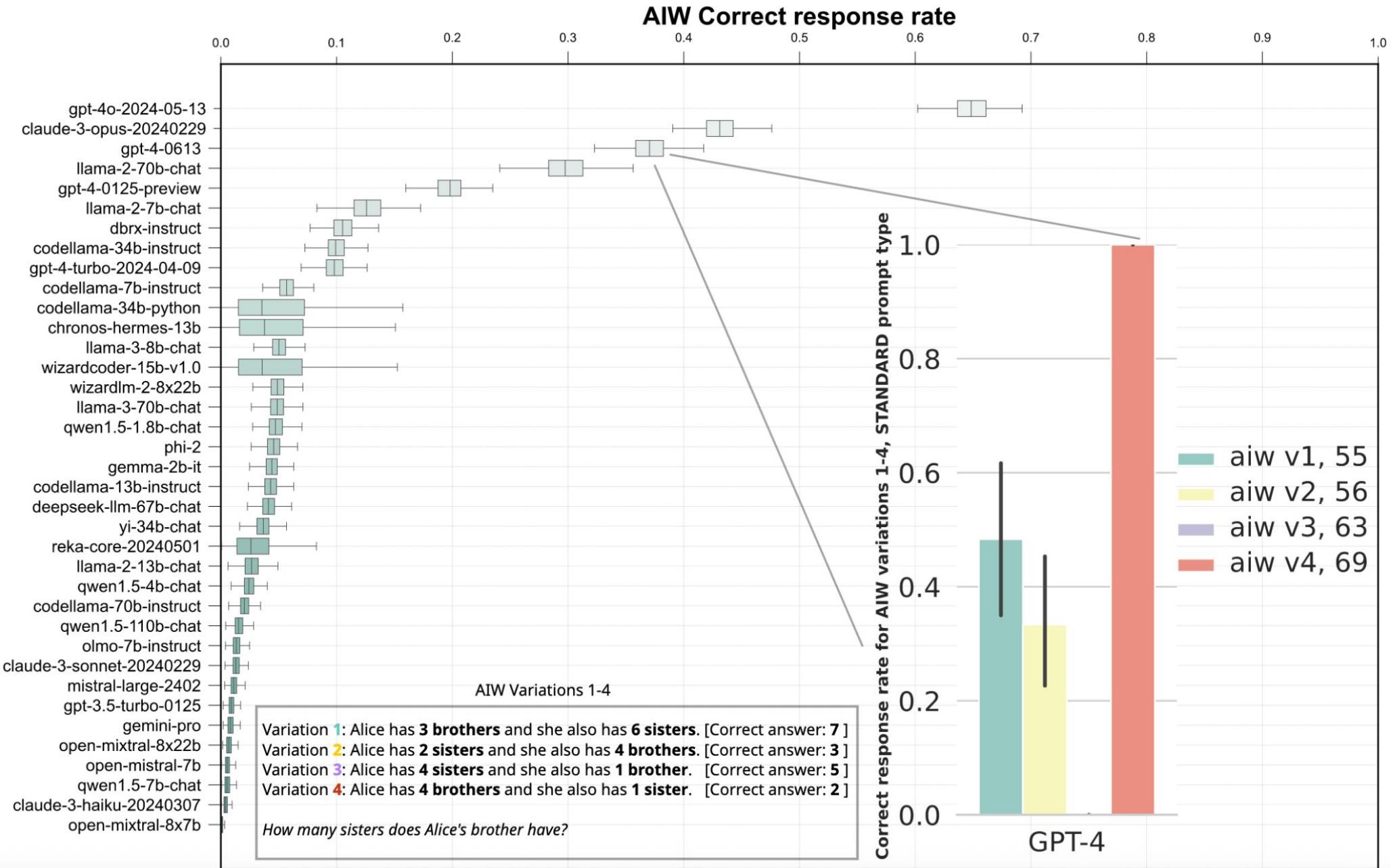
Variation 3. Alice has 4 sisters and she also has 1 brother. [Correct answer: 5]

Variation 4. Alice has 4 brothers and she also has 1 sister. [Correct answer: 2]

- 진정한 generalization 능력을 가진 모델이라면, 이러한 변화에 영향을 받지 않고 안정적으로 높은 성능을 유지해야 함

Methods & Experiment Setup

- 3가지 프롬프트를 사용하여 프롬프트 차이에 따른 성능 변화 확인
 - Standard (최종 답변 양식만 고정)
Solve this problem and provide the final answer in following form: "### Answer: "
 - Thinking (CoT 사용 유도)
Before providing answer to this problem, think carefully and double check the path to the correct solution
for any mistakes. Provide then the final answer in following form: "### Answer: "
 - Restricted (최종 답변 외 response token 생성 금지)
To answer the question, DO NOT OUTPUT ANY TEXT EXCEPT following format that contains final answer:
"### Answer: "
- Control Problems for Fine–Grained Diagnosis (“AIW Light”)
 - Original AIW 문제가 요구하는 추론 요소를 줄여 성능 변화의 요인 분석
 - Arithmetic Siblings: “Alice의 siblings 수?” → M+N (Simple arithmetic operation 능력 측정)
 - Family: “Alice의 sister가 가진 brothers 수?” → N (Basic family structure 이해 능력 측정)
 - Arithmetic Total Girls: “Family 내 girls 총 몇 명?” → M+1



