How Far Are We From Automating AI Research

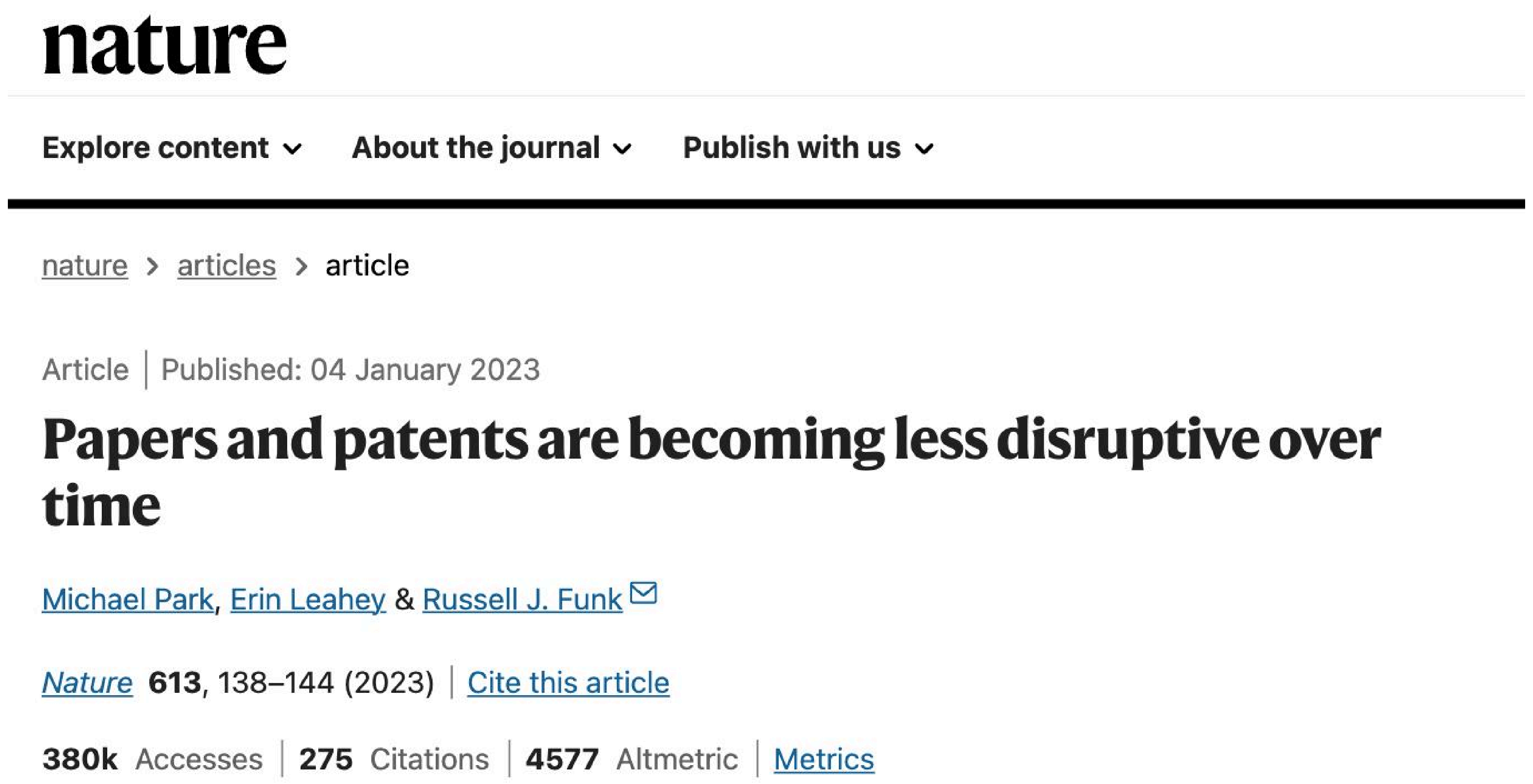
Stanford NLP

Chenglei Si
clsi@stanford.edu

Big Picture

Scientific research is important,

Scientific research is important, but difficult to scale.



nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > articles > article

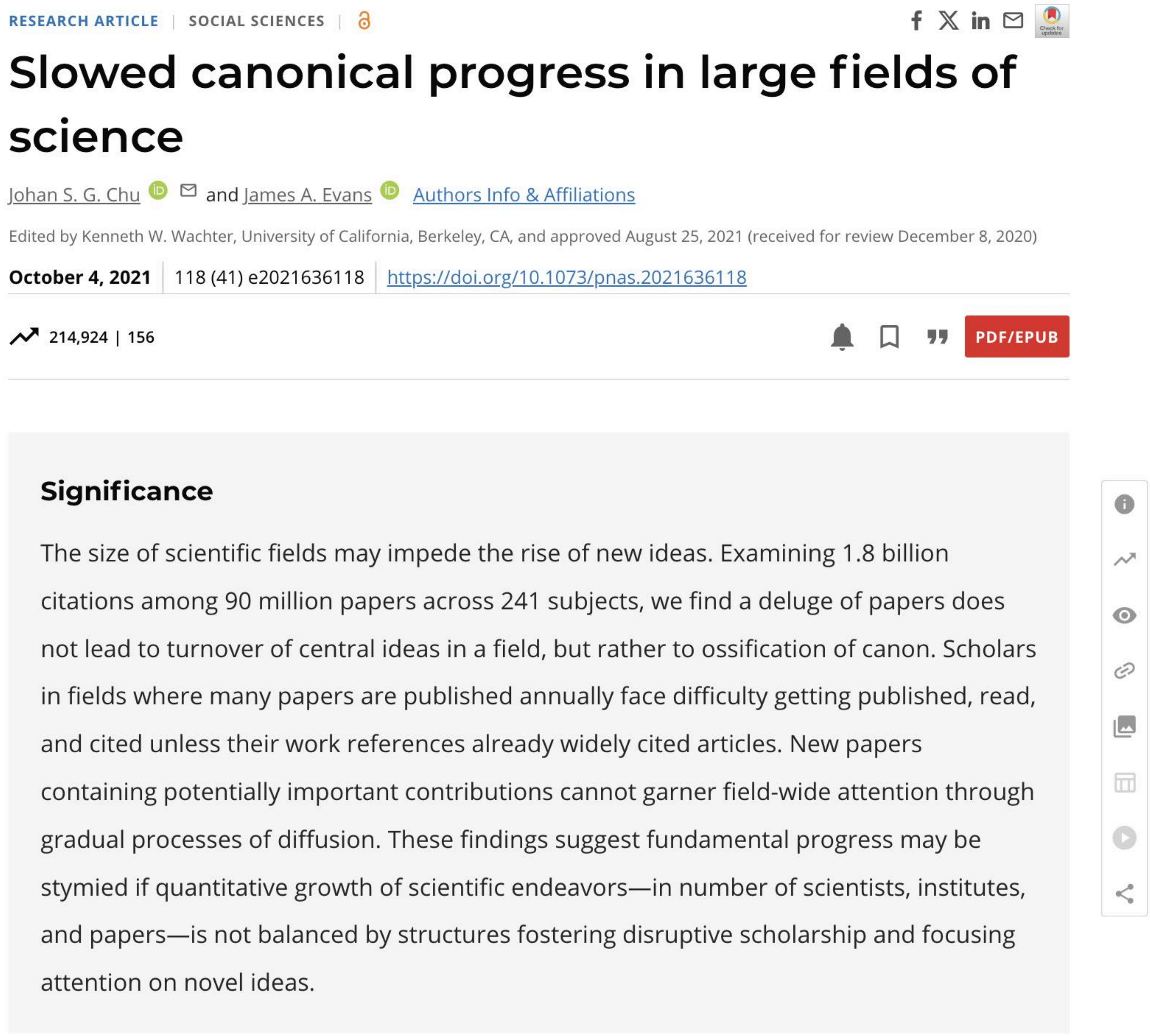
Article | Published: 04 January 2023

Papers and patents are becoming less disruptive over time

Michael Park, Erin Leahey & Russell J. Funk

Nature **613**, 138–144 (2023) | Cite this article

380k Accesses | **275** Citations | **4577** Altmetric | Metrics



RESEARCH ARTICLE | SOCIAL SCIENCES

Slowed canonical progress in large fields of science

Johan S. G. Chu and James A. Evans Authors Info & Affiliations

Edited by Kenneth W. Wachter, University of California, Berkeley, CA, and approved August 25, 2021 (received for review December 8, 2020)

October 4, 2021 | 118 (41) e2021636118 | https://doi.org/10.1073/pnas.2021636118

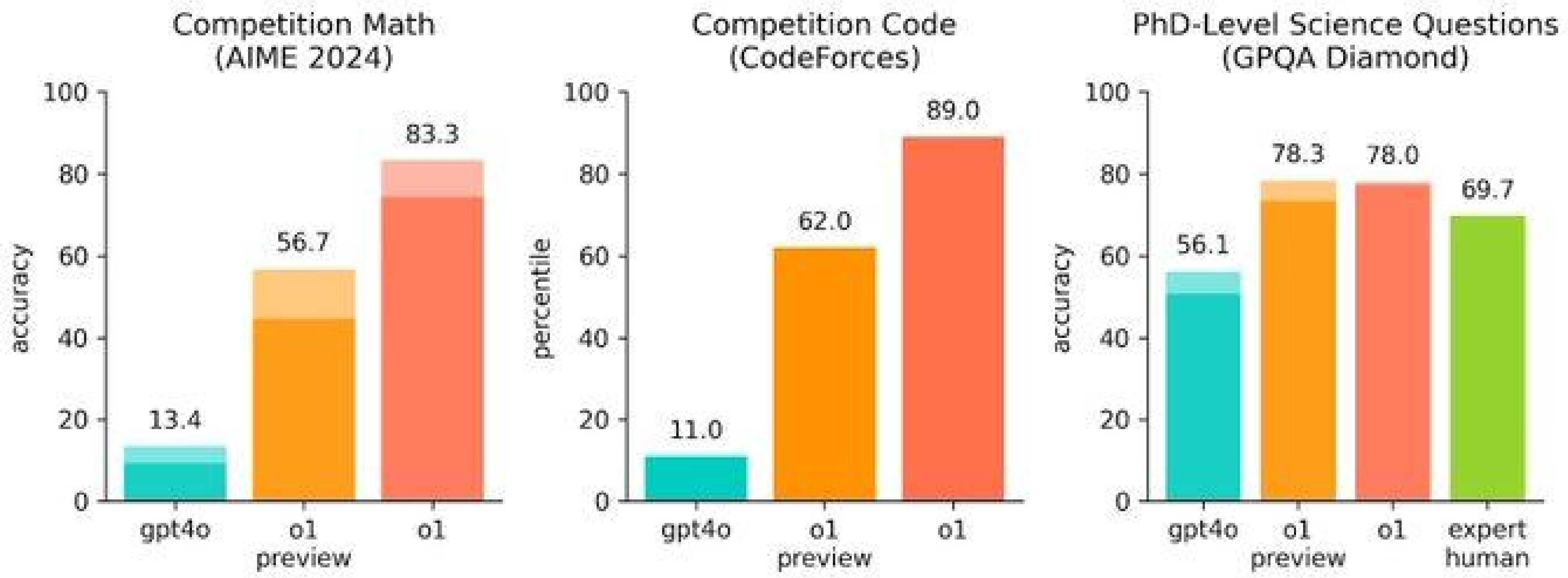
214,924 | 156

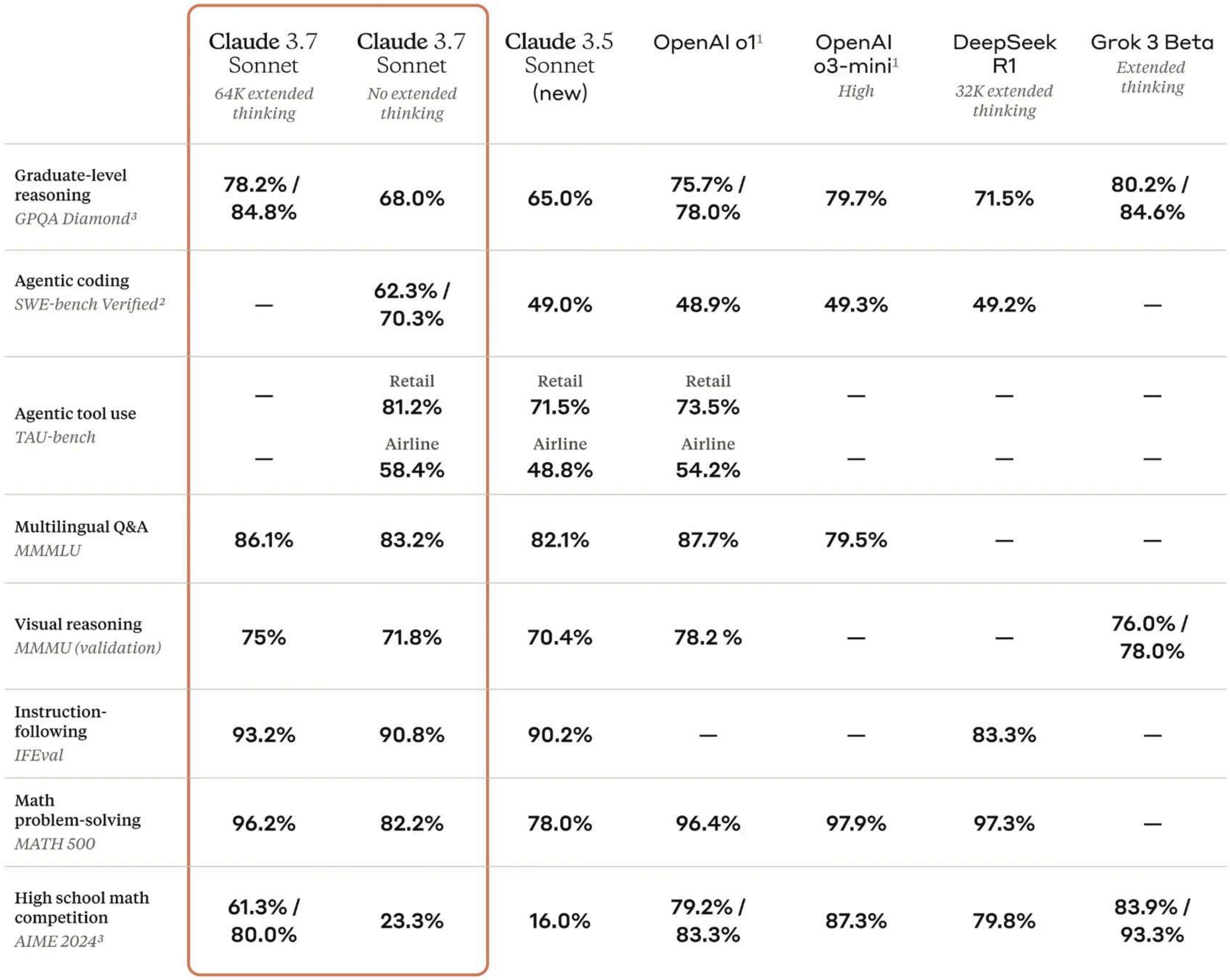
PDF/EPUB

Significance

The size of scientific fields may impede the rise of new ideas. Examining 1.8 billion citations among 90 million papers across 241 subjects, we find a deluge of papers does not lead to turnover of central ideas in a field, but rather to ossification of canon. Scholars in fields where many papers are published annually face difficulty getting published, read, and cited unless their work references already widely cited articles. New papers containing potentially important contributions cannot garner field-wide attention through gradual processes of diffusion. These findings suggest fundamental progress may be stymied if quantitative growth of scientific endeavors—in number of scientists, institutes, and papers—is not balanced by structures fostering disruptive scholarship and focusing attention on novel ideas.

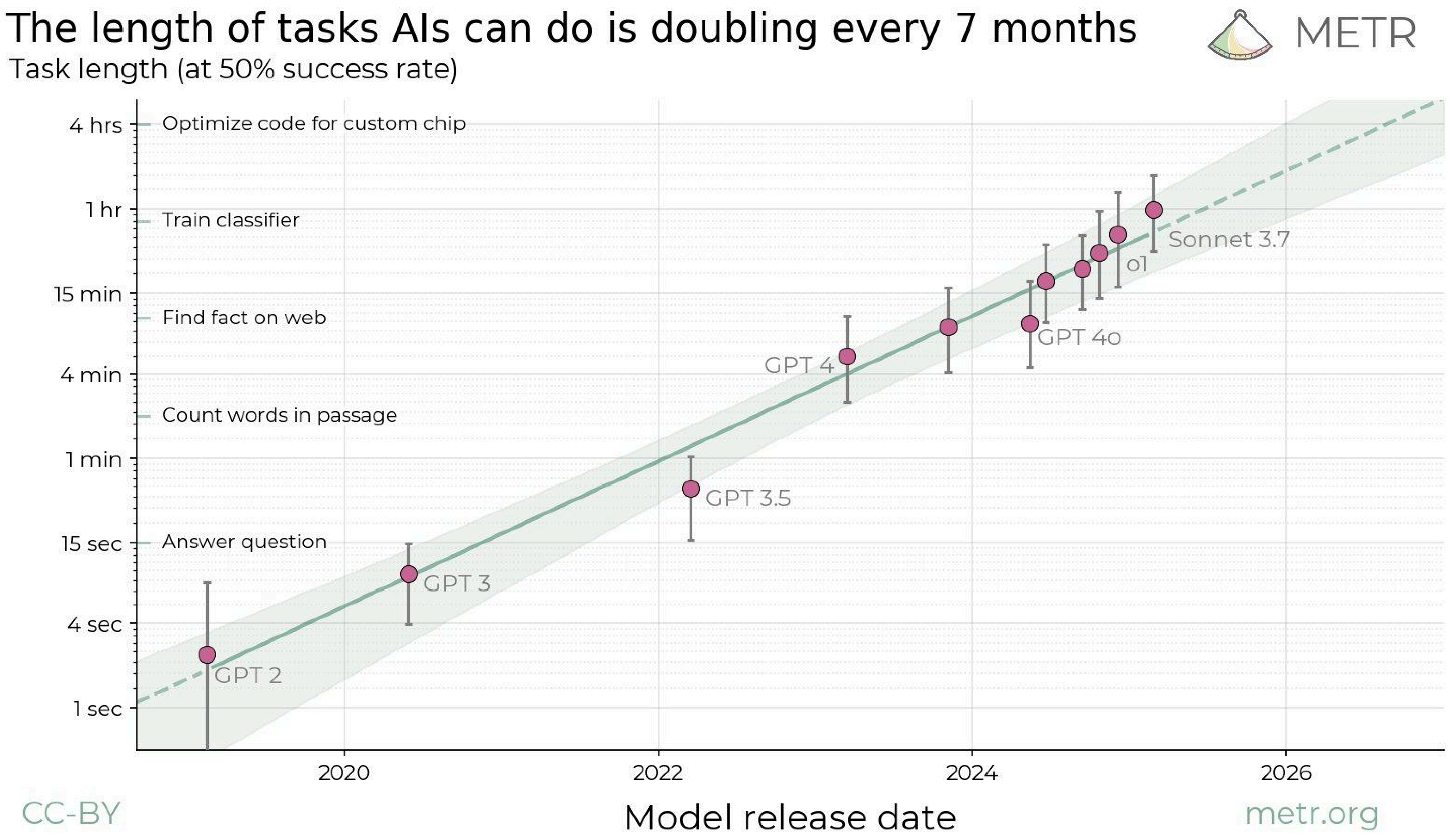
Meanwhile, LLMs are getting better.





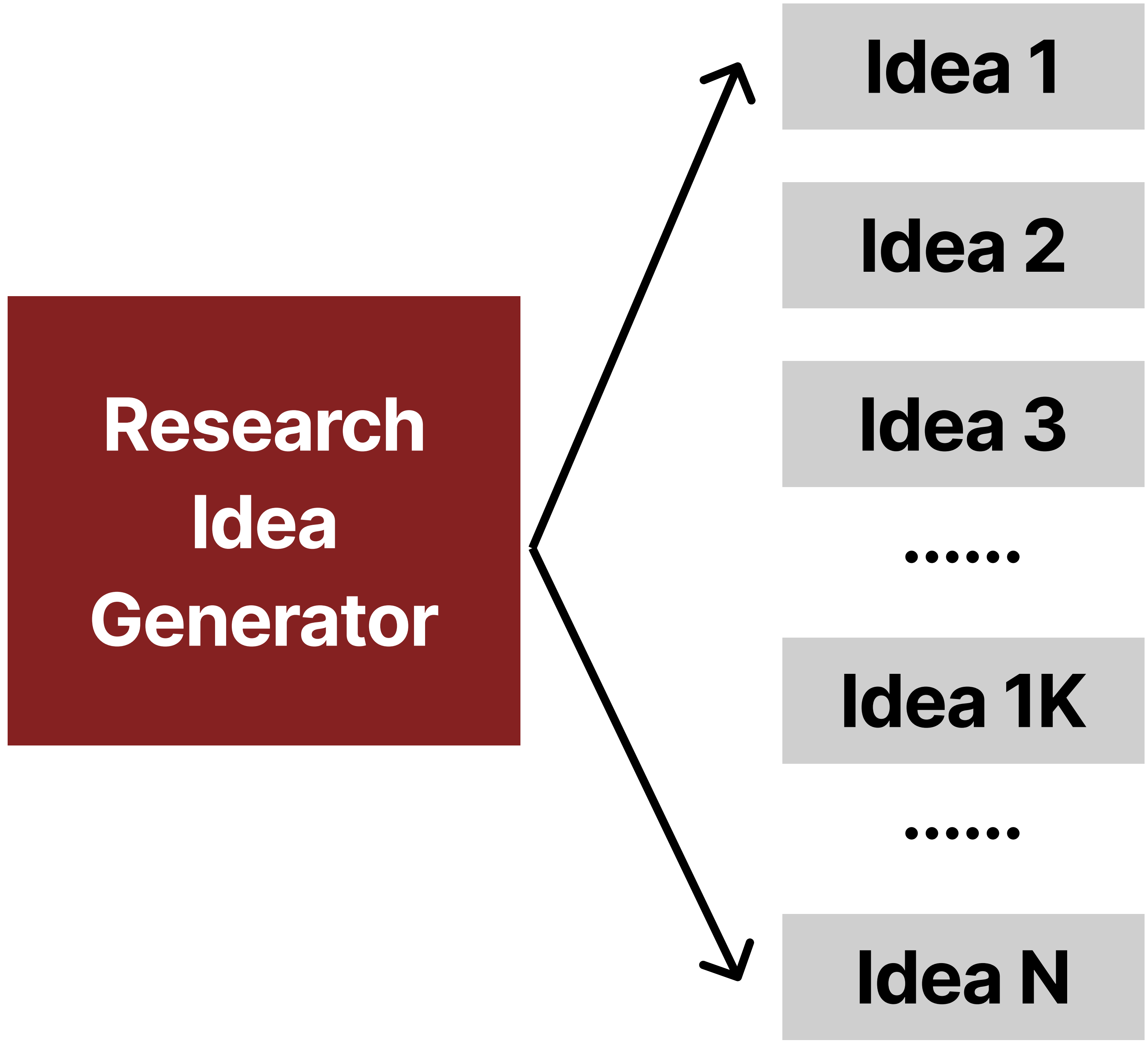
	Claude 3.7 Sonnet *64K extended thinking*	Claude 3.7 Sonnet *No extended thinking*	Claude 3.5 Sonnet (new)	OpenAI o1[1]	OpenAI o3-mini[1] *High*	DeepSeek R1 *32K extended thinking*	Grok 3 Beta *Extended thinking*
Graduate-level reasoning *GPQA Diamond[3]*	78.2% / 84.8%	68.0%	65.0%	75.7% / 78.0%	79.7%	71.5%	80.2% / 84.6%
Agentic coding *SWE-bench Verified[2]*	—	62.3% / 70.3%	49.0%	48.9%	49.3%	49.2%	—
Agentic tool use *TAU-bench*	— —	Retail 81.2% Airline 58.4%	Retail 71.5% Airline 48.8%	Retail 73.5% Airline 54.2%	— —	— —	— —
Multilingual Q&A *MMMLU*	86.1%	83.2%	82.1%	87.7%	79.5%	—	—
Visual reasoning *MMMU (validation)*	75%	71.8%	70.4%	78.2 %	—	—	76.0% / 78.0%
Instruction-following *IFEval*	93.2%	90.8%	90.2%	—	—	83.3%	—
Math problem-solving *MATH 500*	96.2%	82.2%	78.0%	96.4%	97.9%	97.3%	—
High school math competition *AIME 2024[3]*	61.3% / 80.0%	23.3%	16.0%	79.2% / 83.3%	87.3%	79.8%	83.9% / 93.3%