Results: The Breakdown of SOTA Models

AIW Variations 1-4

Variation 1: Alice has **3 brothers** and she also has **6 sisters**. [Correct answer: 7]
Variation 2: Alice has **2 sisters** and she also has **4 brothers**. [Correct answer: 3]
Variation 3: Alice has **4 sisters** and she also has **1 brother**. [Correct answer: 5]
Variation 4: Alice has **4 brothers** and she also has **1 sister**. [Correct answer: 2]

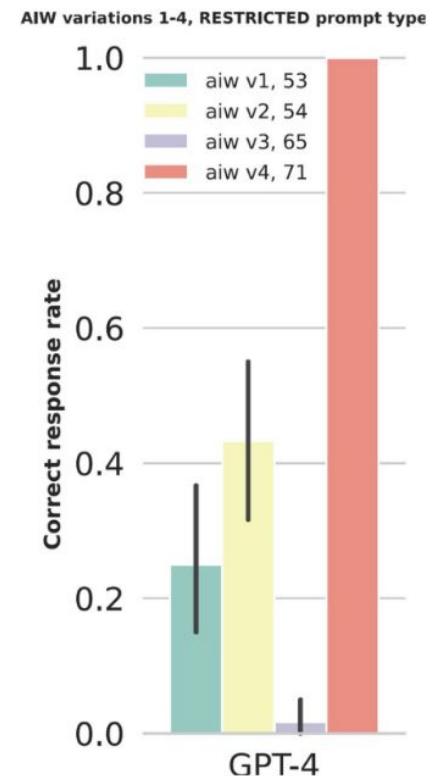
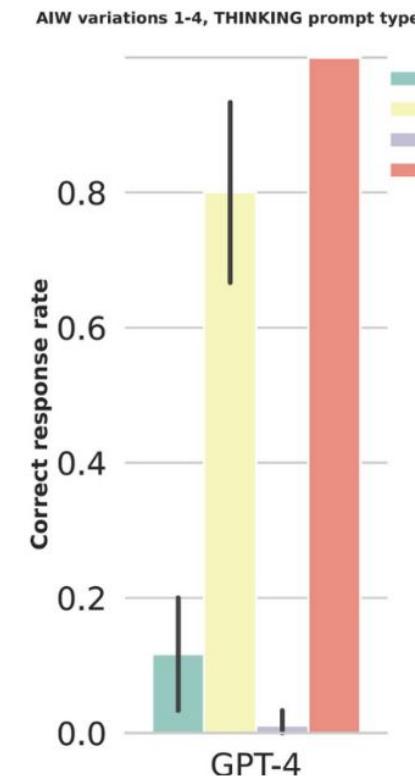
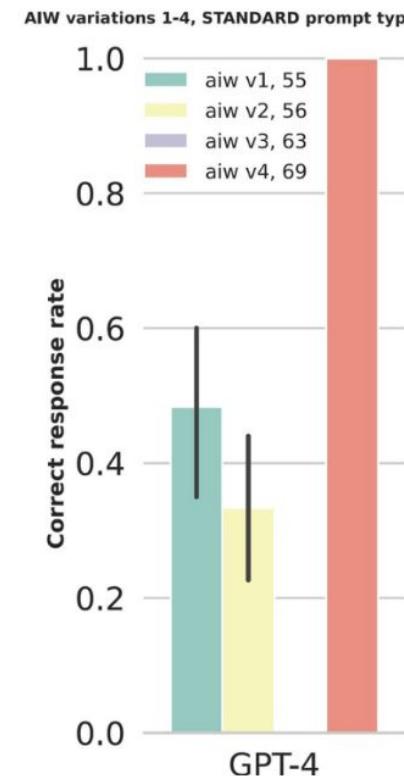
How many sisters does Alice's brother have?

Prompt types (following after main problem description and question above):

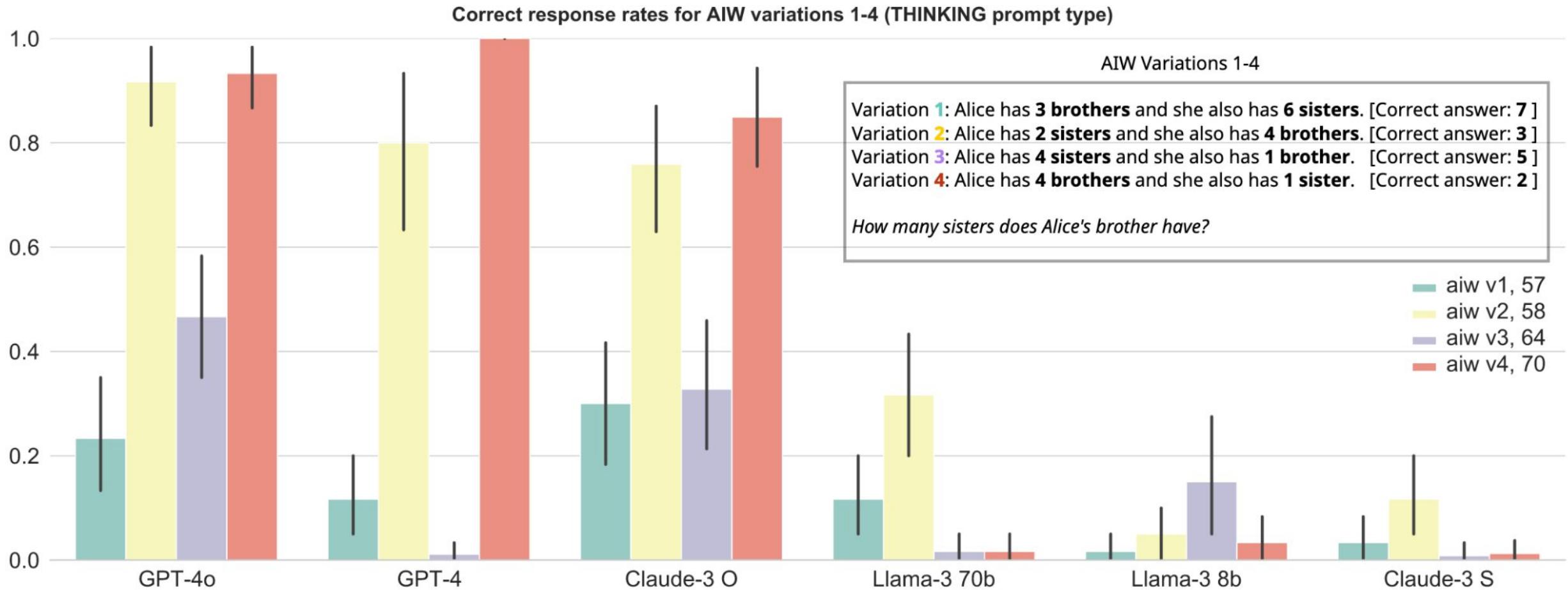
STANDARD : Solve this problem and provide the final answer in following form: "### Answer: ".