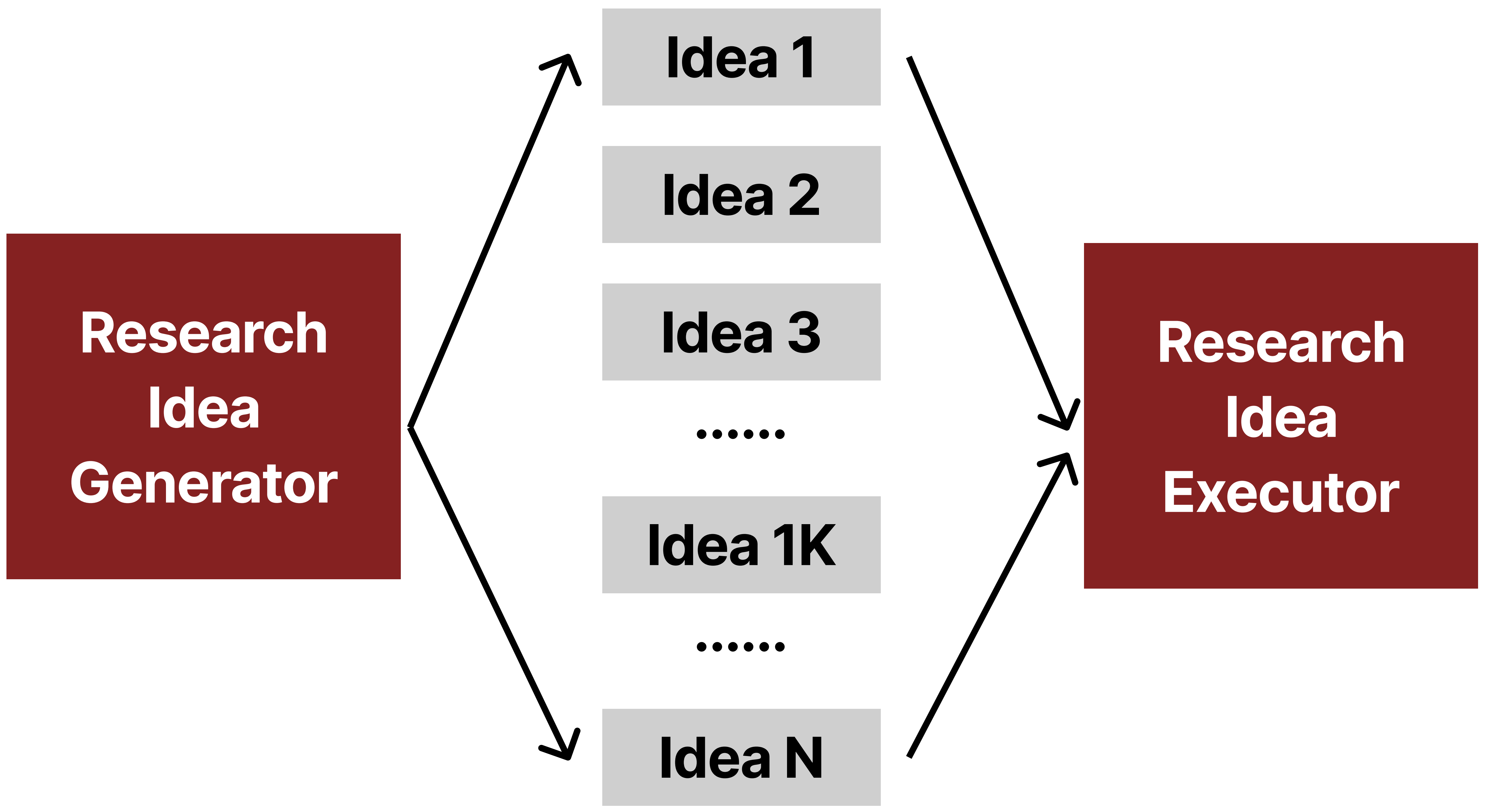
Meanwhile, LLMs are getting better.



So, can LLMs help with scientific research?

Research
Idea
Generator


Research Idea Generator
Idea 1
Idea 2
Idea 3
......
Idea 1K
......
Idea N



Research Idea Generator

Idea 1

Idea 2

Idea 3

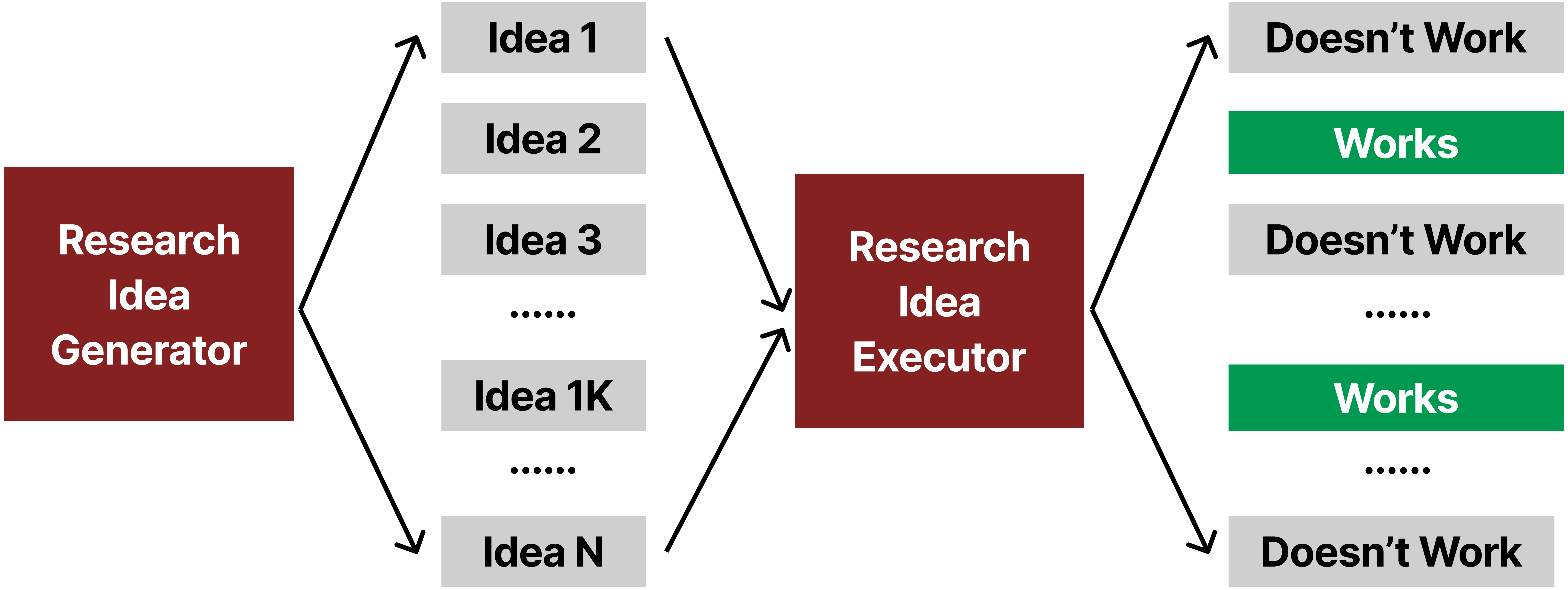
……

Idea 1K

……

Idea N

Research Idea Executor



Research Idea Generator

Idea 1

Idea 2

Idea 3

……

Idea 1K

……

Idea N

Research Idea Executor

Doesn’t Work

Works

Doesn’t Work

……

Works

……

Doesn’t Work

For this paradigm to succeed, we need:

1. LLMs to generate “good” research ideas (Part 1 & 2)

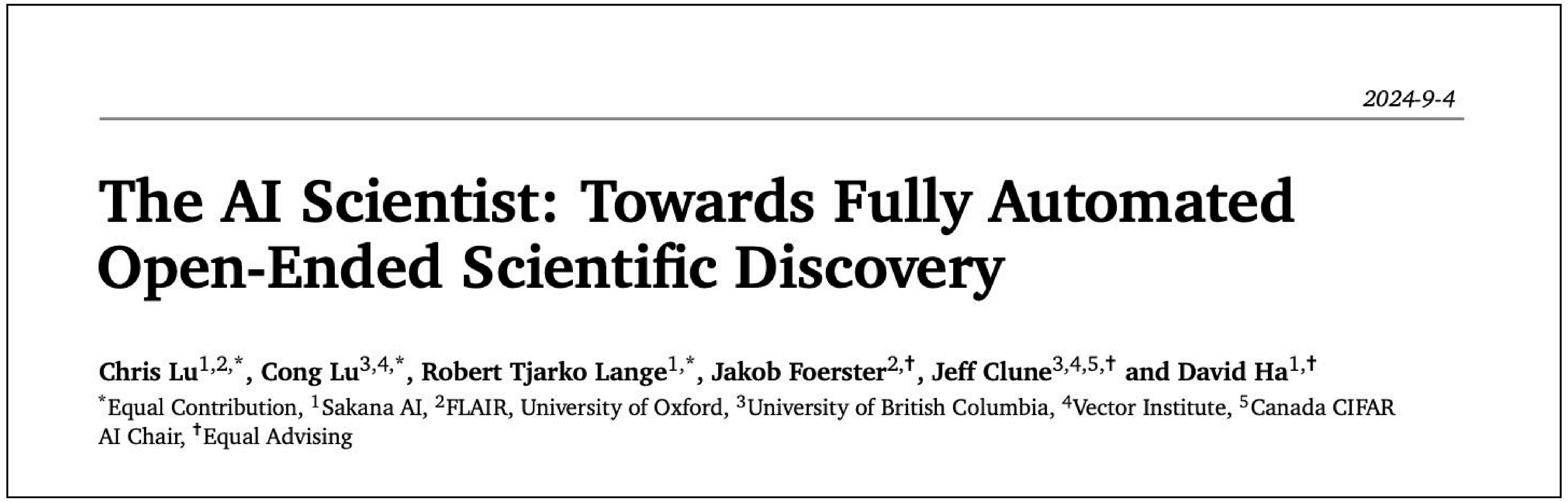
2. LLMs to execute ideas “correctly” (Part 3)

For this paradigm to succeed, we need:

1. LLMs to generate “good” research ideas (Part 1 & 2)

2. LLMs to execute ideas “correctly” (Part 3)

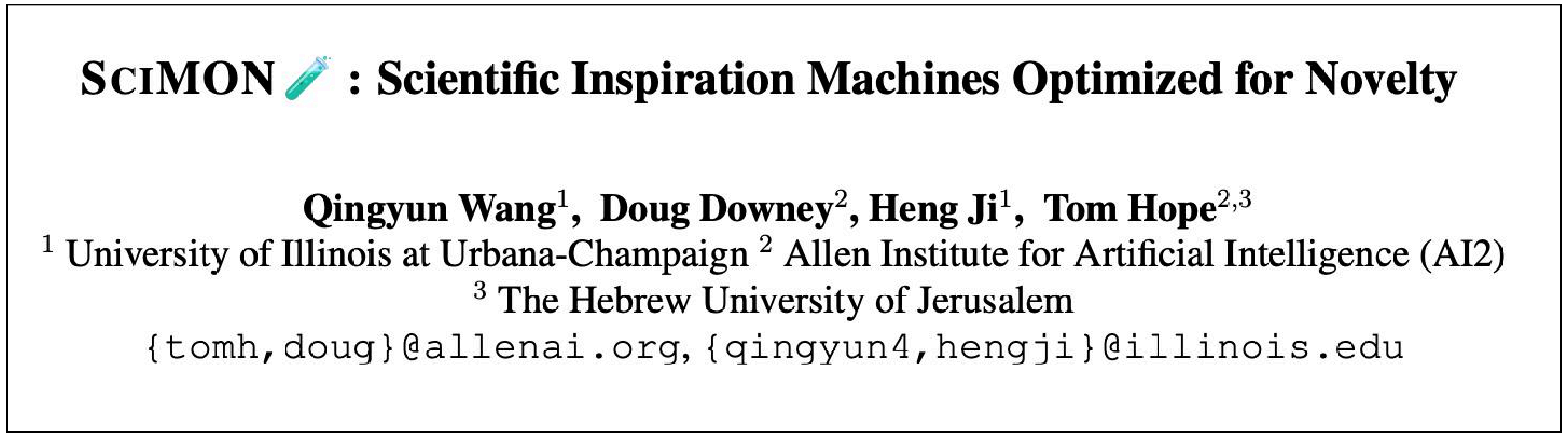
We are not the only one thinking about this.



2024-9-4

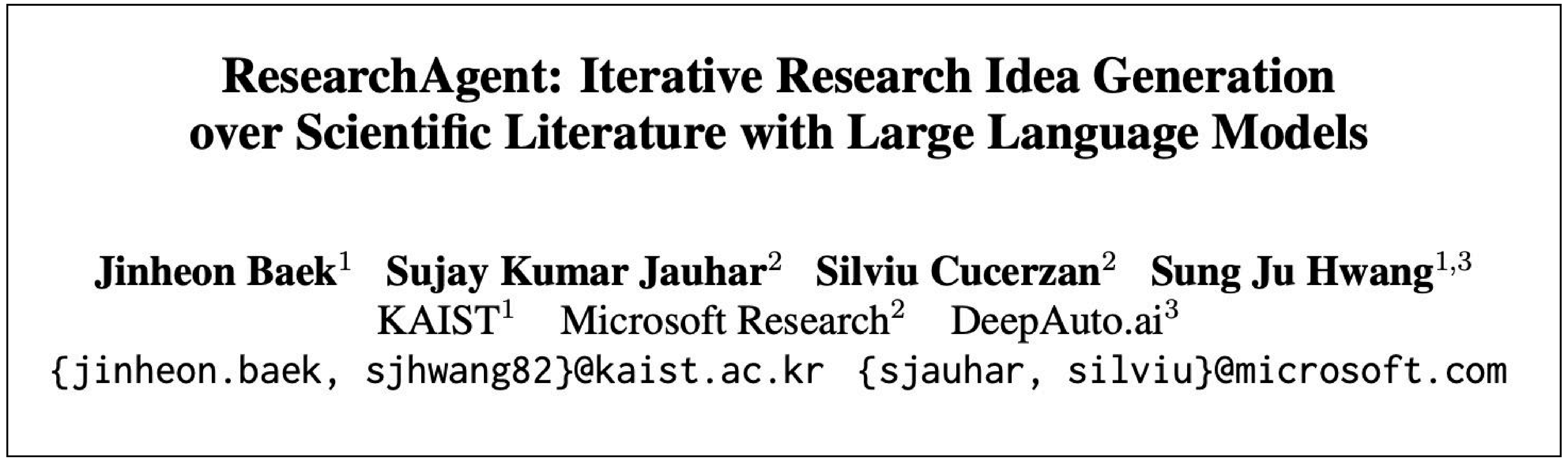
The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery

Chris Lu[1,2,*], Cong Lu[3,4,*], Robert Tjarko Lange[1,*], Jakob Foerster[2,†], Jeff Clune[3,4,5,†] and David Ha[1,†]
[*]Equal Contribution, [1]Sakana AI, [2]FLAIR, University of Oxford, [3]University of British Columbia, [4]Vector Institute, [5]Canada CIFAR AI Chair, [†]Equal Advising



SCIMON 🧪 : Scientific Inspiration Machines Optimized for Novelty

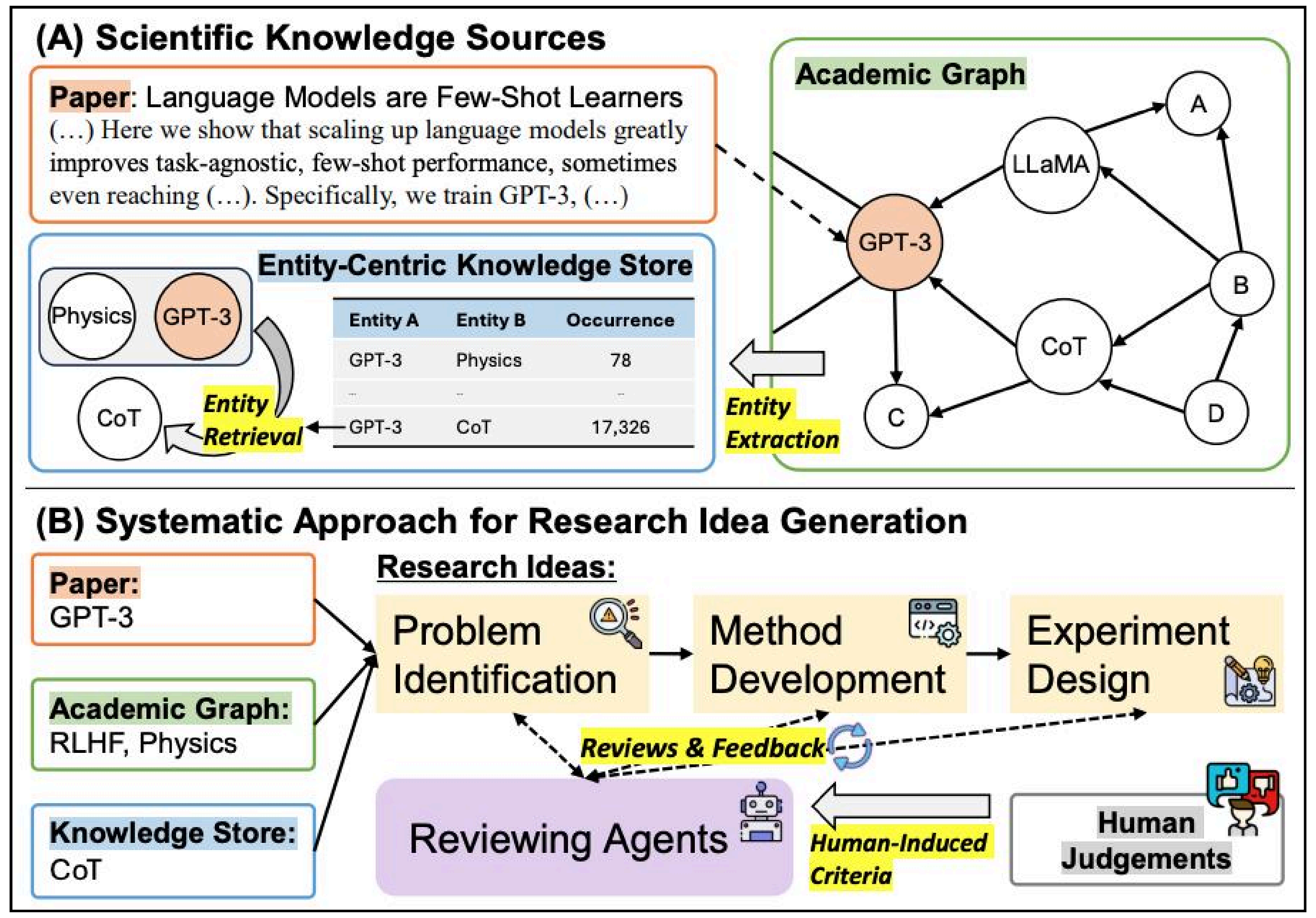
Qingyun Wang[1], Doug Downey[2], Heng Ji[1], Tom Hope[2,3]
[1] University of Illinois at Urbana-Champaign [2] Allen Institute for Artificial Intelligence (AI2)
[3] The Hebrew University of Jerusalem
{tomh,doug}@allenai.org, {qingyun4,hengji}@illinois.edu

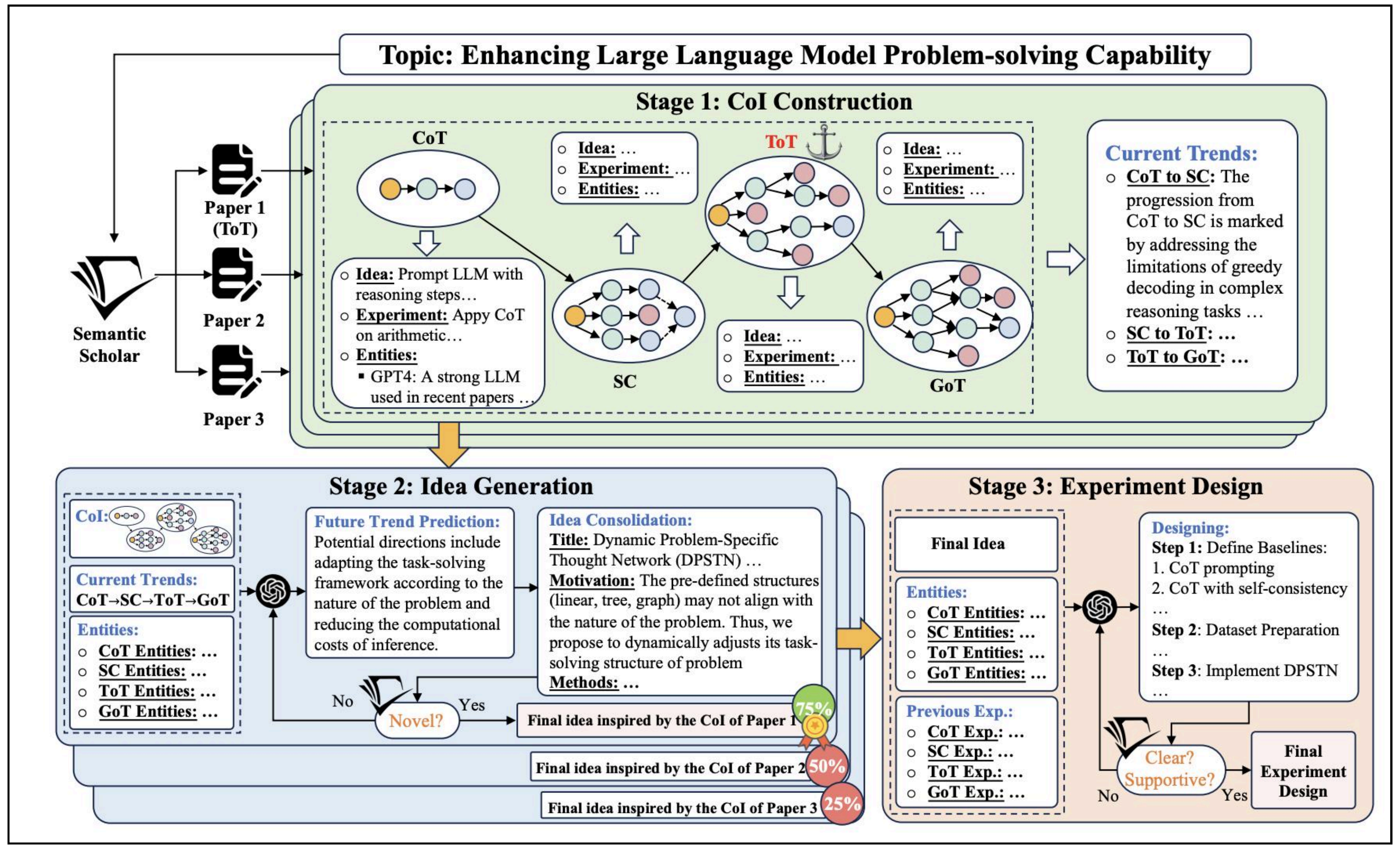


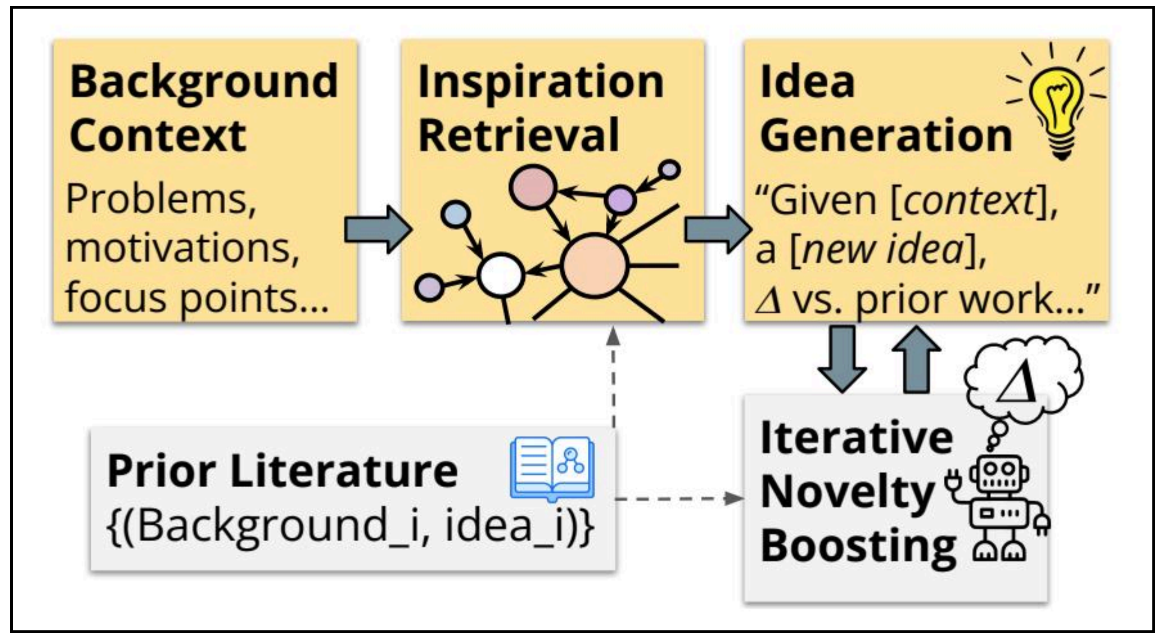
ResearchAgent: Iterative Research Idea Generation over Scientific Literature with Large Language Models

Jinheon Baek[1] Sujay Kumar Jauhar[2] Silviu Cucerzan[2] Sung Ju Hwang[1,3]
KAIST[1] Microsoft Research[2] DeepAuto.ai[3]
{jinheon.baek, sjhwang82}@kaist.ac.kr {sjauhar, silviu}@microsoft.com

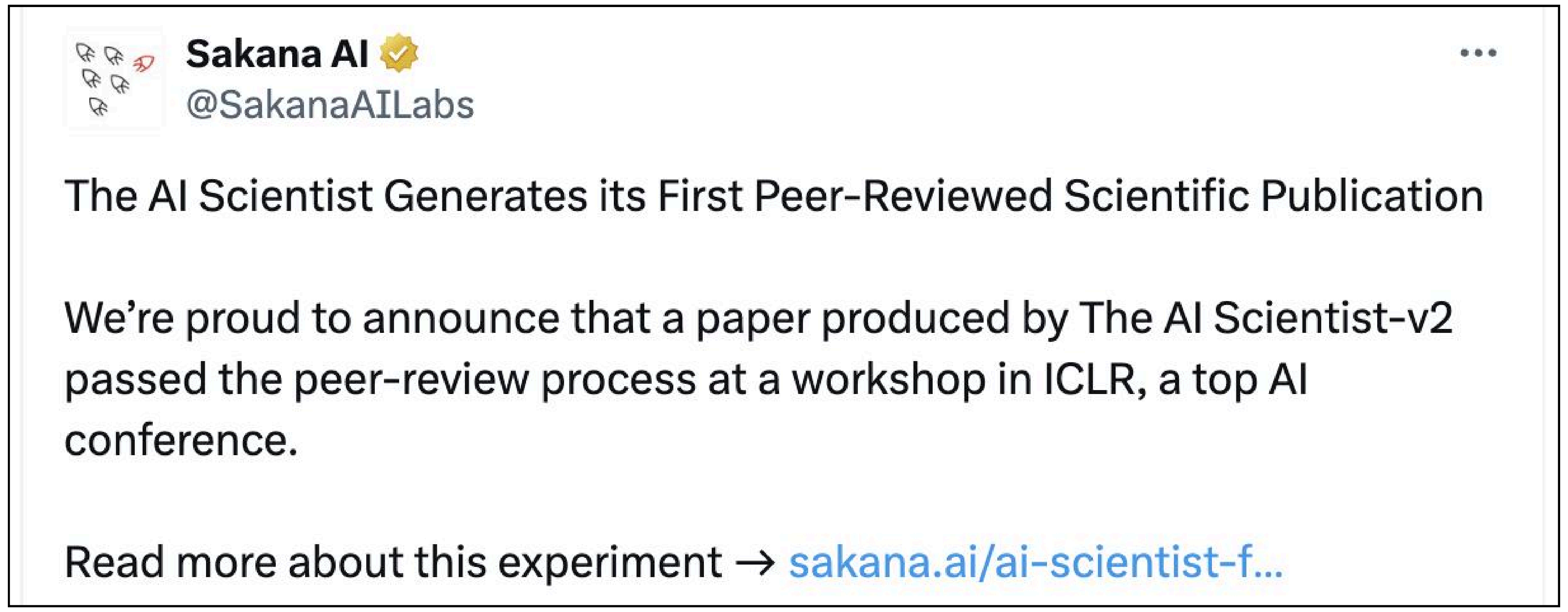
A lot of systems have been built,


(A) Scientific Knowledge Sources
Paper: Language Models are Few-Shot Learners
(…) Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching (…). Specifically, we train GPT-3, (…)
Entity-Centric Knowledge Store
Physics
GPT-3
CoT
Entity Retrieval
Entity A | Entity B | Occurrence
GPT-3 | Physics | 78
GPT-3 | CoT | 17,326
Academic Graph
A
LLaMA
GPT-3
B
CoT
C
D
Entity Extraction
(B) Systematic Approach for Research Idea Generation
Paper: GPT-3
Academic Graph: RLHF, Physics
Knowledge Store: CoT
Research Ideas:
Problem Identification
Method Development
Experiment Design
Reviews & Feedback
Reviewing Agents
Human-Induced Criteria
Human Judgements


Topic: Enhancing Large Language Model Problem-solving Capability
Semantic Scholar
Paper 1 (ToT)
Paper 2
Paper 3
Stage 1: CoI Construction
CoT
Idea: Prompt LLM with reasoning steps…
Experiment: Appy CoT on arithmetic…
Entities:
GPT4: A strong LLM used in recent papers …
SC
Idea: …
Experiment: …
Entities: …
ToT
GoT
Current Trends:
CoT to SC: The progression from CoT to SC is marked by addressing the limitations of greedy decoding in complex reasoning tasks …
SC to ToT: …
ToT to GoT: …
Stage 2: Idea Generation
CoI:
Current Trends: CoT→SC→ToT→GoT
Entities:
CoT Entities: …
SC Entities: …
ToT Entities: …
GoT Entities: …
Future Trend Prediction: Potential directions include adapting the task-solving framework according to the nature of the problem and reducing the computational costs of inference.
Idea Consolidation:
Title: Dynamic Problem-Specific Thought Network (DPSTN) …
Motivation: The pre-defined structures (linear, tree, graph) may not align with the nature of the problem. Thus, we propose to dynamically adjusts its task-solving structure of problem
Methods: …
No
Novel?
Yes
Final idea inspired by the CoI of Paper 1
75%
Final idea inspired by the CoI of Paper 2
50%
Final idea inspired by the CoI of Paper 3
25%
Stage 3: Experiment Design
Final Idea
Entities:
CoT Entities: …
SC Entities: …
ToT Entities: …
GoT Entities: …
Previous Exp.:
CoT Exp.: …
SC Exp.: …
ToT Exp.: …
GoT Exp.: …
Designing:
Step 1: Define Baselines:
1. CoT prompting
2. CoT with self-consistency
…
Step 2: Dataset Preparation
…
Step 3: Implement DPSTN
…
Clear? Supportive?
No
Yes
Final Experiment Design


Background Context
Problems, motivations, focus points…
Inspiration Retrieval
Idea Generation
"Given [context], a [new idea], Δ vs. prior work…"
Prior Literature
{(Background_i, idea_i)}
Iterative Novelty Boosting
Δ

A lot of systems have been built,

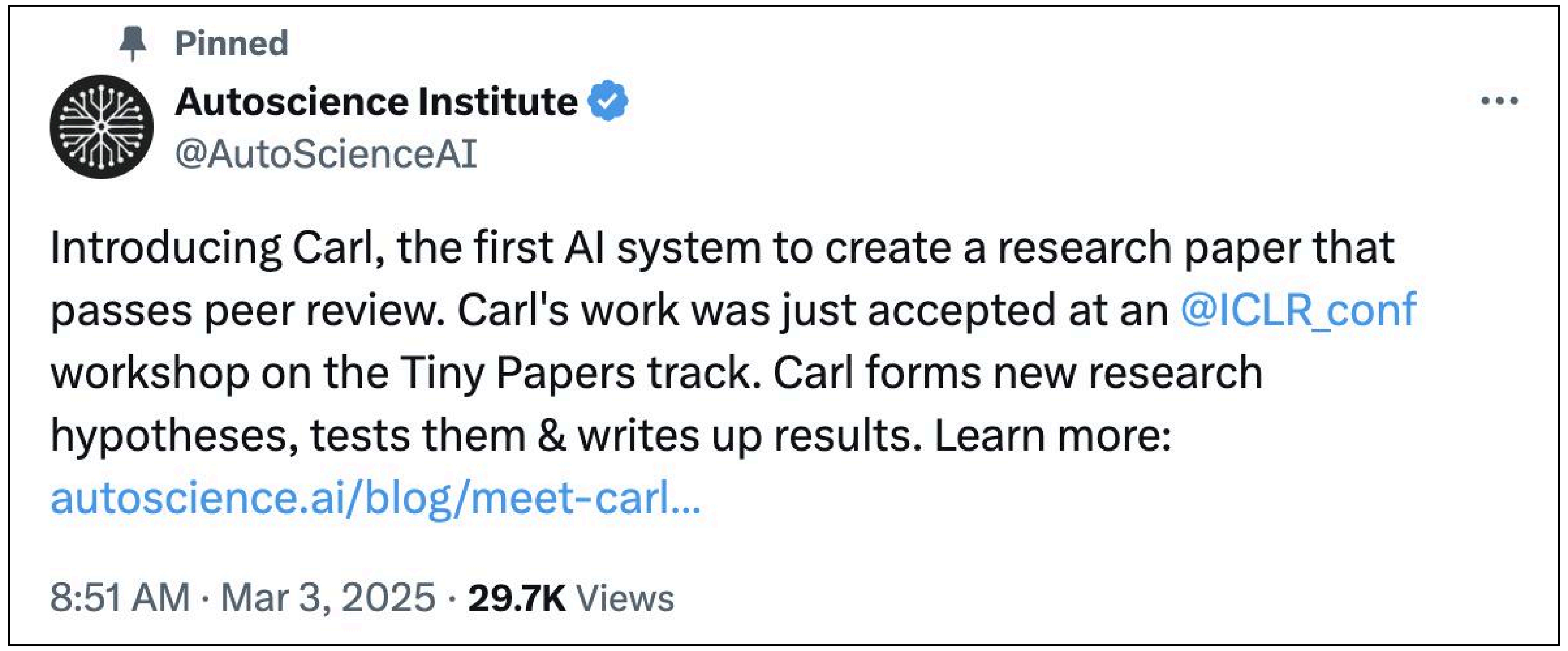


Sakana AI
@SakanaAILabs

The AI Scientist Generates its First Peer-Reviewed Scientific Publication

We're proud to announce that a paper produced by The AI Scientist-v2 passed the peer-review process at a workshop in ICLR, a top AI conference.

Read more about this experiment → sakana.ai/ai-scientist-f...

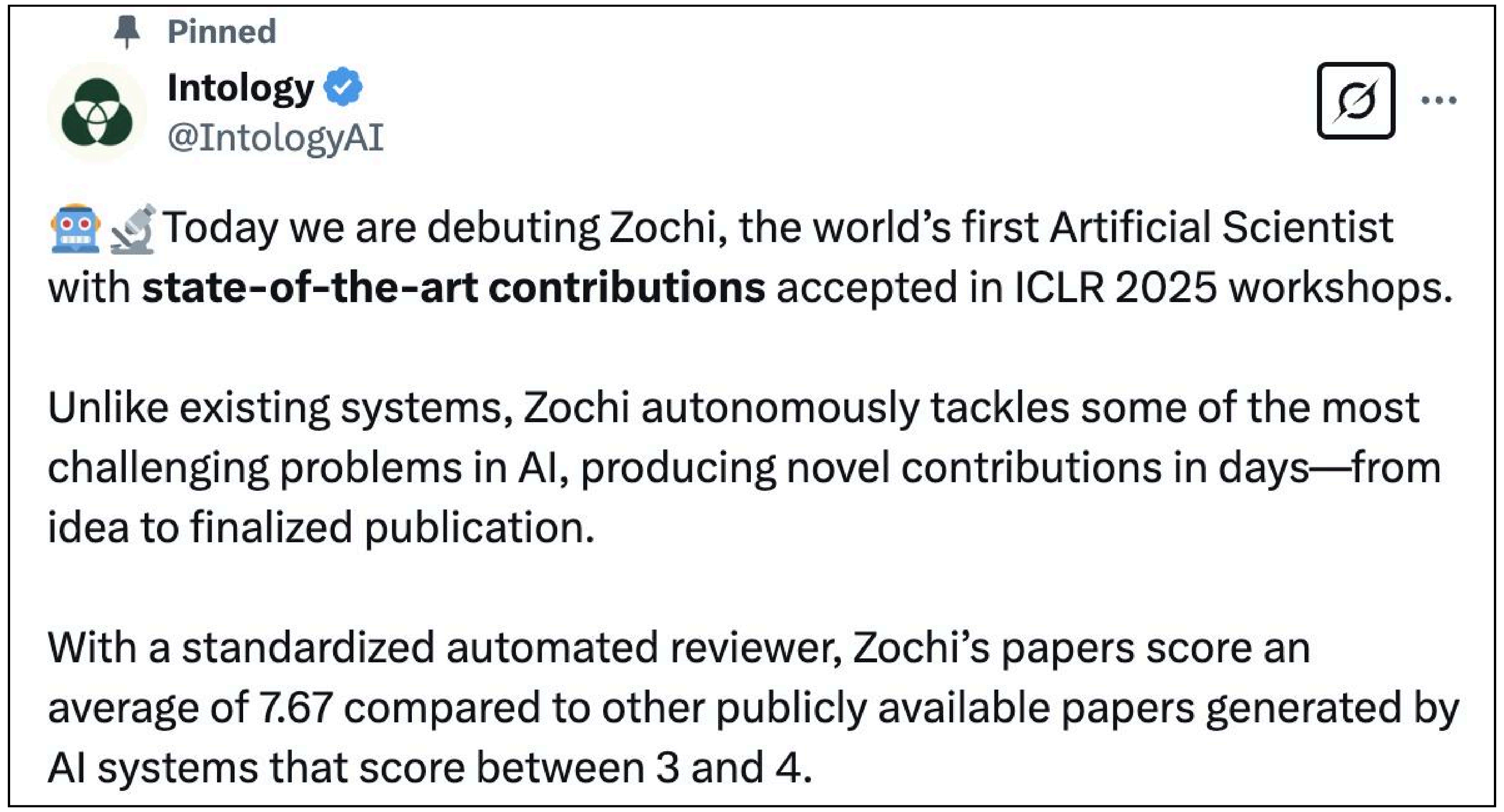


Pinned

Autoscience Institute
@AutoScienceAI

Introducing Carl, the first AI system to create a research paper that passes peer review. Carl's work was just accepted at an @ICLR_conf workshop on the Tiny Papers track. Carl forms new research hypotheses, tests them & writes up results. Learn more: autoscience.ai/blog/meet-carl...

8:51 AM · Mar 3, 2025 · **29.7K** Views



Pinned

Intology
@IntologyAI

🤖🔬Today we are debuting Zochi, the world's first Artificial Scientist with **state-of-the-art contributions** accepted in ICLR 2025 workshops.

Unlike existing systems, Zochi autonomously tackles some of the most challenging problems in AI, producing novel contributions in days—from idea to finalized publication.

With a standardized automated reviewer, Zochi's papers score an average of 7.67 compared to other publicly available papers generated by AI systems that score between 3 and 4.

A lot of systems have been built, but we don’t know how well they work.

Past Works

- No human baseline
- Small-scale human eval or LLM-as-a-Judge

Before moving on, we need some good evaluation to know where we are.

Before moving on, we need some good evaluation to know where we are.

- Can LLMs Generate Novel Research Ideas? (Part 1)
- Can LLM Ideas be Executed as Successful Projects? (Part 2)

Past Works

- No human baseline
- Small-scale human eval or LLM-as-a-Judge

Our Approach (Part 1)

- Compare to expert researchers as the baseline
- Large-scale expert review

Past Works

- No human baseline
- Small-scale human eval or LLM-as-a-Judge

Our Approach (Part 1)

- Compare to expert researchers as the baseline
- Large-scale expert review

Our Approach (Part 2)

- Recruit experts to execute all ideas into full projects
- Large-scale expert review on the full projects

Part 1

Can LLMs Generate Novel Research Ideas?

A Large-Scale Human Study with 100+ NLP Researchers

Chenglei Si, Diyi Yang, Tatsunori Hashimoto
Stanford University
{clsi, diyiy, thashim}@stanford.edu

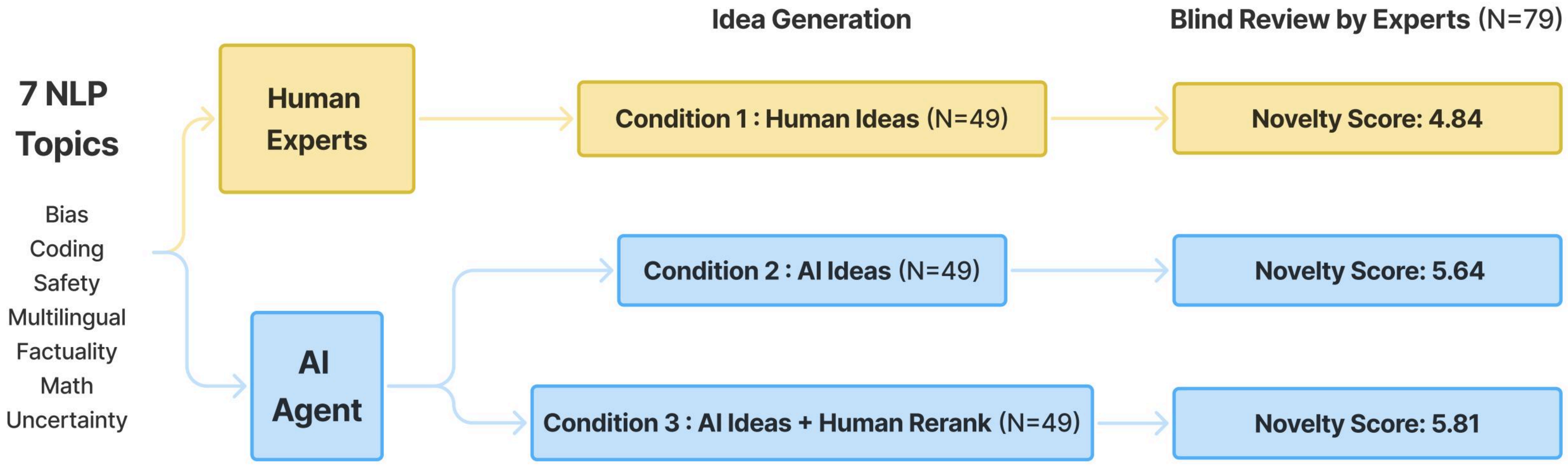
Outline

1. Study Design
2. Idea Generation Agent
3. Human Experts
4. Results
5. Analysis of Ideas & Reviews
6. Limitations of LLMs

Outline

1. Study Design
2. Idea Generation Agent
3. Human Experts
4. Results
5. Analysis of Ideas & Reviews
6. Limitations of LLMs

Study Design: Overview



Idea Generation

Blind Review by Experts (N=79)

7 NLP Topics

Bias
Coding
Safety
Multilingual
Factuality
Math
Uncertainty

Human Experts

AI Agent

Condition 1 : Human Ideas (N=49)

Condition 2 : AI Ideas (N=49)

Condition 3 : AI Ideas + Human Rerank (N=49)

Novelty Score: 4.84

Novelty Score: 5.64

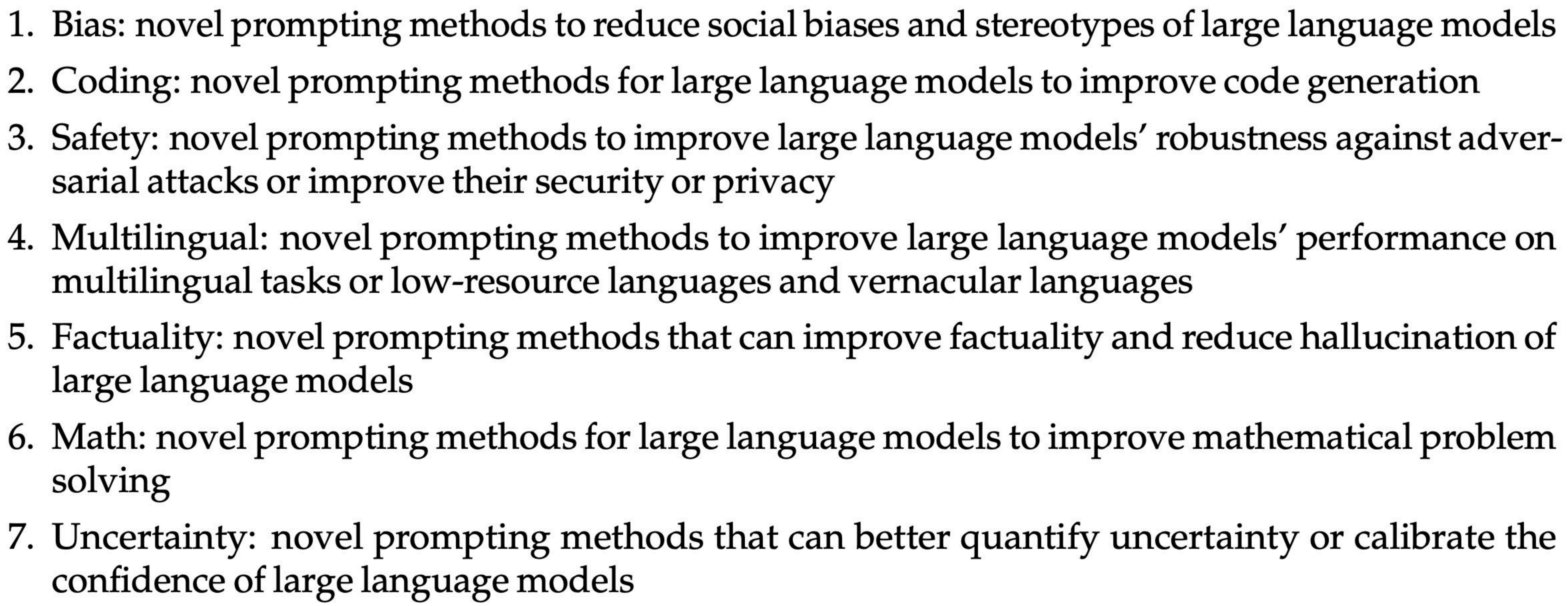
Novelty Score: 5.81

Study Design: Ideation Scope

Why only prompting-based research?