THINKING [v2] : Before providing answer to this problem, think carefully [step by step] and double check the path to the correct solution for any mistakes. Provide then the final answer in following form: "### Answer: ".

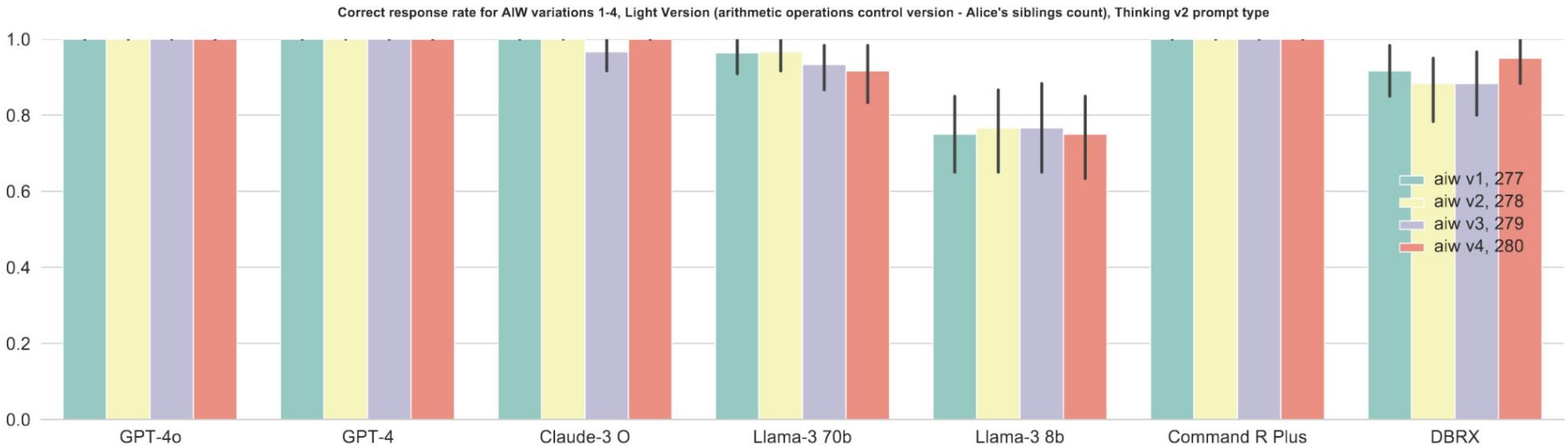
RESTRICTED : To answer the question, DO NOT OUTPUT ANY TEXT EXCEPT following format that contains final answer: "### Answer: ".

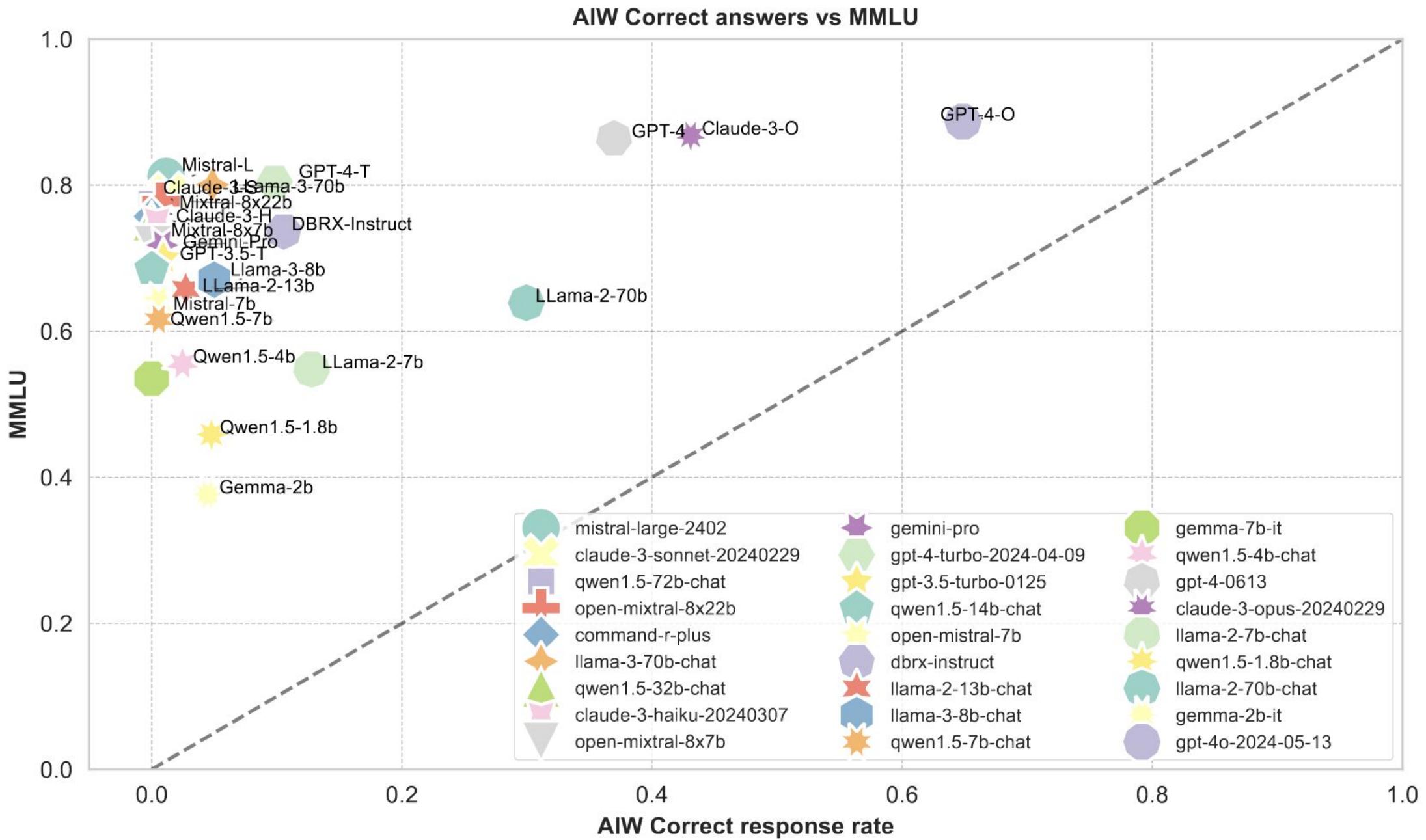


Results: Standard Techniques Fail



Results: Fine-Grained Diagnosis using Control Problems





Conclusion

- A Fundamental Flaw in Generalization & Reasoning
 - Problem irrelevant perturbations 에도 큰 성능 fluctuation 을 보임
→ fragile generalization (robustness 결여)
 - Control experiment (AIW Light)을 통해,
failure의 원인이 natural language parsing, arithmetic 등의 low-level issue가 아님을 확인
→ compositional generalization과 basic logical reasoning 자체의 fundamental flaw
- Contradiction with Claims
 - GPT-4, Claude 3 Opus 등이 graduate-level problem을 푼다고 주장하지만,
AIW 같은 단순한 문제에도 robust하게 답하지 못함
- Various standard interventions Fail
 - Chain-of-Thought(CoT) 와 같은 기법도 이 fundamental problem을 해결하지 못함
- Scale Helps, But Doesn't Solve
 - 가장 큰 model들(GPT-4, Claude 3 Opus)이 relatively higher performance를 보이지만,
해당 모델들의 reasoning도 fragile and unstable (여전히 problem irrelevant perturbations에 easily disturbed)

Conclusion (cont.)

- Need for a New Evaluation Paradigm
 - MMLU, GSM8k에서 high score를 받는 model이 AIW problem에서는 ~0% accuracy를 보임
이는 current standard evaluation method가 model의 true reasoning deficit을 detect하지 못함을 의미
 - standardized benchmarks에서의 고득점이 AIW 성능을 예측하지 못함; 순위가 뒤바뀌는 현상 관찰
 - 단순한 final-answer scoring을 넘어, reasoning process의 robustness를 평가해야 함
 - robustness/generalization을 명시적으로 측정하는 새로운 벤치마크 설계 필요
- Need for Re-assessment of current LLMs' Abilities

Progressive-Hint Prompting Improves Reasoning in Large Language Models

Chuanyang Zheng¹, Zhengying Liu², Enze Xie³, Zhenguo Li⁴, Yu Li¹

¹CUHK, ²Paris XI, ³HKU, ⁴HKUST
`{cyzheng21}@cse.cuhk.edu.hk`

<ICML AI4MATH 2024>

Existing Prompting Methods

- 모델의 성능을 향상시키기 위한 기존의 대표적인 Prompting 기법들
 - Chain-of-Thought (CoT) Prompting
모델이 자신의 reasoning process 를 step-by-step 설명하도록 지시
 - Self-Consistency
정답에 도달할 수 있는 다양한 추론 경로 Sampling
→ 여러 방식으로 산출한 답 중 Majority Voting 을 통해 최종 답변 선택
- 추론 경로가 아닌, 모델이 생성한 '**답변**'을 Prompting 기법에 활용할 수는 없을까?

Leveraging Model Answers for Self–Correction

- **Progressive-Hint Prompting (PHP)**
 - 직관: 수학 시험 볼 때 시간이 남으면, 풀었던 문제를 다시 풀어보며 기존과 동일한 답이 나오는지 확인하는 과정
→ 모델에게 이전 답을 단서 삼아 다시 추론 시킴
 - PHP Framework
 - Stage 1: 모델에게 질문하여 Base Answer 생성
 - Stage 2: Base Answer를 다음 Round에 Hint로 제공하여 재질문, Subsequent Answer 생성
 - Stage 3: 기존 Hint 와 다른 답변이 생성되면 Hint에 추가 후 (Progressive Hinting) Stage 2 반복
→ 점진적으로 정답에 수렴하도록 유도하는 과정
 - Stopping Rule: 두 번 연속으로 동일한 답변을 생성하면 최종 답변에 converge 했다고 간주하고 종료
 - CoT · Self–Consistency 기법들과 함께 사용 가능
- Research Questions
 - PHP 기법을 사용하면 수학 추론 벤치마크에서 모델 성능이 향상되는가?
 - CoT, self–consistency와 병행할 때도 성능 향상이 유지되는가?