- Active area of research
- Execution tends to be quick and requires minimal computing hardware

Topics:



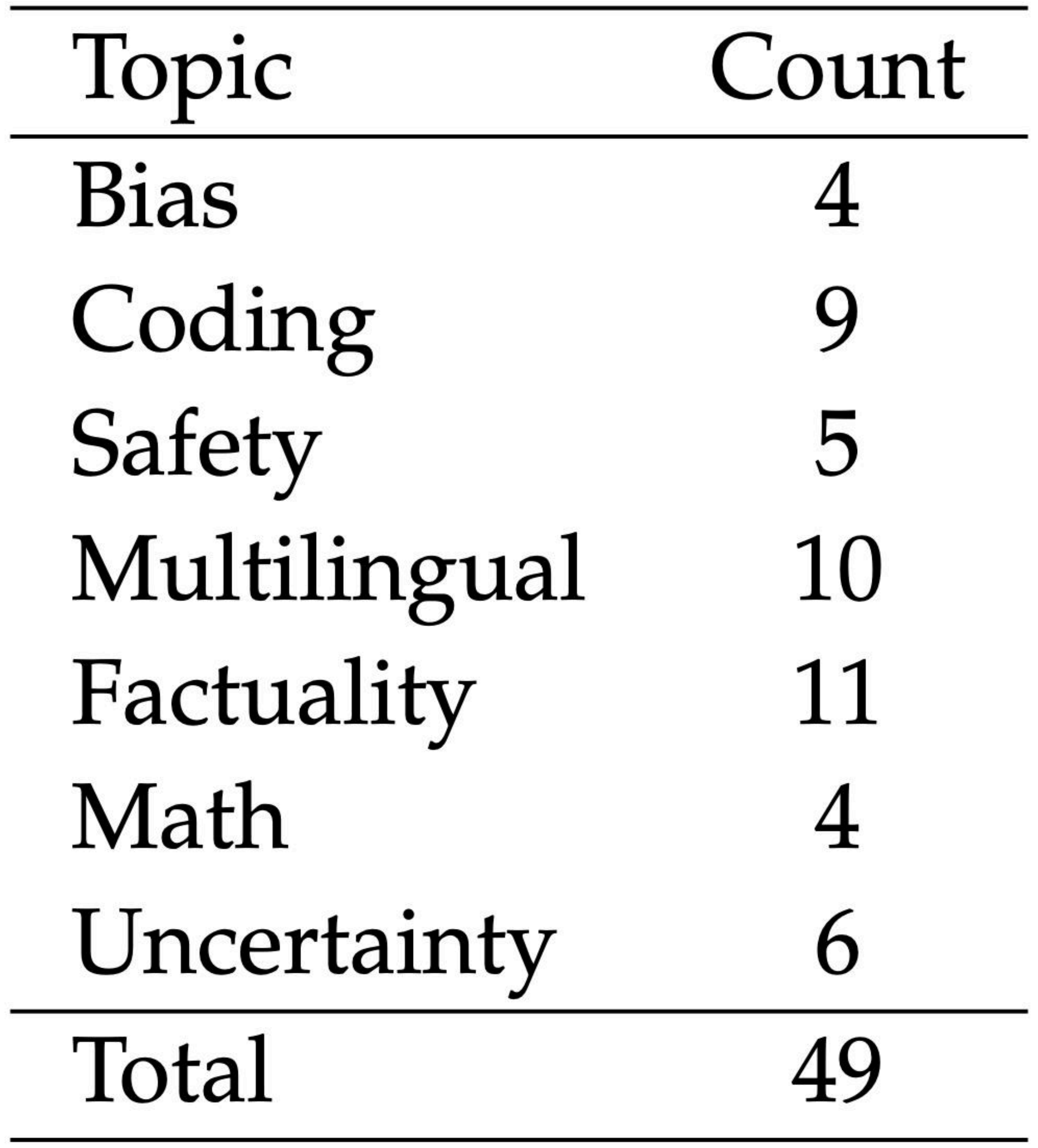
1. Bias: novel prompting methods to reduce social biases and stereotypes of large language models
2. Coding: novel prompting methods for large language models to improve code generation
3. Safety: novel prompting methods to improve large language models' robustness against adversarial attacks or improve their security or privacy
4. Multilingual: novel prompting methods to improve large language models' performance on multilingual tasks or low-resource languages and vernacular languages
5. Factuality: novel prompting methods that can improve factuality and reduce hallucination of large language models
6. Math: novel prompting methods for large language models to improve mathematical problem solving
7. Uncertainty: novel prompting methods that can better quantify uncertainty or calibrate the confidence of large language models

Study Design: Ideation Scope

Why only prompting-based research?

- Active area of research
- Execution tends to be quick and requires minimal computing hardware

Topic distribution:



Topic	Count
Bias	4
Coding	9
Safety	5
Multilingual	10
Factuality	11
Math	4
Uncertainty	6
Total	49

Study Design: Idea Writeup

Same format for both humans and LLM:

- Title
- Problem Statement
- Motivation
- Proposed Method
- Step-by-Step Experiment Plan
- Test Case Examples
- Fallback Plan

Study Design: Style Standardization

For all ideas:

- Use an LLM to standardize writing styles without changing contents.
- Expert judges get 50% accuracy on distinguishing AI vs human ideas.

Study Design: Review & Evaluation

NeurIPS

- **Originality**: Are the tasks or methods new?
- **Quality**: Is the submission technically sound?
- **Clarity**: Is the submission clearly written?
- **Significance**: Are the results important?

- **Overall 10**: "Technically flawless paper with groundbreaking impact on one or more areas of AI, with exceptionally strong evaluation, reproducibility, and resources, and no unaddressed ethical considerations."

https://neurips.cc/Conferences/2024/ReviewerGuidelines

Study Design:
Review & Evaluation

Review form:

- Novelty
- Excitement
- Feasibility
- Expected Effectiveness
- Overall
- For all metrics: 1-10 scale + rationale

Study Design: Experiment Conditions

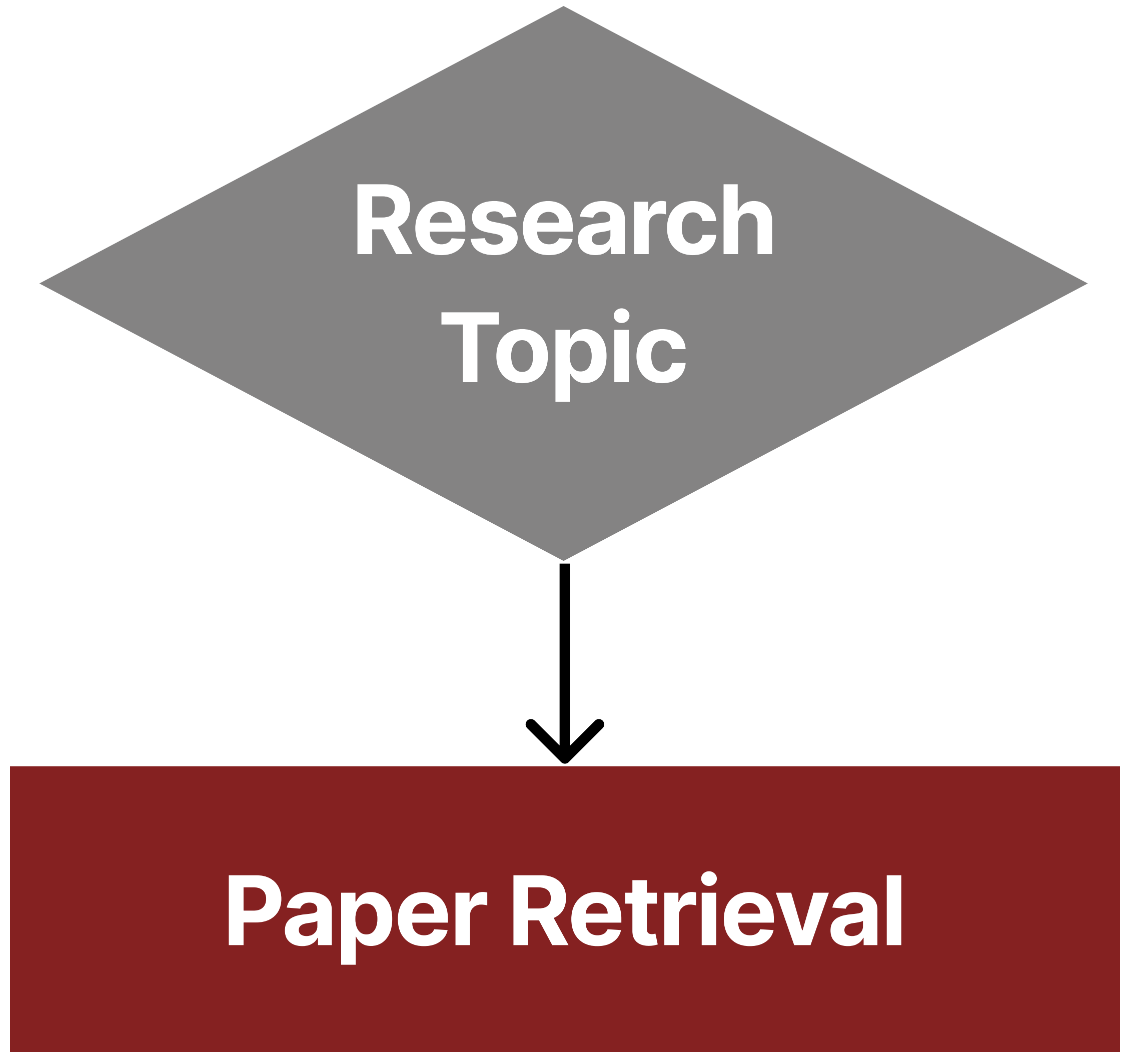
- Human Ideas
- AI Ideas
- AI Ideas + Human Rerank

Outline

1. Study Design
2. **Idea Generation Agent**
3. Human Experts
4. Results
5. Analysis of Ideas & Reviews
6. Limitations of LLMs

Idea Generation Agent: Design Principle

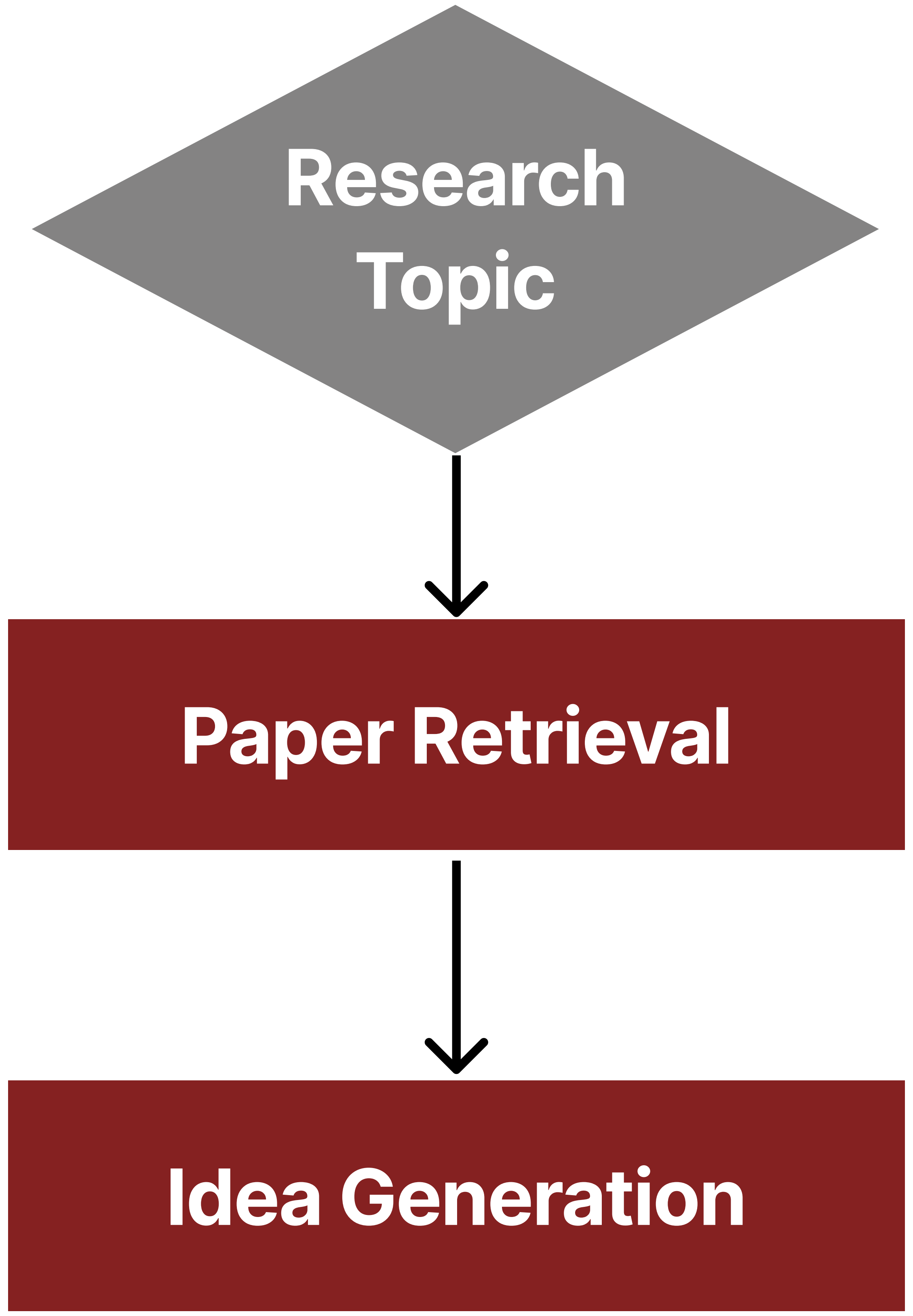
- **Simple but effective**
- **RAG**
- **Inference Scaling: Over-generate & Rerank**



- Generate function calls of Semantic Scholar API
- LLM Reranking

Idea Generation Agent
Step 1: Paper Retrieval

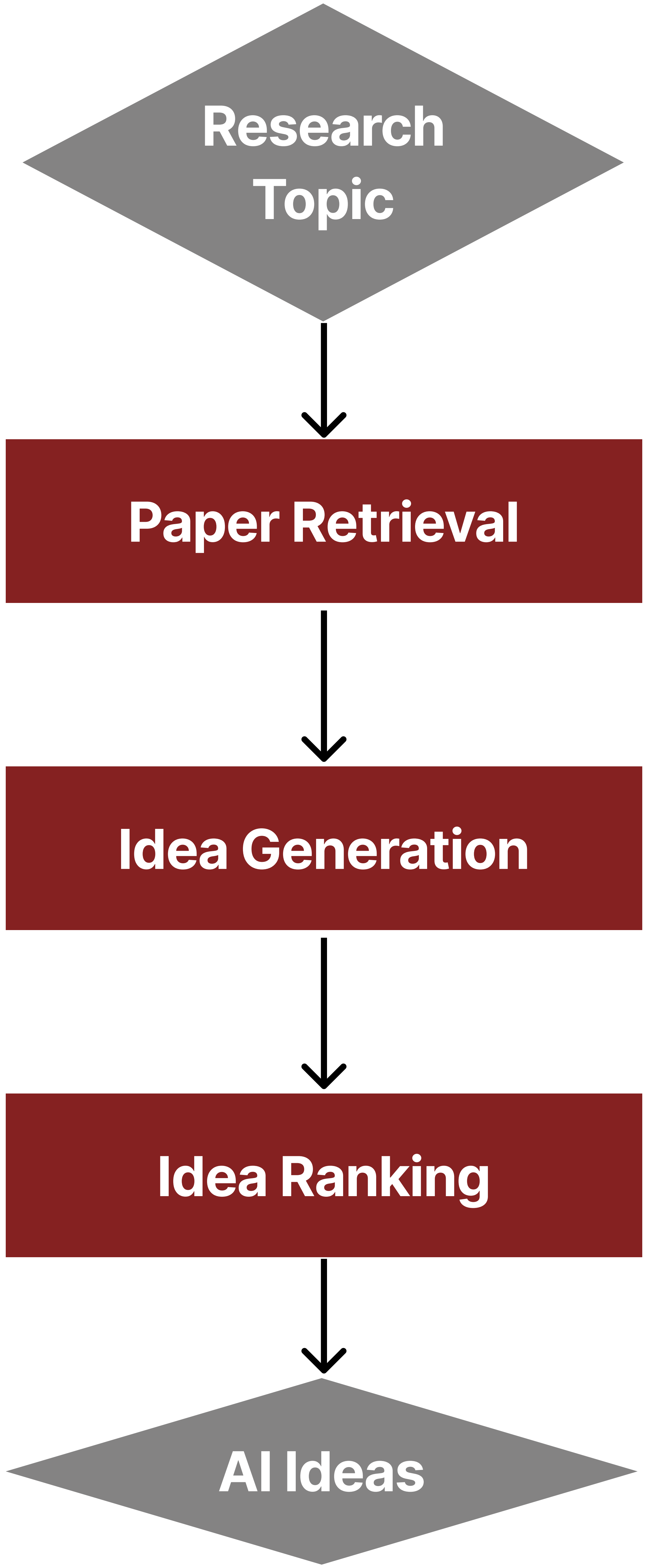
Idea Generation Agent
Step 2: Idea Generation



- Generate function calls of Semantic Scholar API
- LLM Reranking

- RAG
- Generate in batches
- Append previous batches to reduce repetition

Idea Generation Agent
Step 3: Idea Ranking



Research Topic

Paper Retrieval

- Generate function calls of Semantic Scholar API
- LLM Reranking

Idea Generation

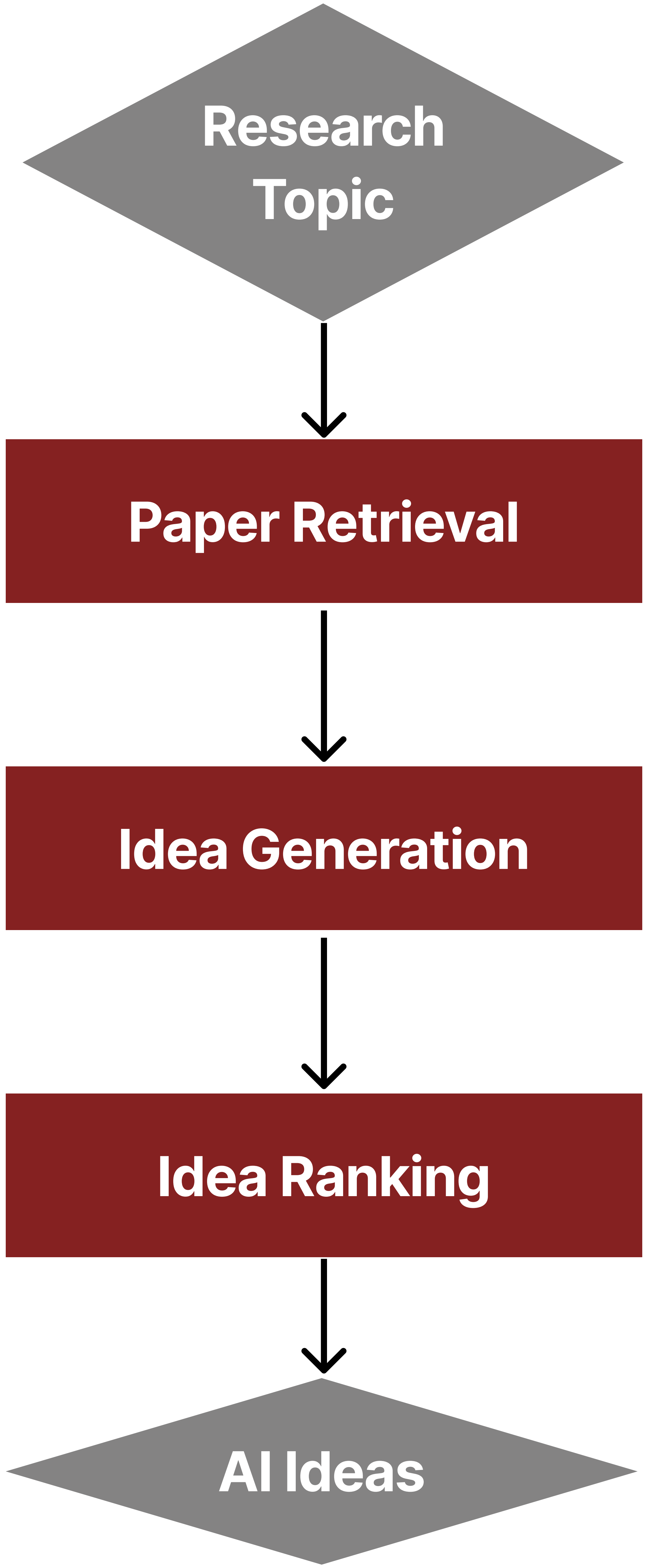
- RAG
- Generate in batches
- Append previous batches to reduce repetition

Idea Ranking

- Pairwise comparison for N rounds
- ICLR data as proxy benchmark

AI Ideas

Idea Generation Agent
Step 3: Idea Ranking



- Generate function calls of Semantic Scholar API
- LLM Reranking

- RAG
- Generate in batches
- Append previous batches to reduce repetition

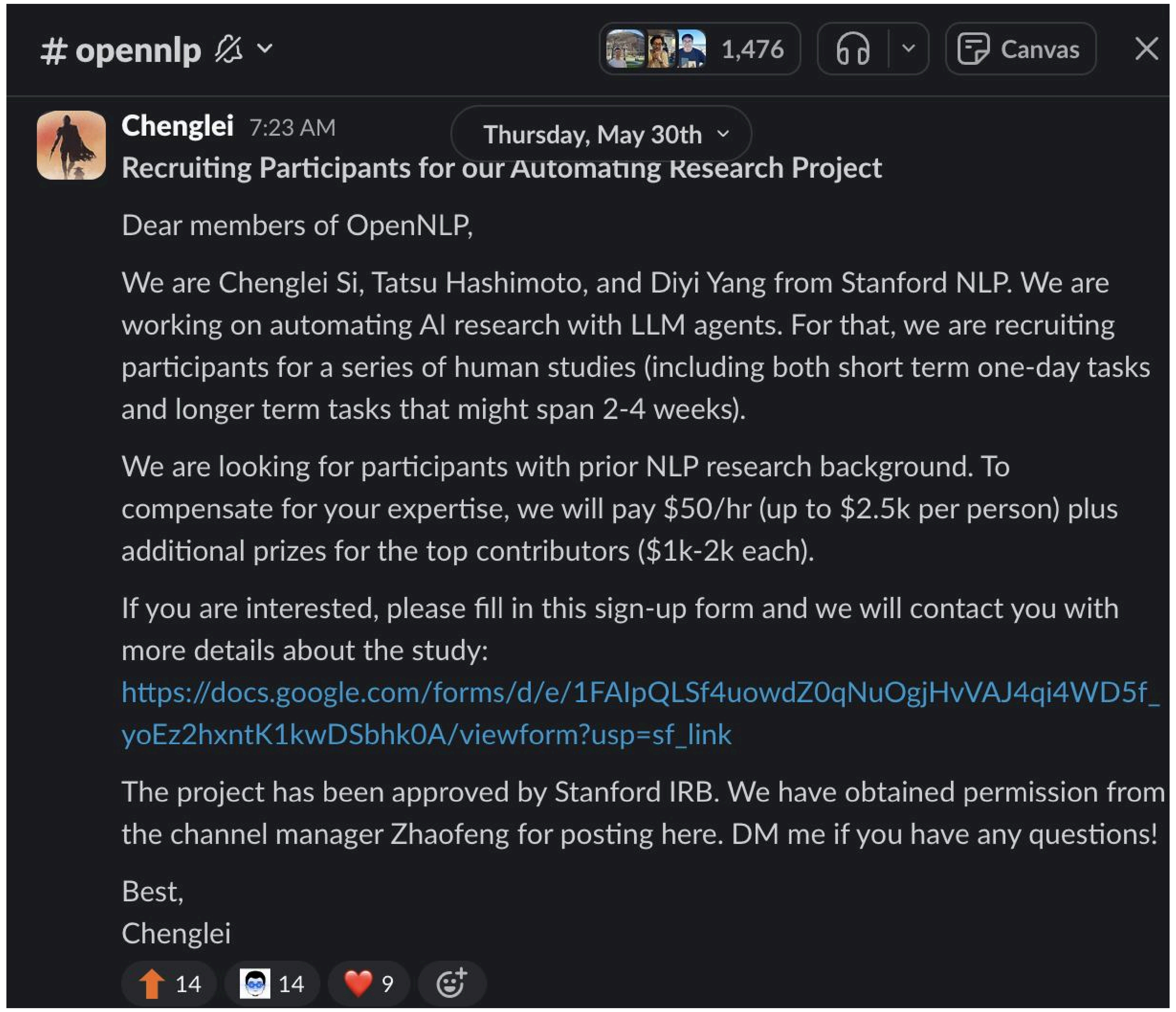
N	Top-10	Bottom-10	Gap
1	6.28	5.72	0.56
2	6.14	5.24	0.90
3	5.83	4.86	0.97
4	5.94	4.99	0.95
5	6.42	4.69	1.73
6	6.11	4.81	1.30

Outline

1. Study Design
2. Idea Generation Agent
3. **Human Experts**
4. Results
5. Analysis of Ideas & Reviews
6. Limitations of LLMs

Human Experts: Recruitment

Human Experts: Recruitment



Chenglei 7:23 AM Thursday, May 30th

Recruiting Participants for our Automating Research Project

Dear members of OpenNLP,

We are Chenglei Si, Tatsu Hashimoto, and Diyi Yang from Stanford NLP. We are working on automating AI research with LLM agents. For that, we are recruiting participants for a series of human studies (including both short term one-day tasks and longer term tasks that might span 2-4 weeks).