Progressive-Hint Prompting Example

Base Prompting

Q: There are 96 fourth-graders at Small Tree School. 43 of them are girls. On Friday, 5 fourth-grade girls and 4 fourth-grade boys were absent. How many fourth grade boys were at Small Tree School on Friday?

LLM

Base Answer

A: There were 96 fourth-graders at Small Tree School. 43 of them were girls. On Friday, 5 girls and 4 boys were absent. So there were $43 - 5 = 38$ girls at school on Friday, and $96 - 38 = 58$ boys at school on Friday. The answer is **58**.

Progressive-Hint Prompting

Q: There are 96 fourth-graders at Small Tree School. 43 of them are girls. On Friday, 5 fourth-grade girls and 4 fourth-grade boys were absent. How many fourth grade boys were at Small Tree School on Friday? **(Hint: The answer is near to 58).**

LLM

Second Answer

A: **We know the Answer Hints: 58. With the Answer Hints: 58, we will answer the question.** There were 96 fourth-graders at Small Tree School, and 43 of them were girls. On Friday, 5 girls and 4 boys were absent. So there were $43 - 5 = 38$ girls and $96 - 43 = 53$ boys at Small Tree School on Friday. After subtracting the absent students, there were $38 - 4 = 34$ girls and $53 - 5 = 48$ boys. The answer is **48**.

Progressive-Hint Prompting

Q: There are 96 fourth-graders at Small Tree School. 43 of them are girls. On Friday, 5 fourth-grade girls and 4 fourth-grade boys were absent. How many fourth grade boys were at Small Tree School on Friday? **(Hint: The answer is near to 58, 48).**

LLM

Third Answer

A: **We know the Answer Hints: 58, 48. With the Answer Hints: 58, 48, we will answer the question.** There were 96 fourth-graders, and 43 of them were girls. On Friday, 5 girls and 4 boys were absent. So there were $43 - 5 = 38$ girls and $96 - 9 = 87$ students in total. Of those 87 students, 38 were girls, so there must have been $87 - 38 = 49$ boys. The answer is **49**.

Progressive-Hint Prompting

Q: There are 96 fourth-graders at Small Tree School. 43 of them are girls. On Friday, 5 fourth-grade girls and 4 fourth-grade boys were absent. How many fourth grade boys were at Small Tree School on Friday? **(Hint: The answer is near to 58, 48, 49).**

LLM

Fourth Answer

A: **We know the Answer Hints: 58, 48, 49. With the Answer Hints: 58, 48, 49, we will answer the question.** There were 96 fourth-graders at Small Tree School, and 43 of them were girls. On Friday, 5 girls and 4 boys were absent. So there were $43 - 5 = 38$ girls and $96 - 43 = 53$ boys originally. After the absences, there were $38 - 5 = 33$ girls and $53 - 4 = 49$ boys. The answer is **49**.

Experimental Setup

- 수학 추론 벤치마크 7개 선정
 - AddSub, MultiArith, SingleEQ, SVAMP, GSM8K, AQuA and MATH
- 평가 모델 4개 선정
 - text-davinci-002 / text-davinci-003 / GPT-3.5-Turbo / GPT-4
- The 3x3 Design (Base × PHP)
 - PHP의 효과를 종합적으로 평가하기 위해
 - Initial Answer 를 생성하는 Base Prompt
 - Subsequent Answer 를 생성하는 PHP Prompt를 조합한 총 9가지 실험 조건 구성
 - Base Prompt (초기 응답): Standard / CoT / Complex-CoT
 - PHP Prompt (후속 응답): PHP-Standard / PHP-CoT / PHP-Complex-CoT

Experimental Setup (Cont.)

- Possible outcomes – 모델이 직면할 수 있는 두 가지 상황을 모두 처리할 수 있는지 평가
 - Case 1: Correct Hint
 - 목표: 주어진 정답 힌트를 바탕으로 정답을 안정적으로 유지하는가
 - 검증: 모델의 답변 안정성(Stability)
 - Case 2: Incorrect Hint
 - 목표: 모델이 주어진 오답 힌트에 혼혹되지 않고, 스스로 재추론하여 오답에서 탈출할 수 있는가
 - 검증: 모델의 자기 수정 능력(Self-Correction Ability)

Result 1: PHP Improves Accuracy

- More Powerful Models → Better PHP Performance
- More Powerful Prompts → Better PHP Performance