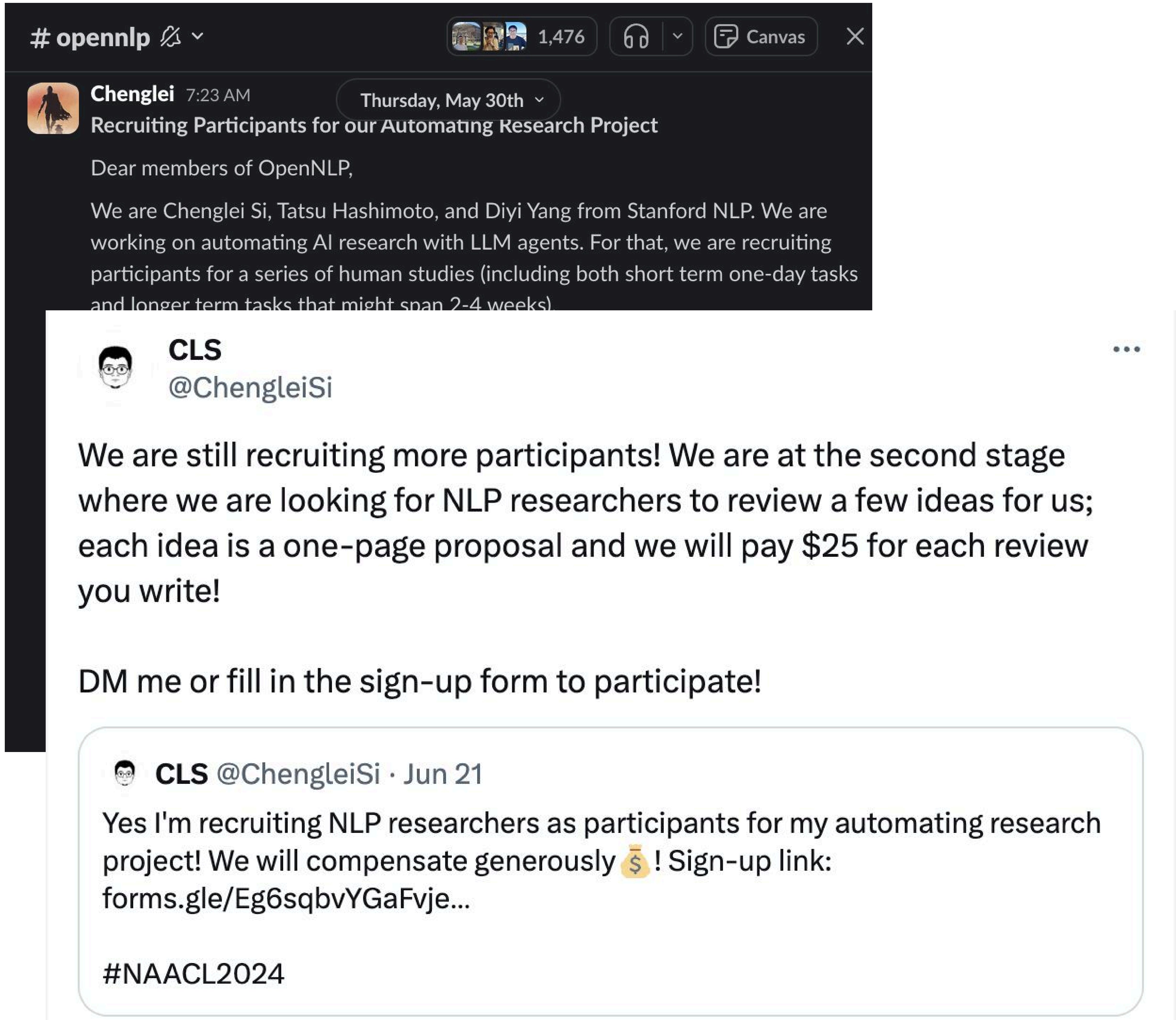
We are looking for participants with prior NLP research background. To compensate for your expertise, we will pay $50/hr (up to $2.5k per person) plus additional prizes for the top contributors ($1k-2k each).

If you are interested, please fill in this sign-up form and we will contact you with more details about the study: https://docs.google.com/forms/d/e/1FAIpQLSf4uowdZ0qNuOgjHvVAJ4qi4WD5f_yoEz2hxntK1kwDSbhk0A/viewform?usp=sf_link

The project has been approved by Stanford IRB. We have obtained permission from the channel manager Zhaofeng for posting here. DM me if you have any questions!

Best,
Chenglei

Human Experts: Recruitment



Chenglei 7:23 AM

Recruiting Participants for our Automating Research Project

Dear members of OpenNLP,

We are Chenglei Si, Tatsu Hashimoto, and Diyi Yang from Stanford NLP. We are working on automating AI research with LLM agents. For that, we are recruiting participants for a series of human studies (including both short term one-day tasks and longer term tasks that might span 2-4 weeks).

CLS
@ChengleiSi

We are still recruiting more participants! We are at the second stage where we are looking for NLP researchers to review a few ideas for us; each idea is a one-page proposal and we will pay $25 for each review you write!

DM me or fill in the sign-up form to participate!

CLS @ChengleiSi · Jun 21

Yes I'm recruiting NLP researchers as participants for my automating research project! We will compensate generously 💰! Sign-up link: forms.gle/Eg6sqbvYGaFvje...

#NAACL2024

Human Experts: Recruitment

Chenglei 7:23 AM
Recruiting Participants for our Automating Research Project

Dear members of OpenNLP,

We are Chenglei Si, Tatsu Hashimoto, and Diyi Yang from Stanford NLP. We are working on automating AI research with L
participants for a series of human studies
and longer term tasks that might span 2-4

CLS
@ChengleiSi

We are still recruiting more
where we are looking for NL
each idea is a one-page pro
you write!

DM me or fill in the sign-up

CLS @ChengleiSi · Jun 21
Yes I'm recruiting NLP researc
project! We will compensate
forms.gle/Eg6sqbvYGaFvje...

#NAACL2024

Human Experts: Recruitment

Writing an idea:

- 10 days
- $300
- $1000 bonus for top 5

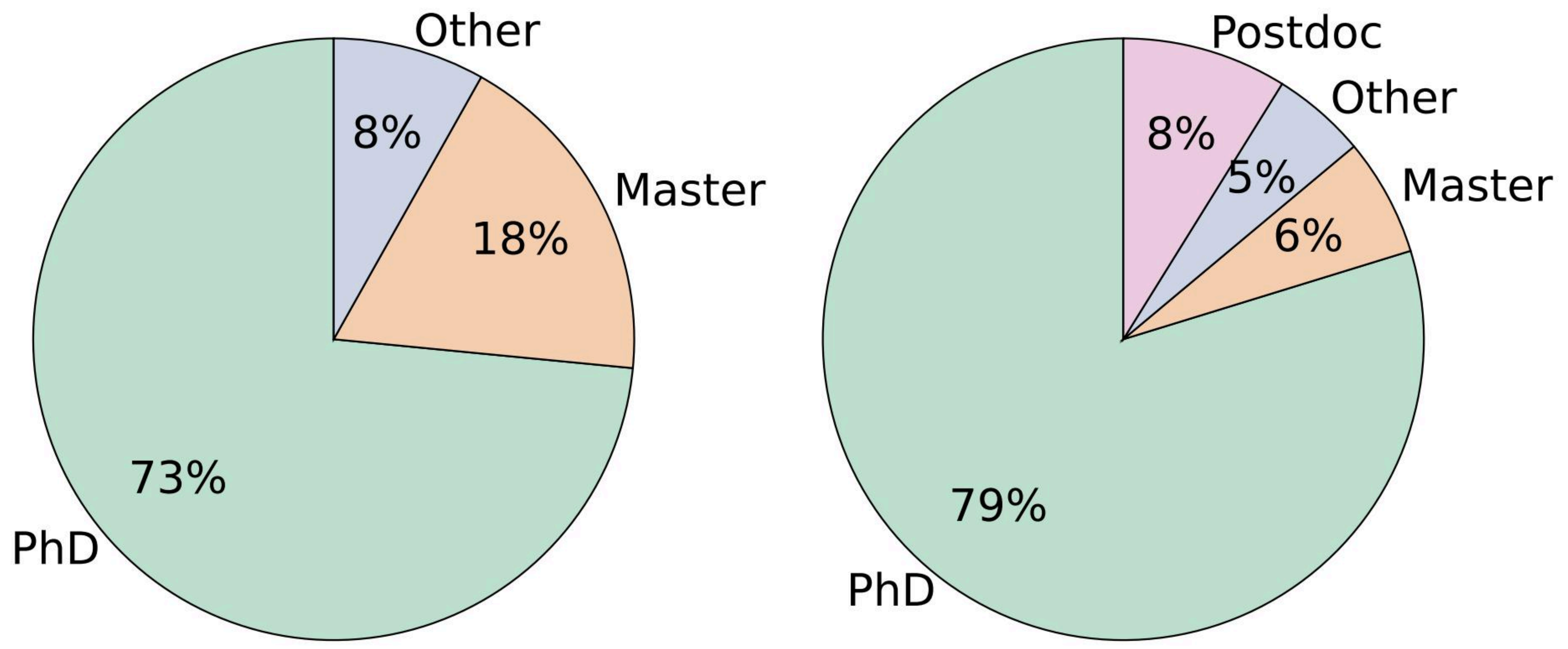
Reviewing an idea:

- one week
- $25

Human Experts: Recruitment

- N = 49 for writing ideas
- N = 79 for reviewing ideas
- 24 did both so N = 104 total participants

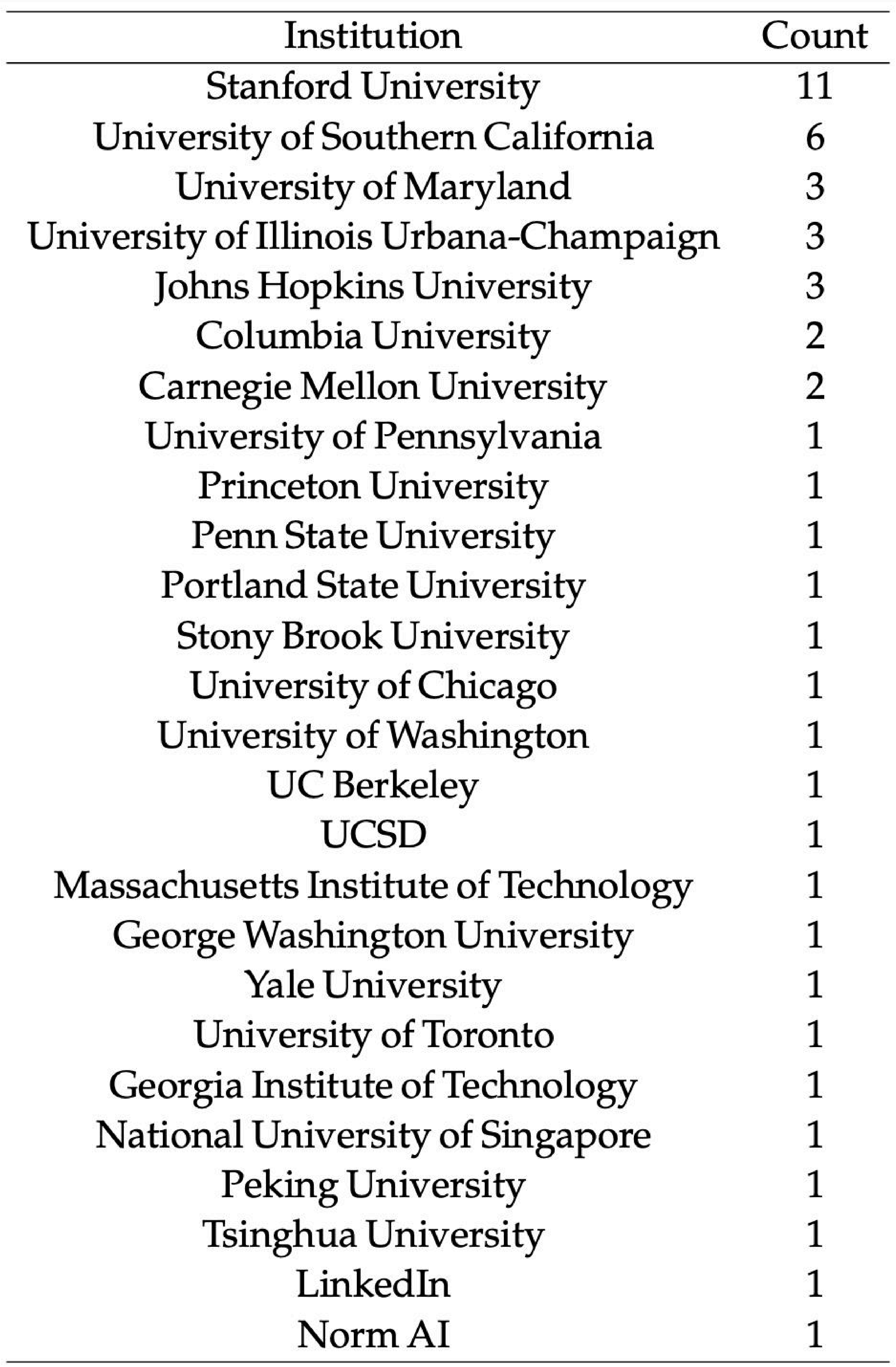
Human Experts: Qualifications



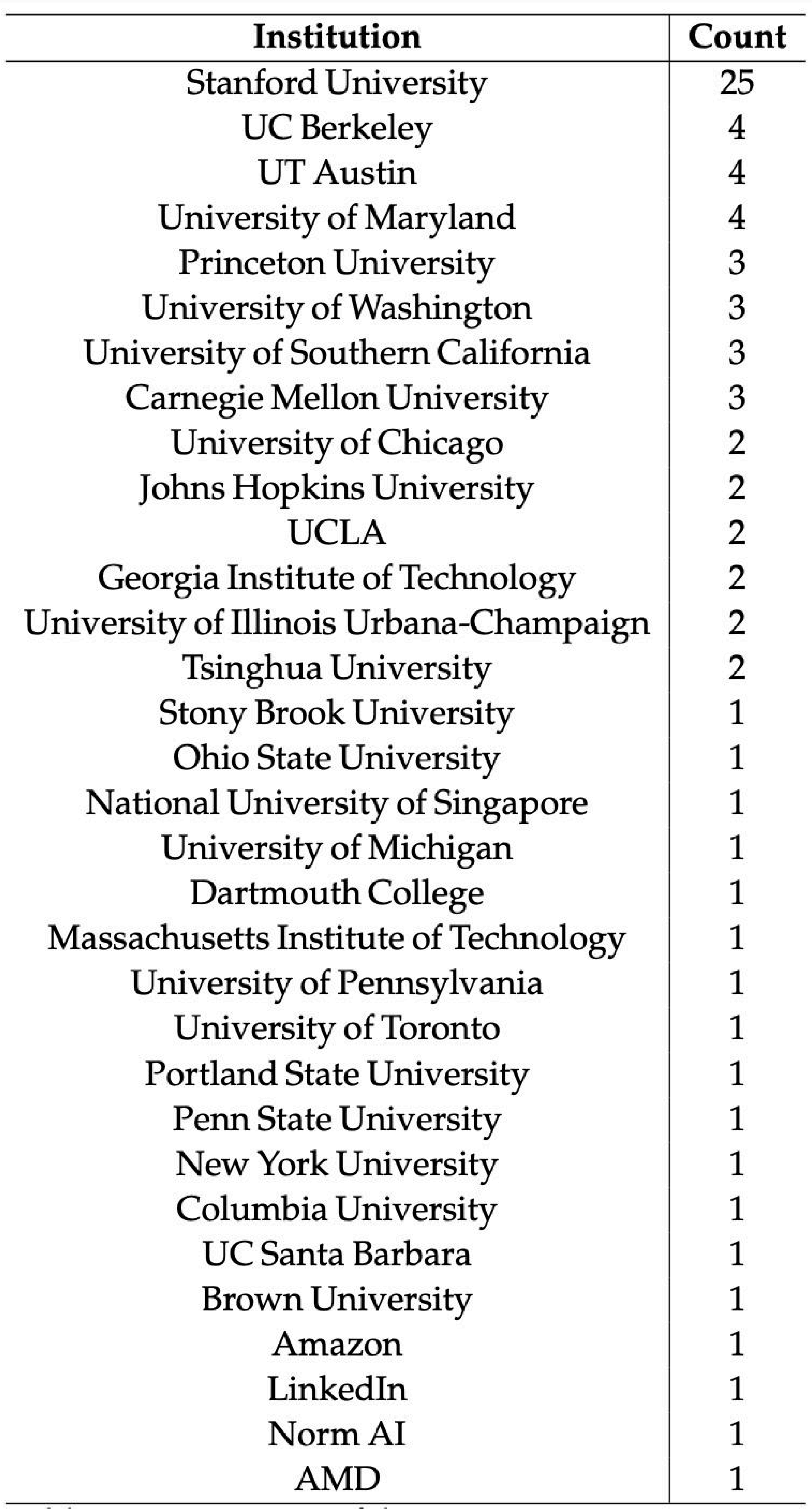
Human Experts: Qualifications

	Idea Writing Participants (N=49)					Idea Reviewing Participants (N=79)				
Metric	Mean	Median	Min	Max	SD	Mean	Median	Min	Max	SD
papers	12	10	2	52	9	15	13	2	52	10
citations	477	125	2	4553	861	635	327	0	7276	989
h-index	5	4	1	21	4	7	7	0	21	4
i10-index	5	4	0	32	6	7	5	0	32	6

Human Experts: Qualifications



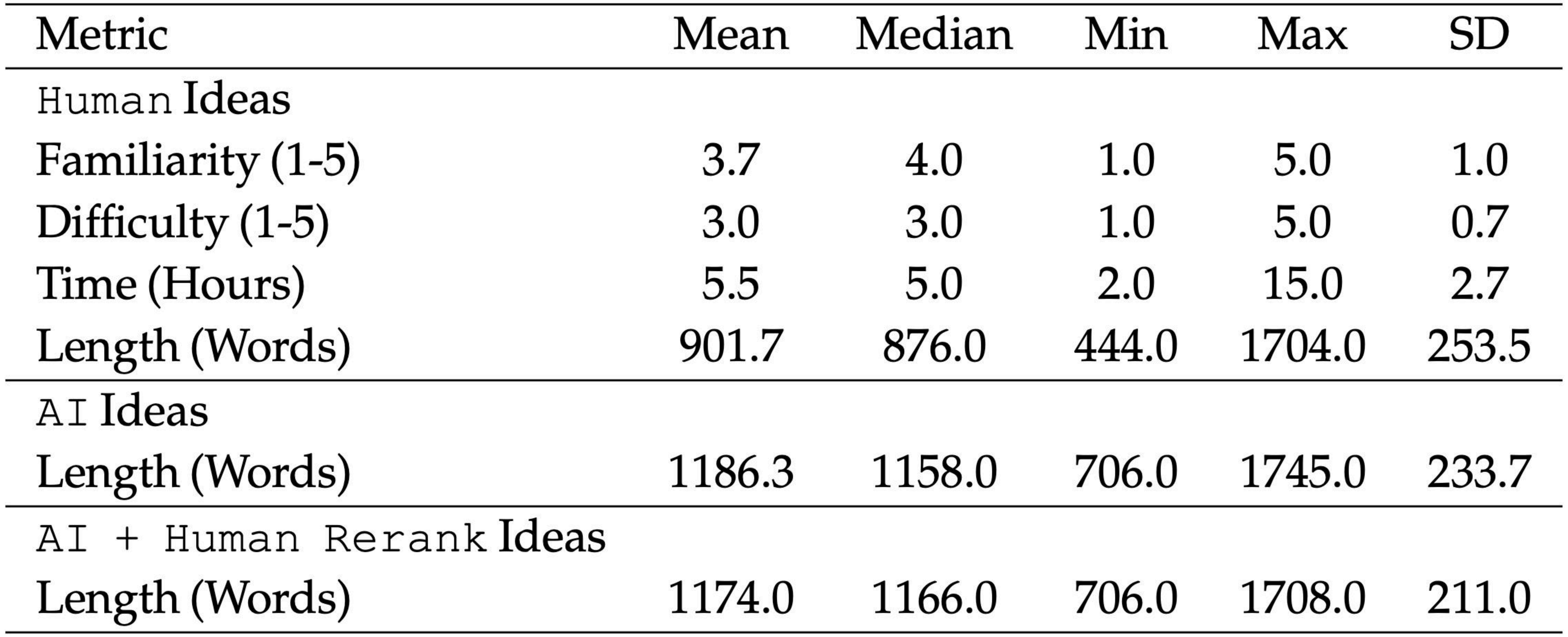
Institution	Count
Stanford University	11
University of Southern California	6
University of Maryland	3
University of Illinois Urbana-Champaign	3
Johns Hopkins University	3
Columbia University	2
Carnegie Mellon University	2
University of Pennsylvania	1
Princeton University	1
Penn State University	1
Portland State University	1
Stony Brook University	1
University of Chicago	1
University of Washington	1
UC Berkeley	1
UCSD	1
Massachusetts Institute of Technology	1
George Washington University	1
Yale University	1
University of Toronto	1
Georgia Institute of Technology	1
National University of Singapore	1
Peking University	1
Tsinghua University	1
LinkedIn	1
Norm AI	1



Institution	Count
Stanford University	25
UC Berkeley	4
UT Austin	4
University of Maryland	4
Princeton University	3
University of Washington	3
University of Southern California	3
Carnegie Mellon University	3
University of Chicago	2
Johns Hopkins University	2
UCLA	2
Georgia Institute of Technology	2
University of Illinois Urbana-Champaign	2
Tsinghua University	2
Stony Brook University	1
Ohio State University	1
National University of Singapore	1
University of Michigan	1
Dartmouth College	1
Massachusetts Institute of Technology	1
University of Pennsylvania	1
University of Toronto	1
Portland State University	1
Penn State University	1
New York University	1
Columbia University	1
UC Santa Barbara	1
Brown University	1
Amazon	1
LinkedIn	1
Norm AI	1
AMD	1

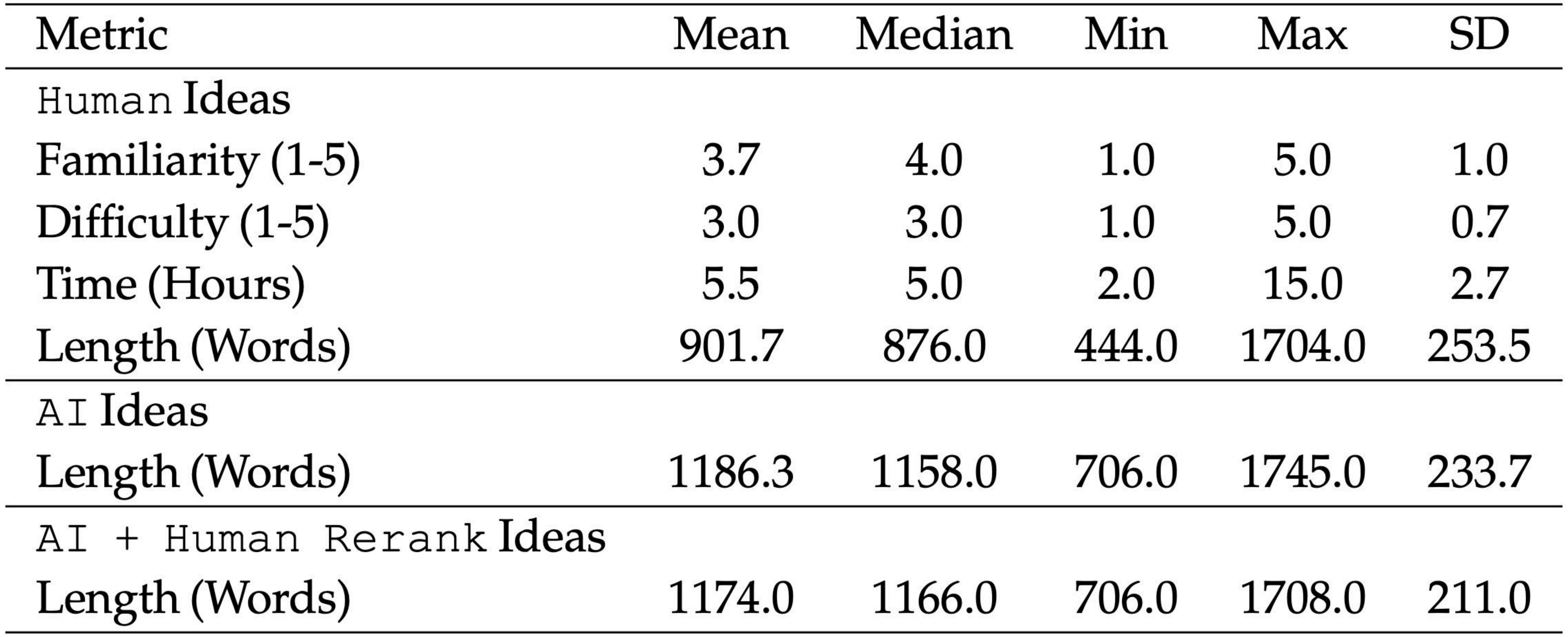
Human Experts: Efforts

Human Experts: Efforts (Ideas)

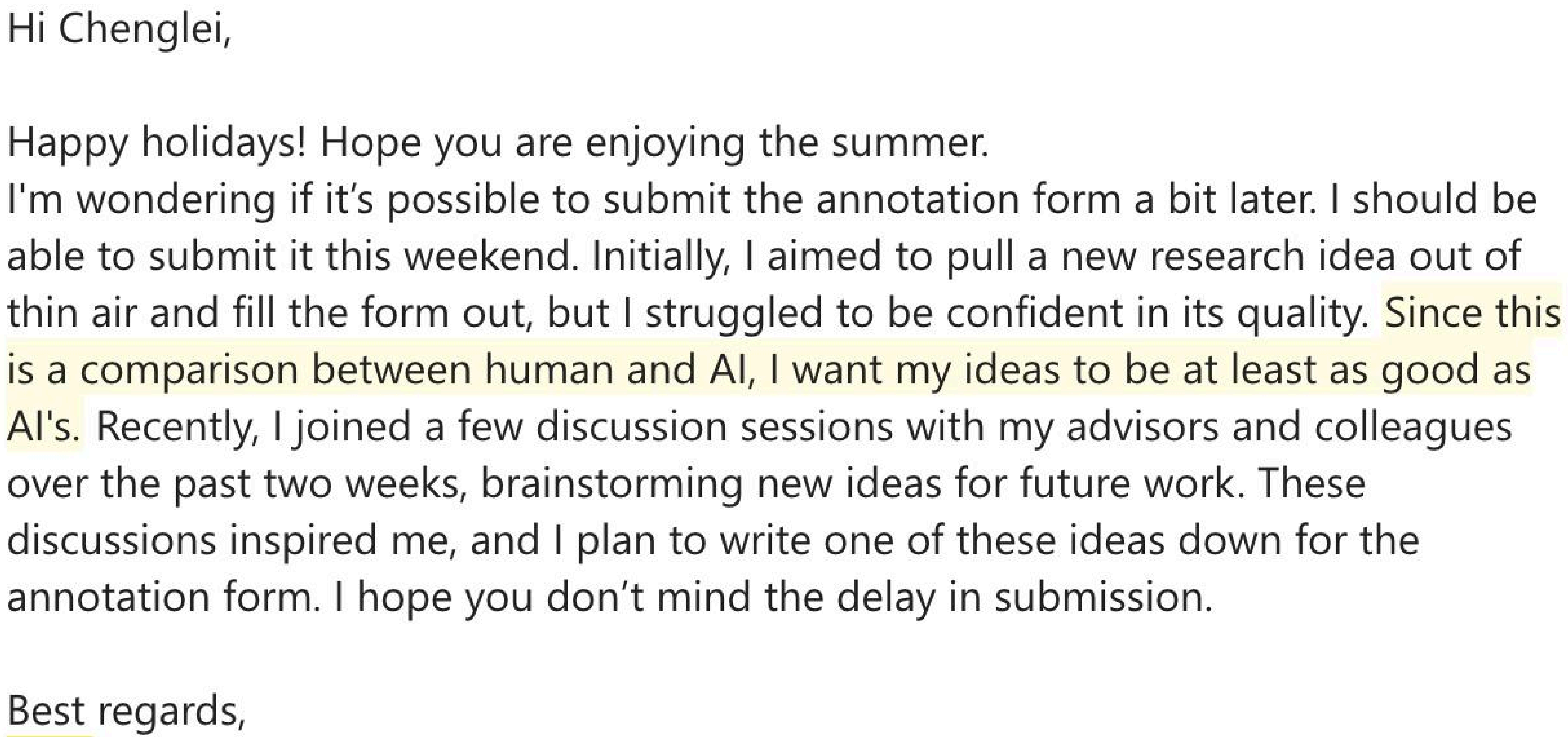


Metric	Mean	Median	Min	Max	SD
Human Ideas					
Familiarity (1-5)	3.7	4.0	1.0	5.0	1.0
Difficulty (1-5)	3.0	3.0	1.0	5.0	0.7
Time (Hours)	5.5	5.0	2.0	15.0	2.7
Length (Words)	901.7	876.0	444.0	1704.0	253.5
AI Ideas					
Length (Words)	1186.3	1158.0	706.0	1745.0	233.7
AI + Human Rerank Ideas					
Length (Words)	1174.0	1166.0	706.0	1708.0	211.0

Human Experts: Efforts (Ideas)



Metric	Mean	Median	Min	Max	SD
Human Ideas					
Familiarity (1-5)	3.7	4.0	1.0	5.0	1.0
Difficulty (1-5)	3.0	3.0	1.0	5.0	0.7
Time (Hours)	5.5	5.0	2.0	15.0	2.7
Length (Words)	901.7	876.0	444.0	1704.0	253.5
AI Ideas					
Length (Words)	1186.3	1158.0	706.0	1745.0	233.7
AI + Human Rerank Ideas					
Length (Words)	1174.0	1166.0	706.0	1708.0	211.0

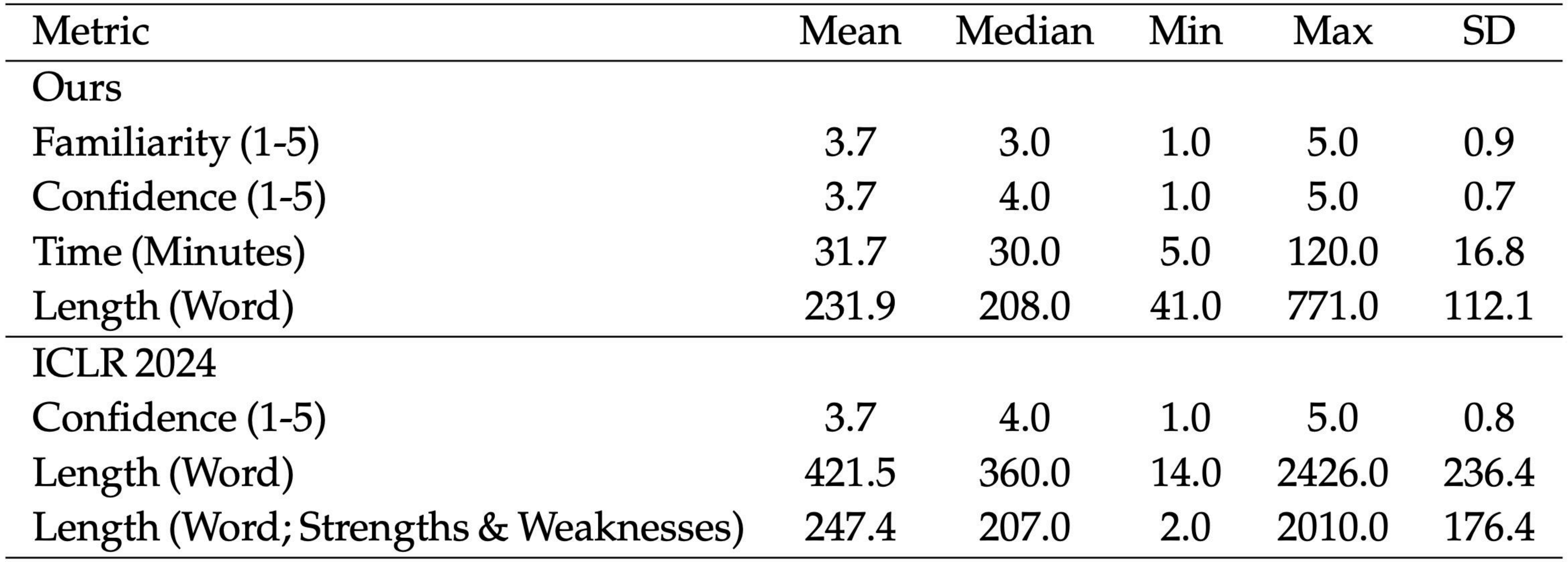


Hi Chenglei,

Happy holidays! Hope you are enjoying the summer.
I'm wondering if it's possible to submit the annotation form a bit later. I should be able to submit it this weekend. Initially, I aimed to pull a new research idea out of thin air and fill the form out, but I struggled to be confident in its quality. Since this is a comparison between human and AI, I want my ideas to be at least as good as AI's. Recently, I joined a few discussion sessions with my advisors and colleagues over the past two weeks, brainstorming new ideas for future work. These discussions inspired me, and I plan to write one of these ideas down for the annotation form. I hope you don't mind the delay in submission.

Best regards,

Human Experts: Efforts (Reviews)



Metric	Mean	Median	Min	Max	SD
Ours					
Familiarity (1-5)	3.7	3.0	1.0	5.0	0.9
Confidence (1-5)	3.7	4.0	1.0	5.0	0.7
Time (Minutes)	31.7	30.0	5.0	120.0	16.8
Length (Word)	231.9	208.0	41.0	771.0	112.1
ICLR 2024					
Confidence (1-5)	3.7	4.0	1.0	5.0	0.8
Length (Word)	421.5	360.0	14.0	2426.0	236.4
Length (Word; Strengths & Weaknesses)	247.4	207.0	2.0	2010.0	176.4

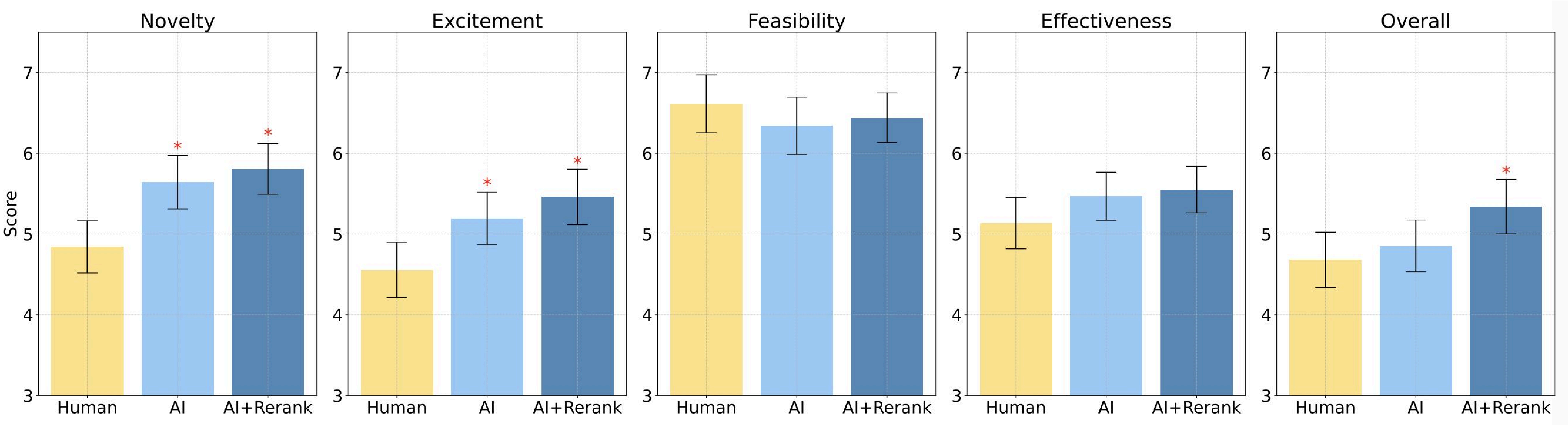
- **Out of the 298 unique reviews, 80 of them provided links to existing papers in their rationales to justify why the proposed method is not novel.**

Outline

1. Study Design
2. Idea Generation Agent
3. Human Experts
4. **Results**
5. Analysis of Ideas & Reviews
6. Limitations of LLMs

Results: Test 1

Each review as an independent data point


Novelty
Excitement
Feasibility
Effectiveness
Overall
Score
3
4
5
6
7
Human
AI
AI+Rerank
*

Condition	Size	Mean	Median	SD	SE	Min	Max	p-value
Novelty Score								
`Human Ideas`	119	4.84	5	1.79	0.16	1	8	–
`AI Ideas`	109	5.64	6	1.76	0.17	1	10	**0.00****
`AI Ideas + Human Rerank`	109	5.81	6	1.66	0.16	2	10	**0.00*****
Excitement Score								
`Human Ideas`	119	4.55	5	1.89	0.17	1	8	–
`AI Ideas`	109	5.19	6	1.73	0.17	1	9	**0.04***
`AI Ideas + Human Rerank`	109	5.46	6	1.82	0.17	1	9	**0.00****
Feasibility Score								
`Human Ideas`	119	6.61	7	1.99	0.18	1	10	–
`AI Ideas`	109	6.34	6	1.88	0.18	2	10	1.00
`AI Ideas + Human Rerank`	109	6.44	6	1.63	0.16	1	10	1.00
Expected Effectiveness Score								
`Human Ideas`	119	5.13	5	1.76	0.16	1	8	–
`AI Ideas`	109	5.47	6	1.58	0.15	1	10	0.67
`AI Ideas + Human Rerank`	109	5.55	6	1.52	0.15	1	9	0.29
Overall Score								
`Human Ideas`	119	4.68	5	1.90	0.17	1	9	–
`AI Ideas`	109	4.85	5	1.70	0.16	1	9	1.00
`AI Ideas + Human Rerank`	109	5.34	6	1.79	0.17	1	9	**0.04***

Results: Test 2

Each idea as an independent data point

Condition	Size	Mean	Median	SD	SE	Min	Max	p-value
Novelty Score								
Human Ideas	49	4.86	5.00	1.26	0.18	1.50	7.00	–
AI Ideas	49	5.62	5.50	1.39	0.20	1.50	8.33	**0.03***
AI Ideas + Human Rerank	49	5.78	6.00	1.07	0.15	3.00	8.33	**0.00****
Excitement Score								
Human Ideas	49	4.56	4.33	1.16	0.17	2.00	7.00	–
AI Ideas	49	5.18	5.50	1.33	0.19	2.50	7.33	0.08
AI Ideas + Human Rerank	49	5.45	5.50	1.36	0.19	1.00	7.33	**0.00****
Feasibility Score								
Human Ideas	49	6.53	7.00	1.50	0.21	3.00	9.00	–
AI Ideas	49	6.30	6.00	1.27	0.18	2.50	8.50	1.00
AI Ideas + Human Rerank	49	6.41	6.50	1.06	0.15	4.00	9.00	1.00
Expected Effectiveness Score								
Human Ideas	49	5.10	5.33	1.14	0.16	3.00	7.00	–
AI Ideas	49	5.48	5.50	1.23	0.18	2.00	7.50	0.58
AI Ideas + Human Rerank	49	5.57	5.50	0.99	0.14	3.00	7.50	0.17
Overall Score								
Human Ideas	49	4.69	4.67	1.16	0.17	2.00	6.67	–
AI Ideas	49	4.83	5.00	1.34	0.19	1.50	7.50	1.00
AI Ideas + Human Rerank	49	5.32	5.50	1.24	0.18	2.00	7.50	0.06

Results: Test 3

Each reviewer as an independent data point

	N	Mean Diff	p-value
Novelty Score			
AI Ideas vs Human Ideas	70	0.94	**0.00****
AI Ideas + Human Rerank vs Human Ideas	65	0.86	**0.00****
Excitement Score			
AI Ideas vs Human Ideas	70	0.73	**0.01***
AI Ideas + Human Rerank vs Human Ideas	65	0.87	**0.00****
Feasibility Score			
AI Ideas vs Human Ideas	70	-0.29	0.36
AI Ideas + Human Rerank vs Human Ideas	65	-0.08	0.74
Effectiveness Score			
AI Ideas vs Human Ideas	70	0.42	0.16
AI Ideas + Human Rerank vs Human Ideas	65	0.39	0.16
Overall Score			
AI Ideas vs Human Ideas	70	0.24	0.36
AI Ideas + Human Rerank vs Human Ideas	65	0.66	**0.01***

Results:

Conclusions that hold robustly

- Novelty: AI Ideas > Human Ideas
- Novelty: AI Ideas+ Human Rerank > Human Ideas
- Excitement: AI Ideas+ Human Rerank > Human Ideas

Outline

1. Study Design
2. Idea Generation Agent
3. Human Experts
4. Results
5. **Analysis of Ideas & Reviews**
6. Limitations of LLMs
7. Future Work

Analysis:

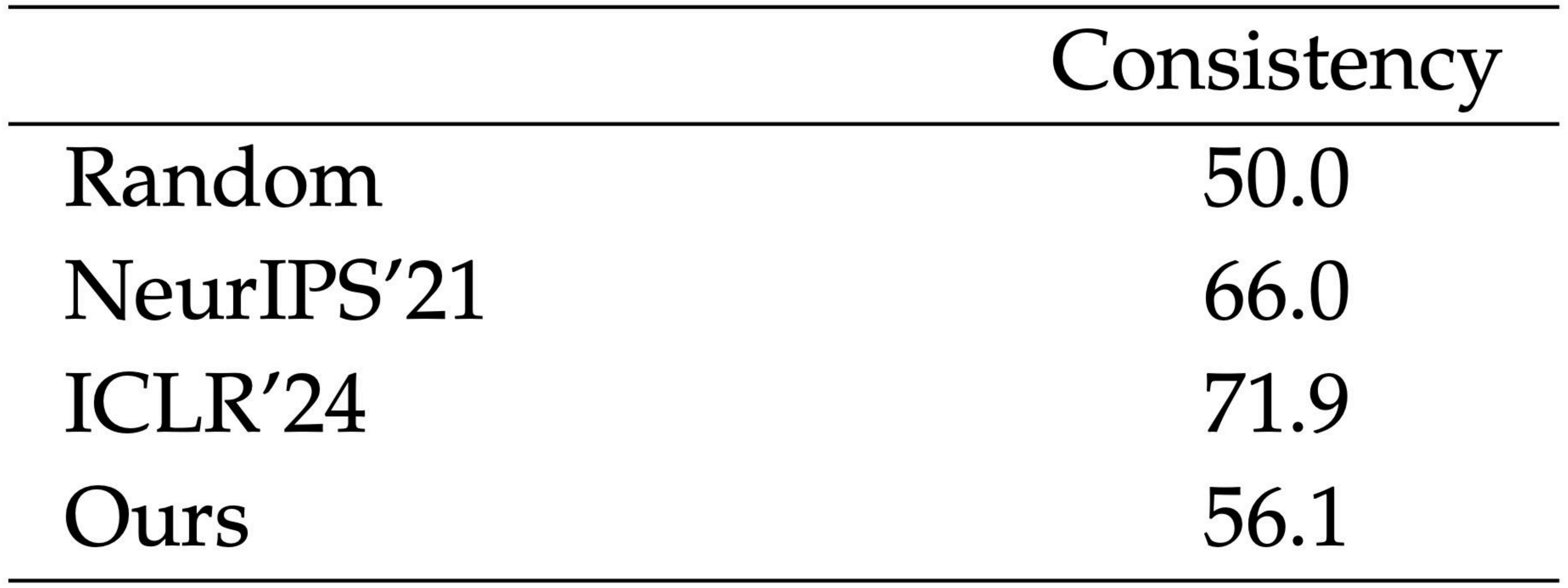
Expert Ideas

- 37 (out of 49) experts came up with the idea on the spot.