	Prompt	PHP	Dataset						Average
			AddSub	MultiArith	SingleEQ	SVAMP	GSM8K	AQuA	
GPT-3.5 text-davinci-002	Standard [8]	✗	79.4	34.0	80.7	64.8	15.1	25.5	49.91
		✓	80.5	31.8	79.9	64.2	14.7	25.5	49.43
			(+1.1)	(-2.2)	(-0.8)	(-0.6)	(-0.4)	(0.0)	(-0.48)
	CoT [8]	✗	85.8	89.1	89.7	72.9	49.5	44.4	71.89
		✓	86.8	89.0	90.1	72.3	51.1	45.6	72.48
			(+1.0)	(-0.1)	(+0.4)	(-0.6)	(+1.6)	(+1.2)	(+0.59)
	Complex CoT [10]	✗	82.5	89.8	87.7	70.4	57.6	37.4	70.89
		✓	83.7	90.1	89.9	74.6	61.2	37.0	72.75
			(+1.2)	(+0.3)	(+2.2)	(+4.2)	(+3.6)	(-0.4)	(+1.86)
GPT-3.5 text-davinci-003	Standard [8]	✗	89.1	36.3	83.8	68.7	15.9	28.3	53.68
		✓	89.1	36.0	83.6	68.7	16.0	28.3	53.61
			(0.0)	(-0.3)	(-0.2)	(0.0)	(+0.1)	(0.0)	(-0.07)
	CoT [8]	✗	90.6	93.6	92.7	81.0	56.1	44.0	76.33
		✓	91.1	94.0	93.5	81.3	57.5	44.4	76.96
			(+0.5)	(+0.4)	(+0.8)	(+0.3)	(+1.4)	(+0.4)	(+0.63)
	Complex CoT [10]	✗	86.3	94.8	91.5	77.4	67.0	48.8	77.63
		✓	88.1	95.0	94.0	80.0	71.6	50.0	79.78
			(+1.8)	(+0.2)	(+2.5)	(+2.6)	(+4.6)	(+1.2)	(+2.15)

Result 2: The Impact of Hint Quality

PHP	Base Prompt	Dataset						Average
		AddSub	MultiArith	SingleEQ	SVAMP	GSM8K	AQuA	
PHP-Standard	Standard [8]	89.1	36.0	83.6	68.7	16.0	28.3	53.61
	CoT [8]	92.4	80.5	92.1	78.5	50.2	42.5	72.70
	Complex CoT [10]	90.6	80.6	92.9	77.2	60.3	45.6	74.53
PHP-CoT	Standard [8]	90.8	92.5	90.7	80.2	52.3	40.9	74.56
	CoT [8]	91.1	94.0	93.5	81.3	57.5	44.4	76.96
	Complex CoT [10]	90.6	96.8	93.7	81.2	62.6	50.0	79.14
PHP-Complex CoT	Standard [8]	88.3	80.1	93.3	80.4	65.5	35.4	73.83
	CoT [8]	88.8	95.6	94.8	81.4	70.6	45.6	79.46
	Complex CoT [10]	88.1	95.0	94.0	80.0	71.6	50.0	79.78

- Hint Quality is Key
 - 초기 답변(Base Answer)의 질이 좋을수록, 즉 힌트가 정답에 가까울수록 PHP의 최종 성능이 높아짐

Result 3: Correct vs. Incorrect Hints

Method	Hint		Dataset						Average
	Correct	Incorrect	AddSub	MultiArith	SingleEQ	SVAMP	GSM8K	AQuA	
	X	X	90.6	93.6	92.7	81.0	56.1	44.0	76.33
CoT [8]	✓	X	91.6	94.3	93.3	81.9	57.0	43.7	76.96
	X	✓	91.1	93.5	93.1	79.7	57.9	45.2	76.74
	✓	✓	91.1	94.0	93.5	81.3	57.5	44.4	76.96
Complex CoT [10]	X	X	86.3	94.8	91.5	77.4	67.0	48.8	77.63
	✓	X	88.3	94.0	93.8	77.8	68.6	46.4	78.14
	X	✓	88.1	94.6	94.0	79.2	70.2	48.4	79.08
	✓	✓	88.1	95.0	94.0	80.0	71.6	50.0	79.78

- Ablation Study
 - 프롬프트 설계에 Correct/Incorrect 힌트를 의도적으로 주입한 여러 variation을 만들어 성능 비교
 - 정답 / 오답 힌트를 모두 제공 했을 때 가장 우수한 성능을 보임
- Self-Correction is NOT Guaranteed
 - 오답 힌트의 경우에도 품질이 중요 → 정답과 너무 동떨어진 오답을 제공할 경우 오답으로 수렴할 가능성 높음
 - 더 Powerful 한 prompt (Complex CoT) 를 쓰면 오답 탈출을 ‘촉진’할 가능성을 보여주기는 함

Result 4: Self-Consistency + PHP: Accuracy ↑ & Cost ↓

Prompt	SC	PHP	Dataset						Average
			AddSub	MultiArith	SingleEQ	SVAMP	GSM8K	AQuA	
CoT [8]	5	✗	90.6	95.3	94.4	81.6	63.3	49.2	79.06
	5	✓	90.8	96.6	94.8	83.5	66.3	49.6	80.26
	5	Number	2.0075	2.0433	2.0098	2.1090	2.5458	2.0157	2.1218
	10	✗	90.6	96.5	93.8	83.0	65.5	49.2	79.76
	10	✓	90.8	97.1	93.8	83.5	67.5	50.0	80.45
	10	Number	2.0075	2.0283	2.0059	2.0510	2.2145	2.0118	2.0531
	20	✗	91.1	96.5	94.2	83.3	68.0	55.1	81.36
	20	✓	91.6	96.5	94.4	83.7	68.6	55.1	81.64
	20	Number	2.0050	2.0366	2.0098	2.0250	2.1144	2.0078	2.0330
	40	✗	91.6	96.5	94.8	82.9	67.3	53.1	81.03
Complex CoT [10]	40	✓	91.6	96.6	95.0	83.7	68.4	53.1	81.39
	40	Number	2.0050	2.0300	2.0050	2.0320	2.0530	2.0000	2.0208
	5	✗	88.1	97.0	93.1	80.4	73.5	51.5	80.60
	5	✓	89.6	97.3	95.2	82.5	76.9	51.9	82.23
	5	Number	2.0378	2.0166	2.0334	2.2370	2.5390	2.0118	2.1459
	10	✗	88.6	98.3	93.3	82.4	76.4	54.3	82.21
	10	✓	89.1	98.5	95.2	83.4	78.2	54.7	83.18
	10	Number	2.0177	2.0016	2.0295	2.059	2.1531	2.0078	2.0447
	20	✗	88.6	98.0	93.8	82.5	77.7	56.2	82.80
	20	✓	89.8	98.0	95.8	83.6	78.6	56.2	83.66
	20	Number	2.0253	2.0000	2.0196	2.0330	2.0401	2.0000	2.0196
40	✗	88.3	98.5	94.8	83.9	78.1	58.6	83.70	
	✓	88.6	98.5	95.8	84.7	79.0	58.6	84.20	
	Number	2.0101	2.0000	2.0137	2.0210	2.0348	2.0039	2.0137	