- Submitted ideas indicate top 43% of all their past ideas.

Analysis:
Expert Reviews

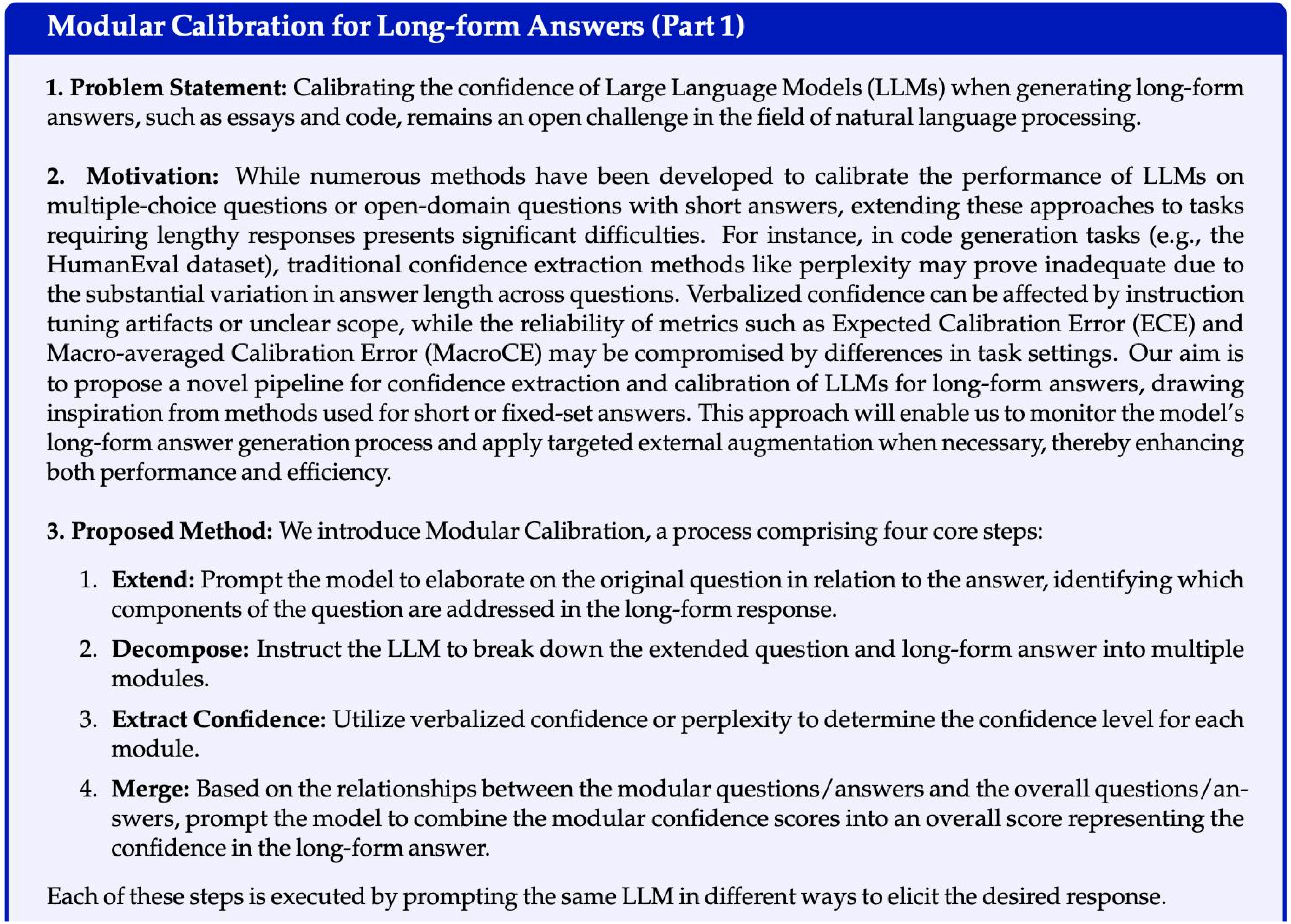
- **Reviewers have a relatively low agreement.**



	Consistency
Random	50.0
NeurIPS’21	66.0
ICLR’24	71.9
Ours	56.1

Analysis:

Example Idea #1



Modular Calibration for Long-form Answers (Part 1)

1. Problem Statement: Calibrating the confidence of Large Language Models (LLMs) when generating long-form answers, such as essays and code, remains an open challenge in the field of natural language processing.

2. Motivation: While numerous methods have been developed to calibrate the performance of LLMs on multiple-choice questions or open-domain questions with short answers, extending these approaches to tasks requiring lengthy responses presents significant difficulties. For instance, in code generation tasks (e.g., the HumanEval dataset), traditional confidence extraction methods like perplexity may prove inadequate due to the substantial variation in answer length across questions. Verbalized confidence can be affected by instruction tuning artifacts or unclear scope, while the reliability of metrics such as Expected Calibration Error (ECE) and Macro-averaged Calibration Error (MacroCE) may be compromised by differences in task settings. Our aim is to propose a novel pipeline for confidence extraction and calibration of LLMs for long-form answers, drawing inspiration from methods used for short or fixed-set answers. This approach will enable us to monitor the model's long-form answer generation process and apply targeted external augmentation when necessary, thereby enhancing both performance and efficiency.

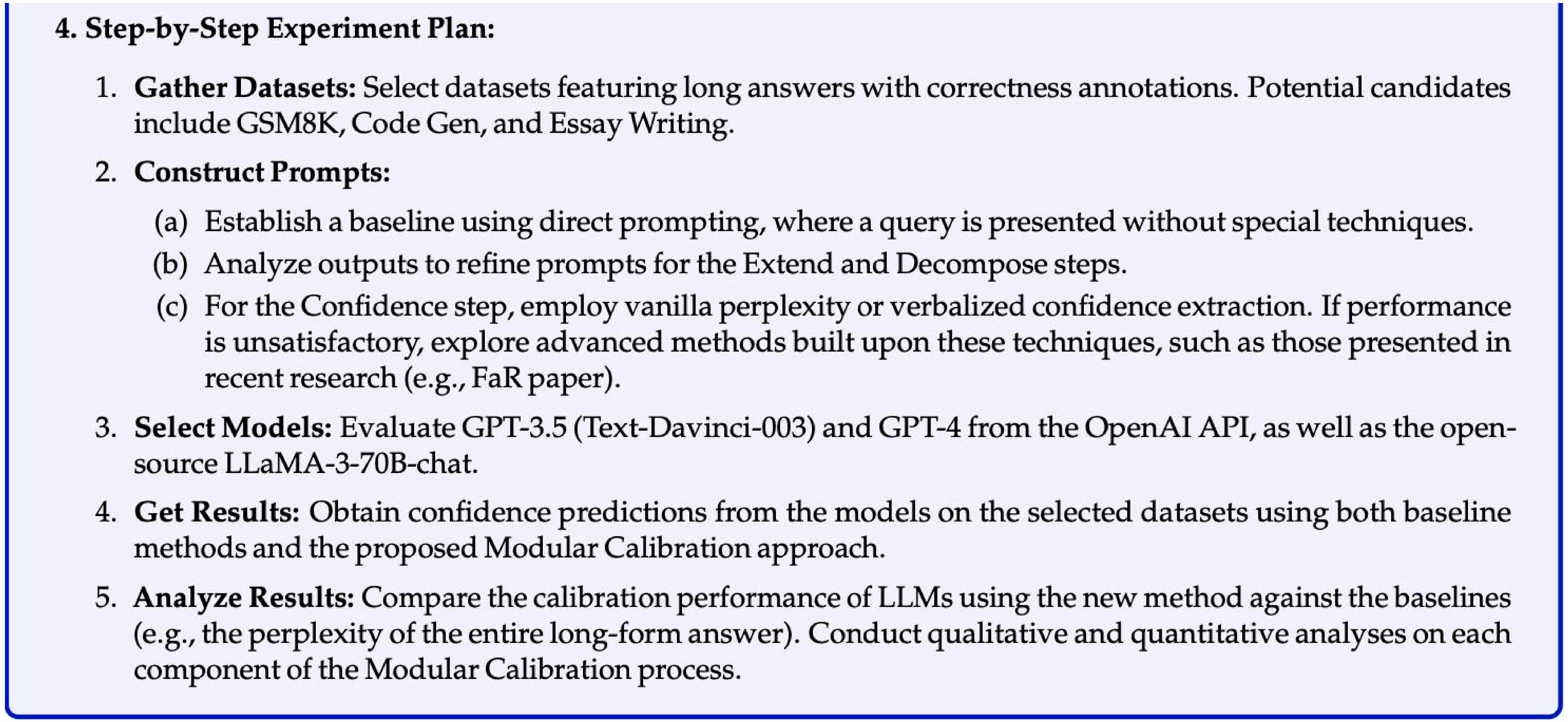
3. Proposed Method: We introduce Modular Calibration, a process comprising four core steps:

1. **Extend:** Prompt the model to elaborate on the original question in relation to the answer, identifying which components of the question are addressed in the long-form response.
2. **Decompose:** Instruct the LLM to break down the extended question and long-form answer into multiple modules.
3. **Extract Confidence:** Utilize verbalized confidence or perplexity to determine the confidence level for each module.
4. **Merge:** Based on the relationships between the modular questions/answers and the overall questions/answers, prompt the model to combine the modular confidence scores into an overall score representing the confidence in the long-form answer.

Each of these steps is executed by prompting the same LLM in different ways to elicit the desired response.

Analysis:

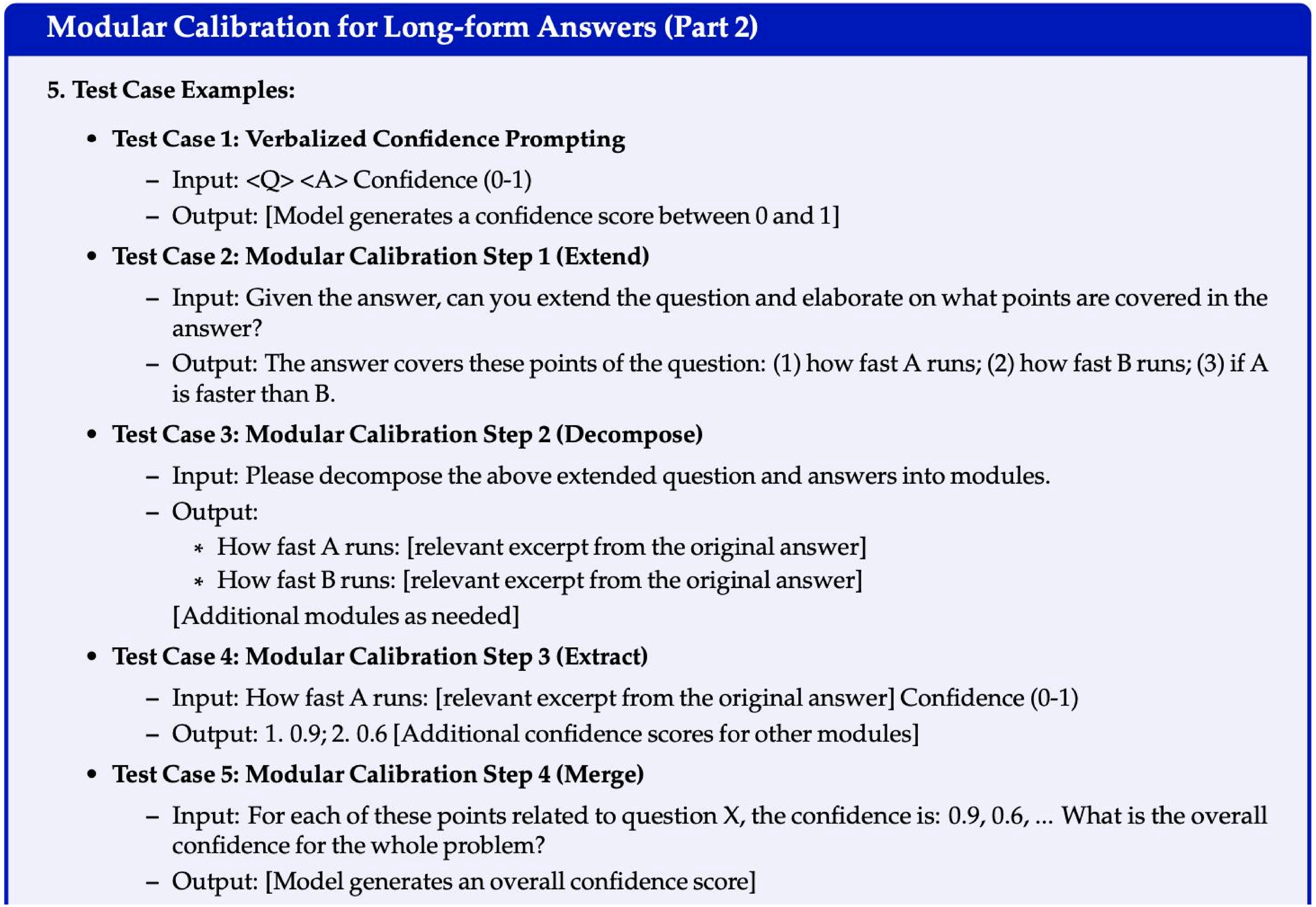
Example Idea #1



4. Step-by-Step Experiment Plan:

1. **Gather Datasets:** Select datasets featuring long answers with correctness annotations. Potential candidates include GSM8K, Code Gen, and Essay Writing.
2. **Construct Prompts:**
   (a) Establish a baseline using direct prompting, where a query is presented without special techniques.
   (b) Analyze outputs to refine prompts for the Extend and Decompose steps.
   (c) For the Confidence step, employ vanilla perplexity or verbalized confidence extraction. If performance is unsatisfactory, explore advanced methods built upon these techniques, such as those presented in recent research (e.g., FaR paper).
3. **Select Models:** Evaluate GPT-3.5 (Text-Davinci-003) and GPT-4 from the OpenAI API, as well as the open-source LLaMA-3-70B-chat.
4. **Get Results:** Obtain confidence predictions from the models on the selected datasets using both baseline methods and the proposed Modular Calibration approach.
5. **Analyze Results:** Compare the calibration performance of LLMs using the new method against the baselines (e.g., the perplexity of the entire long-form answer). Conduct qualitative and quantitative analyses on each component of the Modular Calibration process.

Analysis:
Example Idea #1



Modular Calibration for Long-form Answers (Part 2)

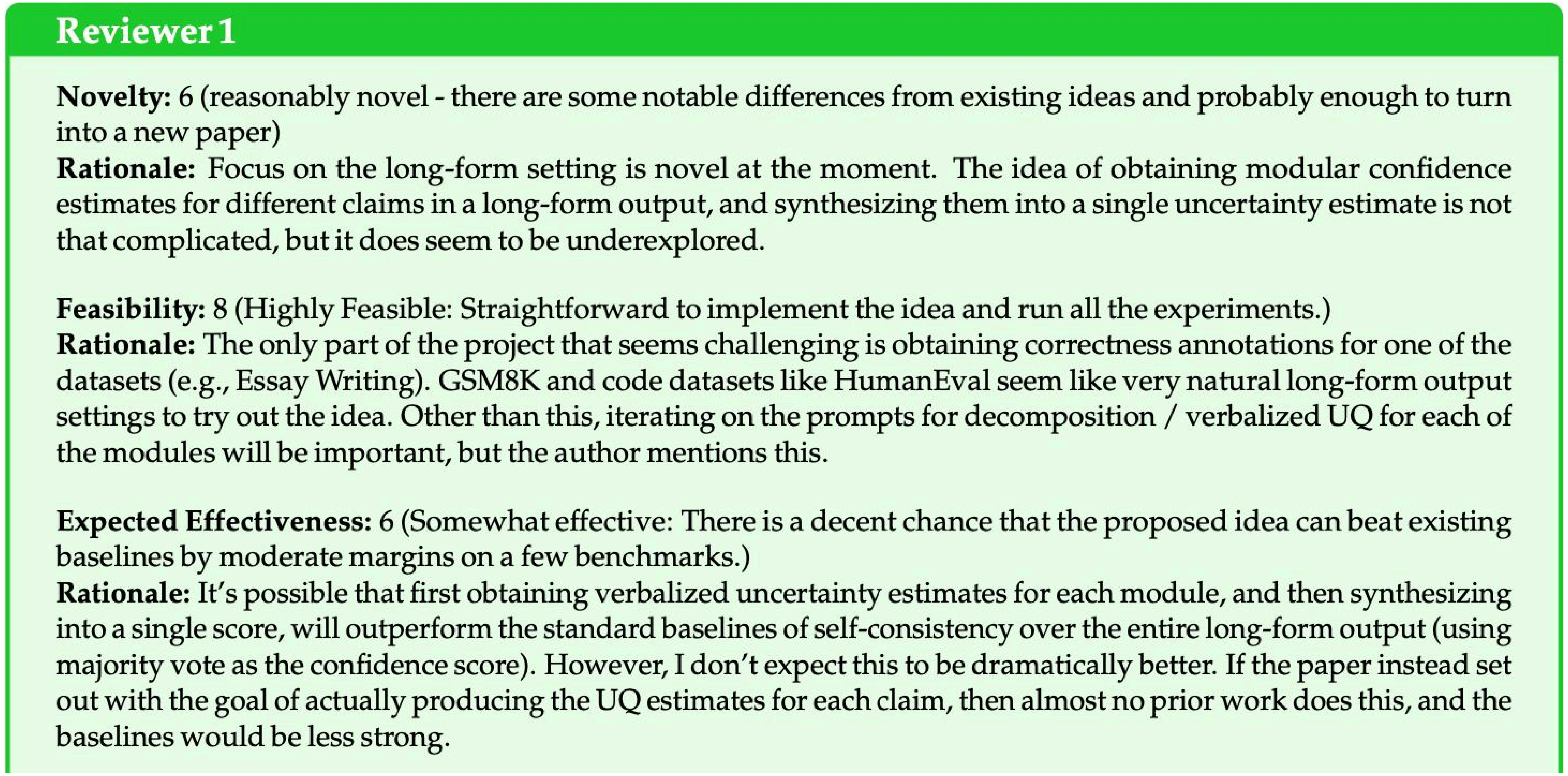
5. Test Case Examples:

- **Test Case 1: Verbalized Confidence Prompting**
  - Input: <Q> <A> Confidence (0-1)
  - Output: [Model generates a confidence score between 0 and 1]
- **Test Case 2: Modular Calibration Step 1 (Extend)**
  - Input: Given the answer, can you extend the question and elaborate on what points are covered in the answer?
  - Output: The answer covers these points of the question: (1) how fast A runs; (2) how fast B runs; (3) if A is faster than B.
- **Test Case 3: Modular Calibration Step 2 (Decompose)**
  - Input: Please decompose the above extended question and answers into modules.
  - Output:
    - How fast A runs: [relevant excerpt from the original answer]
    - How fast B runs: [relevant excerpt from the original answer]

    [Additional modules as needed]
- **Test Case 4: Modular Calibration Step 3 (Extract)**
  - Input: How fast A runs: [relevant excerpt from the original answer] Confidence (0-1)
  - Output: 1. 0.9; 2. 0.6 [Additional confidence scores for other modules]
- **Test Case 5: Modular Calibration Step 4 (Merge)**
  - Input: For each of these points related to question X, the confidence is: 0.9, 0.6, ... What is the overall confidence for the whole problem?
  - Output: [Model generates an overall confidence score]

Analysis:

Example Review #1



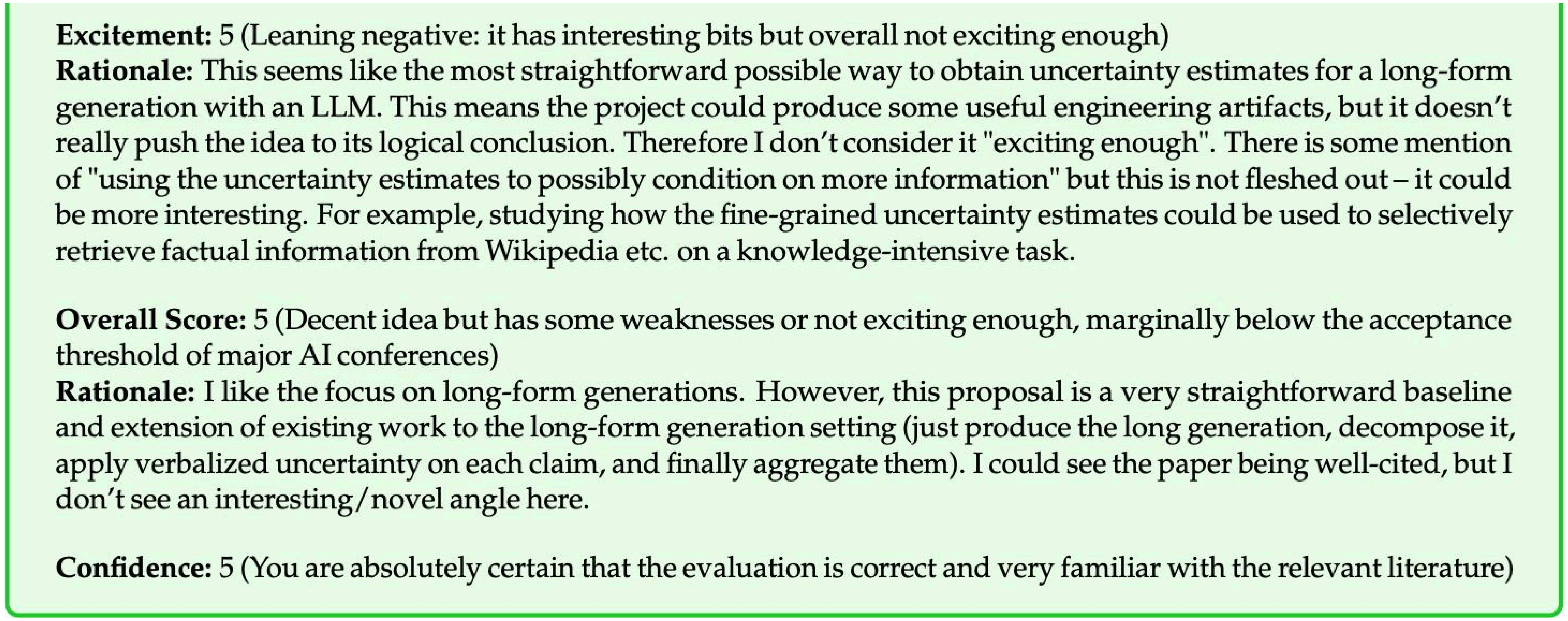
Reviewer 1

Novelty: 6 (reasonably novel - there are some notable differences from existing ideas and probably enough to turn into a new paper)
Rationale: Focus on the long-form setting is novel at the moment. The idea of obtaining modular confidence estimates for different claims in a long-form output, and synthesizing them into a single uncertainty estimate is not that complicated, but it does seem to be underexplored.

Feasibility: 8 (Highly Feasible: Straightforward to implement the idea and run all the experiments.)
Rationale: The only part of the project that seems challenging is obtaining correctness annotations for one of the datasets (e.g., Essay Writing). GSM8K and code datasets like HumanEval seem like very natural long-form output settings to try out the idea. Other than this, iterating on the prompts for decomposition / verbalized UQ for each of the modules will be important, but the author mentions this.