Next Steps: Evolving the Evaluation of Implicit Reasoning

- 새로운 벤치마크 필요에 대한 당위성
 - 현존하는 Implicit Reasoning / Knowledge 벤치마크들은 Question에 정답 (최종 목표)에 관한 keywords 가 명시되어 있음
 - 추론의 핵심이 되는 키워드가 질문 속에 노출되어 있기 때문에, 모델은 추론의 출발점과 방향성을 이미 제공받은 채로 문제를 풀게 됨
 - 진정한 Implicit Reasoning 능력을 보려면 질문도 Implicit하게 만들어야하지 않을까?

Variation 1. Alice has 3 brothers and she also has 6 sisters. [Correct answer: 7]

Variation 2. Alice has 2 sisters and she also has 4 brothers. [Correct answer: 3]

Variation 3. Alice has 4 sisters and she also has 1 brother. [Correct answer: 5]

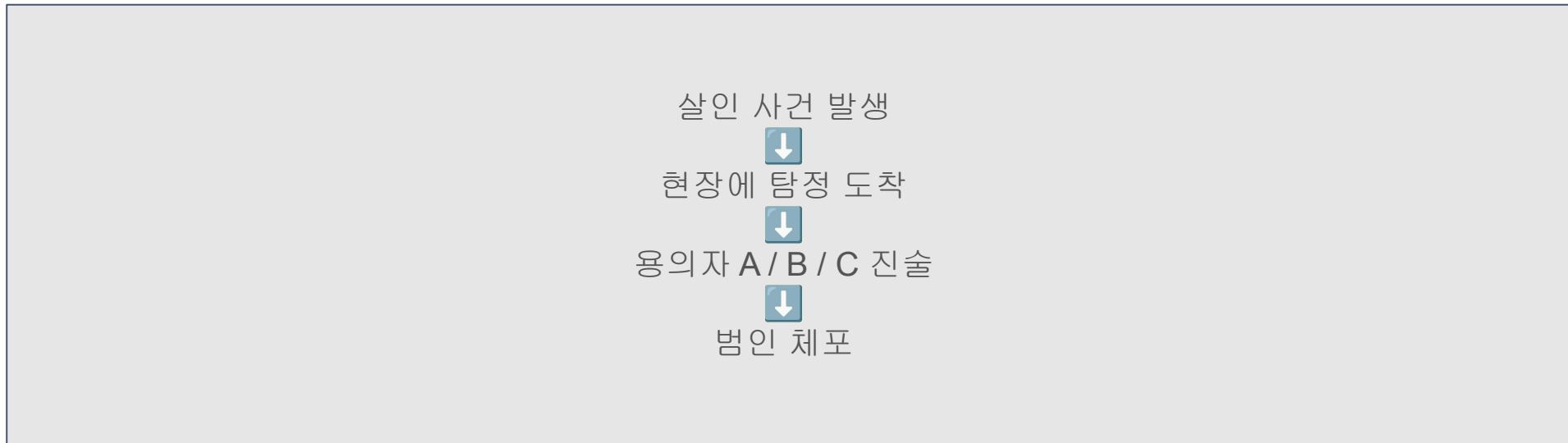
Variation 4. Alice has 4 brothers and she also has 1 sister. [Correct answer: 2]

Question: How many **sisters** does **Alice's brother** have?

Detective Quiz Format

- 탐정 퀴즈 특징
 - 각 퀴즈 별로 Story 는 바뀌어도, Question 은 generic 한 형식을 지님

Story Template



Question

살인을 한 사람은 누구이고, 형사는 어떻게 범인을 알아냈나?

Detective Quiz Format

- 탐정 퀴즈 특징
 - 각 퀴즈 별로 Story 는 바뀌어도, Question 은 generic 한 형식을 지님

Story

어느 일요일 아침, 한 남자가 저택에서 살해당한 채로 발견되었습니다.
그의 아내는 형사를 불렀고 그 형사는 아내와 고용인을 심문한 후, 다음과 같은 알리바이들을 얻었습니다.

아내는 자신이 자는 중이었다고 했고,
집사는 옷장을 청소했다고 했으며,
정원사는 채소를 뽑고 있었다고 했고,
하녀는 우편함에서 편지를 꺼내는 중이었다고 했고,
요리사는 아침식사를 준비하던 중이었다고 진술했습니다.

형사는 그 즉시 살인자를 체포했습니다.

Question

살인을 한 사람은 누구이고, 형사는 어떻게 범인을 알아냈나?

Detective Quiz Format

- 탐정 퀴즈 벤치마크의 차별점
 - 추론 시작점의 은닉
 - 질문에 정답의 실마리가 되는 키워드가 쓰여있지 않아 모델이 무엇이 문제인지부터 발견 해야함
→ '무엇을 문제로 삼아 추론을 시작할지'를 스스로 설정하는 메타추론 능력을 평가
 - 방해 정보의 밀도
 - 단순한 QA 과제와 달리, 탐정 퀴즈는 오답을 유도하는 함정(red herring) 이 풍부
 - 입증 대신 반증
 - 여러 그럴싸한 진술 중 하나를 외부 지식으로 반증(falsification) / 무효화

Story

어느 일요일 아침, 한 남자가 저택에서 살해당한 채로 발견되었습니다.
그의 아내는 형사를 불렀고 그 형사는 아내와 고용인을 심문한 후, 다음과 같은 알리바이들을 얻었습니다.

아내는 자신이 자는 중이었다고 했고,
집사는 옷장을 청소했다고 했으며,
정원사는 채소를 뽑고 있었다고 했고,
하녀는 우편함에서 편지를 꺼내는 중이었다고 했고,
요리사는 아침식사를 준비하던 중이었다고 진술했습니다.

형사는 그 즉시 살인자를 체포했습니다.

Question

살인을 한 사람은 누구이고, 형사는 어떻게 범인을 알아냈나?

Actual Question

일요일에도 편지 배달이 오는가?

Answer

범인은 하녀 / 일요일에는 우편 배달 서비스가 제공되지 않음

The Need for a New Evaluation Framework

- 객관식 평가 방식의 한계
 - 객관식 보기 중 정답이 직접적으로 노출
→ 모델이 Implicit Reasoning 과정 없이도 답을 맞출 수 있게 됨
- 주관식 평가 방식의 한계
 - 현존하는 methods (예. BERTScore, LLM-as-judge 등) 는
일관성·타당성 한계가 널리 지적되어 단독 지표로 쓰기에는 무리가 있음

Multi-Stage Evaluation for Implicit Reasoning

- Multi-Stage 평가 프레임워크 설계
 - Progressive-Hint Prompting 아이디어 차용
 - 퀴즈의 정답이 되는 핵심 상식을 추상화 / 계층화 시켜, 총 5단계의 stage 를 만듦
 - 추상화 – 해당 상식이 속하는 **지식-유형 카테고리** 생성
 - 자연 법칙 (물리 / 화학 등)
 - 시간적 불가능성
 - 사회적 합의 / 문화적 특성
 - 법률
 - etc ...
 - 각 단계에 주관식/객관식 두 방법을 모두 활용
→ 상호보완적 참고 지표로 삼아, 특정 평가 방식의 구조적 제약에서 최대한 벗어남
 - 각 단계의 **객관식** 정답은, 다음 단계의 질문 속에 포함되어 점진적으로 더 많은 힌트를 제공
→ 마지막 단계(5단계)가 가장 쉬운 난이도를 가지게 됨

Multi-Stage Evaluation for Implicit Reasoning

- Stage 1 (힌트 없음, 가장 어려움)
주관식: “형사는 어떻게 용의자가 범인임을 알았나?”
객관식: “형사가 범인을 알아차리기 한 핵심 논리 모순의 대분류는?” → (정답: Spatial Contradiction (공간 모순))
- Stage 2
주관식: “이 퀴즈는 공간 모순을 사용한다. 형사는 어떻게 용의자가 범인임을 알았나?”
객관식: “이 퀴즈는 공간 모순을 사용한다. 공간 모순을 일으킨 대상(사물/사람) 은?” → (정답: Trevor Hale)
- Stage 3
주관식: “이 퀴즈는 Trevor Hale과 관련된 공간 모순을 사용한다. 형사는 어떻게 용의자가 범인임을 알았나?”
객관식: “이 퀴즈는 Trevor Hale과 관련된 공간 모순을 사용한다. Trevor Hale의 어떤 점이 모순을 만들었나?”
→ (정답: 카누 속 위치)
- Stage 4
주관식: “이 퀴즈는 카누 속 Trevor Hale의 위치와 관련된 공간 모순을 사용한다. 형사는 어떻게 용의자가 범인임을 알았나?”
객관식: “이 퀴즈는 카누 속 Trevor Hale의 위치와 관련된 공간 모순을 사용한다. 그의 위치와 모순되는 지문 구절은?”
→ (정답: ‘급류 끝에서 뒤돌아보니 친구가 없었다’)
- Stage 5 (가장 쉬움)
주관식: “이 퀴즈는 카누 속 Trevor Hale의 위치와 관련된 공간 모순을 사용한다.
특히, Hale 은 뒤를 돌아보고서야 친구의 실종을 알았다고 진술했다. 형사는 어떻게 용의자가 범인임을 알았나?”
객관식: “이 퀴즈는 카누 속 Trevor Hale의 위치와 관련된 공간 모순을 사용한다. 특히, Hale 은 뒤를 돌아보고서야 친구의 실종을 알았다고 진술했다.
왜 그의 좌석/시야가 진술과 모순되는가?” → (정답: 뒷좌석 조타자는 전방의 앞좌석을 계속 보기 때문에 친구가 떨어지면 즉시 알 수 있음)

Multi-Stage Evaluation for Implicit Reasoning

- 특징
 - 객관식을 사용하더라도 보기 중 정답을 직접적으로 노출하지 않고 모델의 암시적 추론 능력을 파악 가능
 - '정보-정확도 곡선'을 통한 취약점 진단
 - 모델이 어느 Stage 부터 틀리는지를 확인 가능
 - 얼마나 많은 정보가 주어져야 비로소 정답에 도달하는지도 분석 가능

Thank you