Expected Effectiveness: 6 (Somewhat effective: There is a decent chance that the proposed idea can beat existing baselines by moderate margins on a few benchmarks.)
Rationale: It's possible that first obtaining verbalized uncertainty estimates for each module, and then synthesizing into a single score, will outperform the standard baselines of self-consistency over the entire long-form output (using majority vote as the confidence score). However, I don't expect this to be dramatically better. If the paper instead set out with the goal of actually producing the UQ estimates for each claim, then almost no prior work does this, and the baselines would be less strong.

Analysis:
Example Review #1



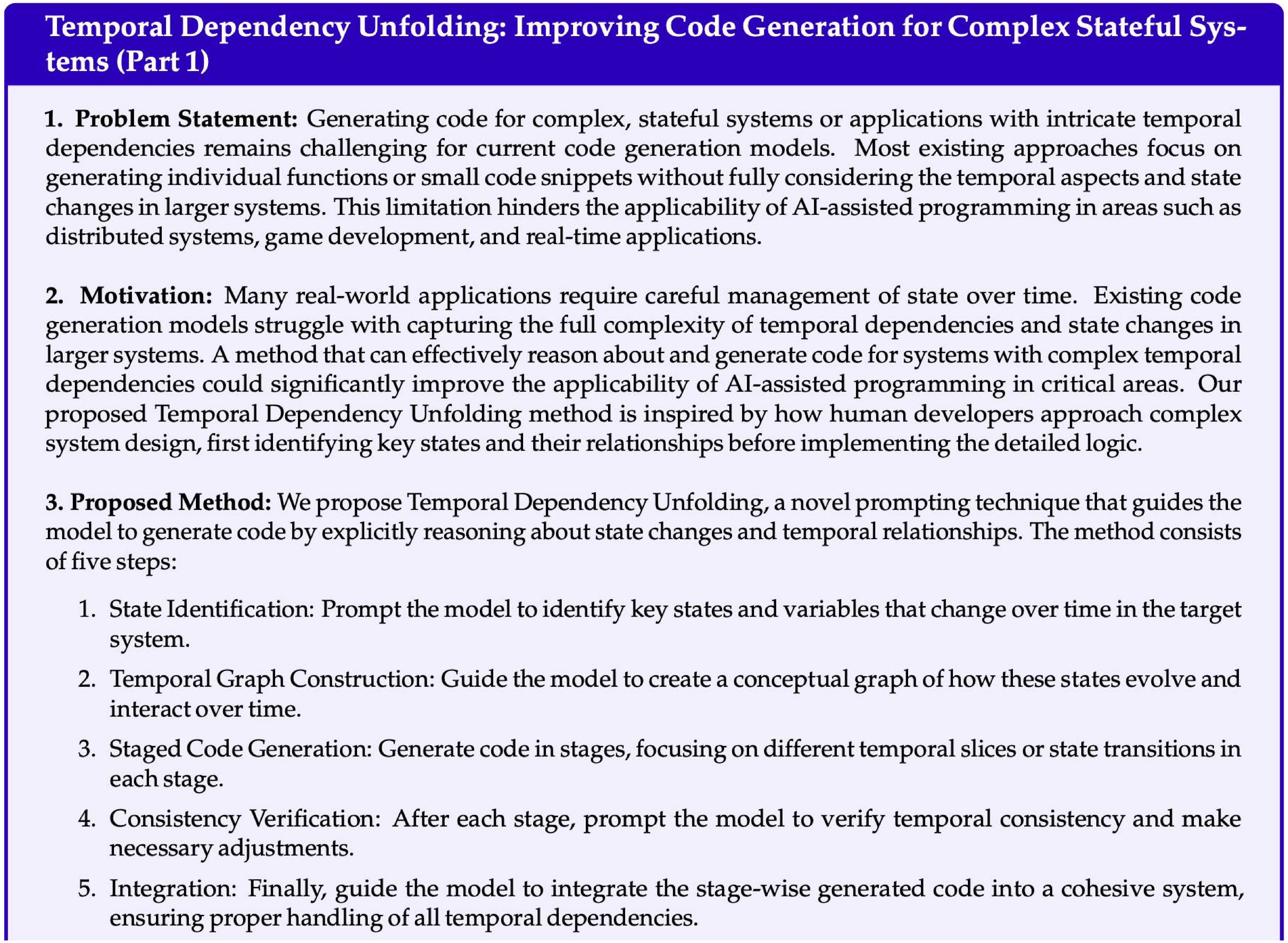
Excitement: 5 (Leaning negative: it has interesting bits but overall not exciting enough)
Rationale: This seems like the most straightforward possible way to obtain uncertainty estimates for a long-form generation with an LLM. This means the project could produce some useful engineering artifacts, but it doesn't really push the idea to its logical conclusion. Therefore I don't consider it "exciting enough". There is some mention of "using the uncertainty estimates to possibly condition on more information" but this is not fleshed out – it could be more interesting. For example, studying how the fine-grained uncertainty estimates could be used to selectively retrieve factual information from Wikipedia etc. on a knowledge-intensive task.

Overall Score: 5 (Decent idea but has some weaknesses or not exciting enough, marginally below the acceptance threshold of major AI conferences)
Rationale: I like the focus on long-form generations. However, this proposal is a very straightforward baseline and extension of existing work to the long-form generation setting (just produce the long generation, decompose it, apply verbalized uncertainty on each claim, and finally aggregate them). I could see the paper being well-cited, but I don't see an interesting/novel angle here.

Confidence: 5 (You are absolutely certain that the evaluation is correct and very familiar with the relevant literature)

Analysis:

Example Idea #2



Temporal Dependency Unfolding: Improving Code Generation for Complex Stateful Systems (Part 1)

1. Problem Statement: Generating code for complex, stateful systems or applications with intricate temporal dependencies remains challenging for current code generation models. Most existing approaches focus on generating individual functions or small code snippets without fully considering the temporal aspects and state changes in larger systems. This limitation hinders the applicability of AI-assisted programming in areas such as distributed systems, game development, and real-time applications.

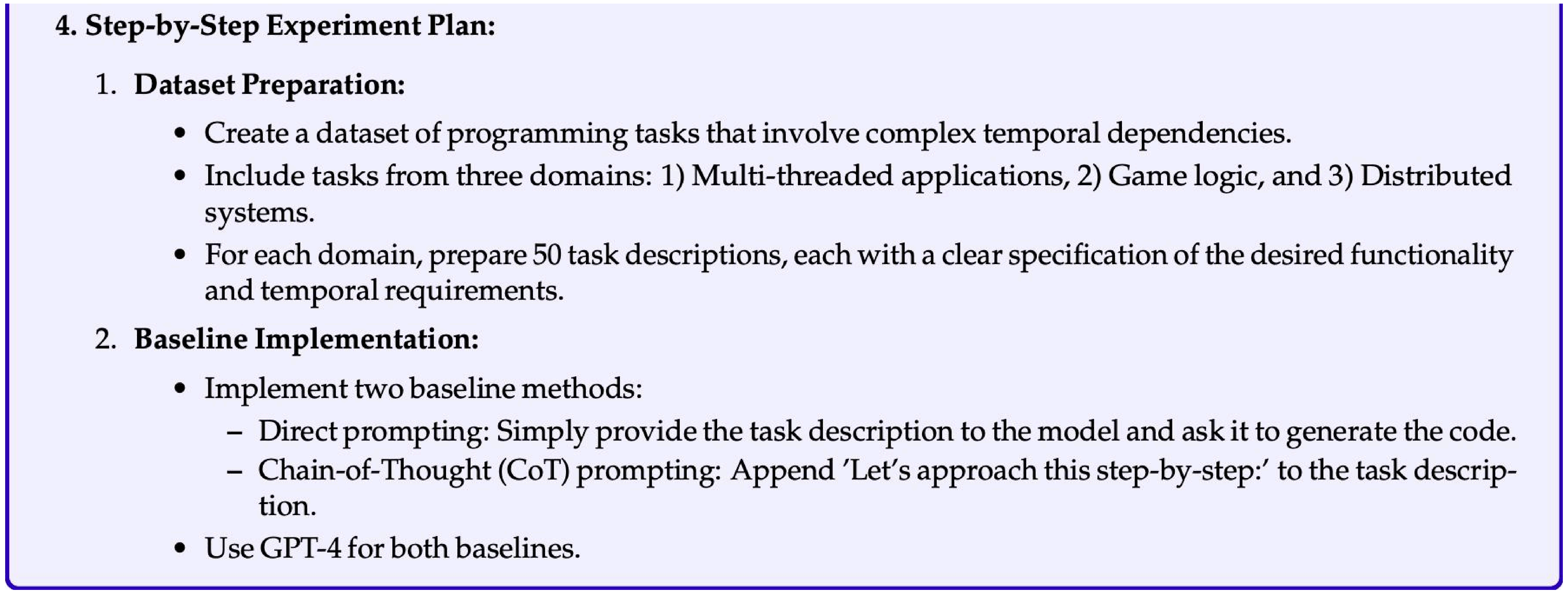
2. Motivation: Many real-world applications require careful management of state over time. Existing code generation models struggle with capturing the full complexity of temporal dependencies and state changes in larger systems. A method that can effectively reason about and generate code for systems with complex temporal dependencies could significantly improve the applicability of AI-assisted programming in critical areas. Our proposed Temporal Dependency Unfolding method is inspired by how human developers approach complex system design, first identifying key states and their relationships before implementing the detailed logic.

3. Proposed Method: We propose Temporal Dependency Unfolding, a novel prompting technique that guides the model to generate code by explicitly reasoning about state changes and temporal relationships. The method consists of five steps:

1. State Identification: Prompt the model to identify key states and variables that change over time in the target system.
2. Temporal Graph Construction: Guide the model to create a conceptual graph of how these states evolve and interact over time.
3. Staged Code Generation: Generate code in stages, focusing on different temporal slices or state transitions in each stage.
4. Consistency Verification: After each stage, prompt the model to verify temporal consistency and make necessary adjustments.
5. Integration: Finally, guide the model to integrate the stage-wise generated code into a cohesive system, ensuring proper handling of all temporal dependencies.

Analysis:

Example Idea #2

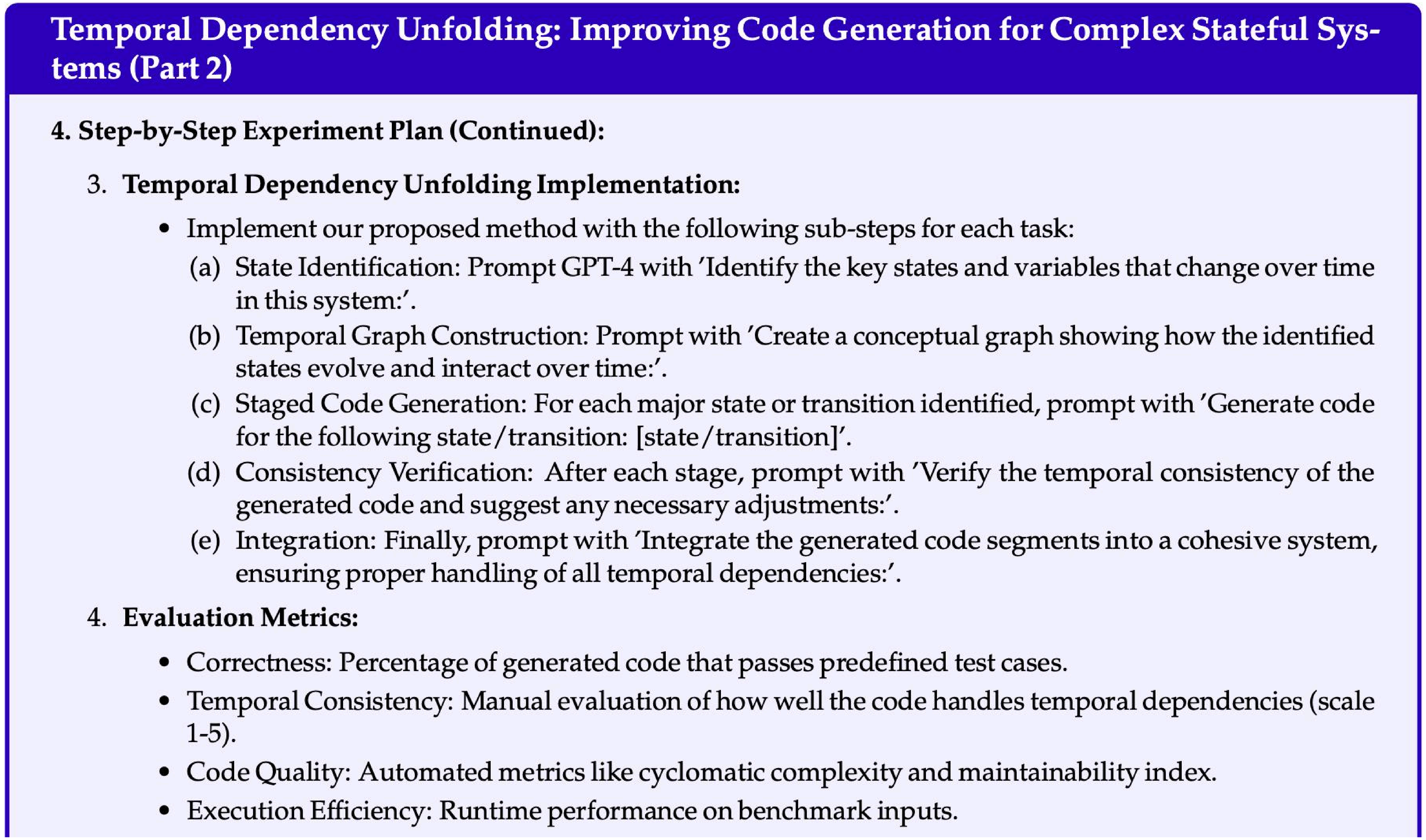


4. Step-by-Step Experiment Plan:

1. **Dataset Preparation:**
   - Create a dataset of programming tasks that involve complex temporal dependencies.
   - Include tasks from three domains: 1) Multi-threaded applications, 2) Game logic, and 3) Distributed systems.
   - For each domain, prepare 50 task descriptions, each with a clear specification of the desired functionality and temporal requirements.
2. **Baseline Implementation:**
   - Implement two baseline methods:
     - Direct prompting: Simply provide the task description to the model and ask it to generate the code.
     - Chain-of-Thought (CoT) prompting: Append 'Let's approach this step-by-step:' to the task description.
   - Use GPT-4 for both baselines.

Analysis:

Example Idea #2



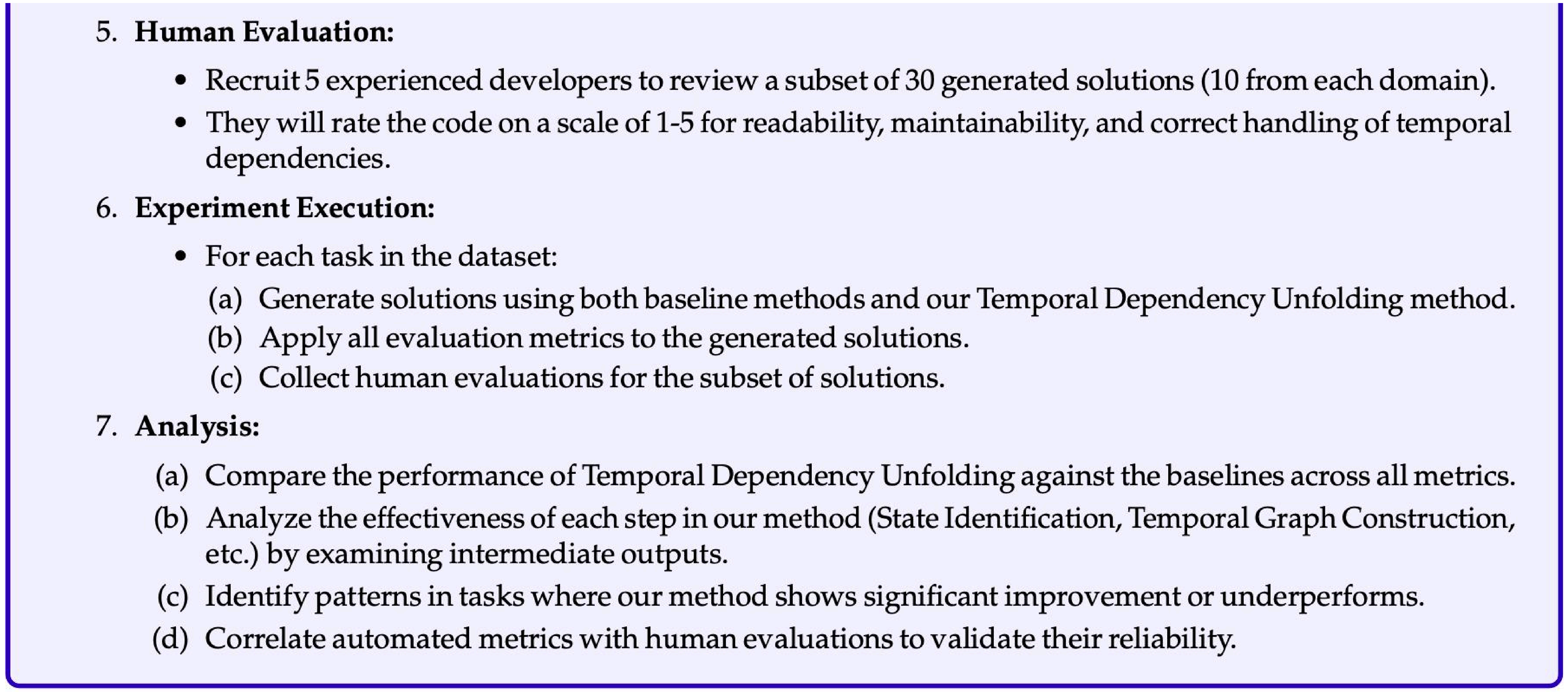
Temporal Dependency Unfolding: Improving Code Generation for Complex Stateful Systems (Part 2)

4. Step-by-Step Experiment Plan (Continued):

3. **Temporal Dependency Unfolding Implementation:**
   - Implement our proposed method with the following sub-steps for each task:
     (a) State Identification: Prompt GPT-4 with ’Identify the key states and variables that change over time in this system:’.
     (b) Temporal Graph Construction: Prompt with ’Create a conceptual graph showing how the identified states evolve and interact over time:’.
     (c) Staged Code Generation: For each major state or transition identified, prompt with ’Generate code for the following state/transition: [state/transition]’.
     (d) Consistency Verification: After each stage, prompt with ’Verify the temporal consistency of the generated code and suggest any necessary adjustments:’.
     (e) Integration: Finally, prompt with ’Integrate the generated code segments into a cohesive system, ensuring proper handling of all temporal dependencies:’.
4. **Evaluation Metrics:**
   - Correctness: Percentage of generated code that passes predefined test cases.
   - Temporal Consistency: Manual evaluation of how well the code handles temporal dependencies (scale 1-5).
   - Code Quality: Automated metrics like cyclomatic complexity and maintainability index.
   - Execution Efficiency: Runtime performance on benchmark inputs.

Analysis:

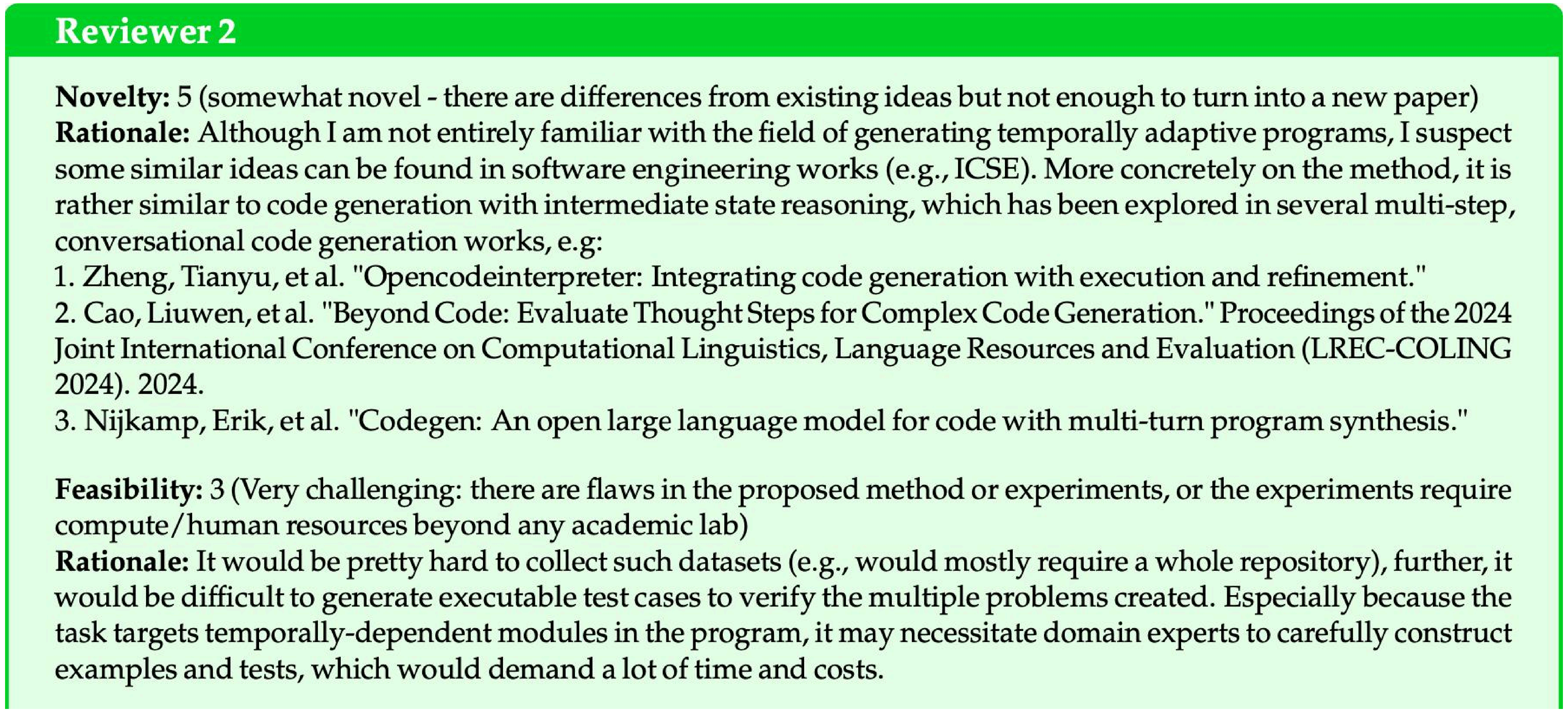
Example Idea #2



5. **Human Evaluation:**
   - Recruit 5 experienced developers to review a subset of 30 generated solutions (10 from each domain).
   - They will rate the code on a scale of 1-5 for readability, maintainability, and correct handling of temporal dependencies.
6. **Experiment Execution:**
   - For each task in the dataset:
     (a) Generate solutions using both baseline methods and our Temporal Dependency Unfolding method.
     (b) Apply all evaluation metrics to the generated solutions.
     (c) Collect human evaluations for the subset of solutions.
7. **Analysis:**
   (a) Compare the performance of Temporal Dependency Unfolding against the baselines across all metrics.
   (b) Analyze the effectiveness of each step in our method (State Identification, Temporal Graph Construction, etc.) by examining intermediate outputs.
   (c) Identify patterns in tasks where our method shows significant improvement or underperforms.
   (d) Correlate automated metrics with human evaluations to validate their reliability.

Analysis:

Example Review #2



Reviewer 2

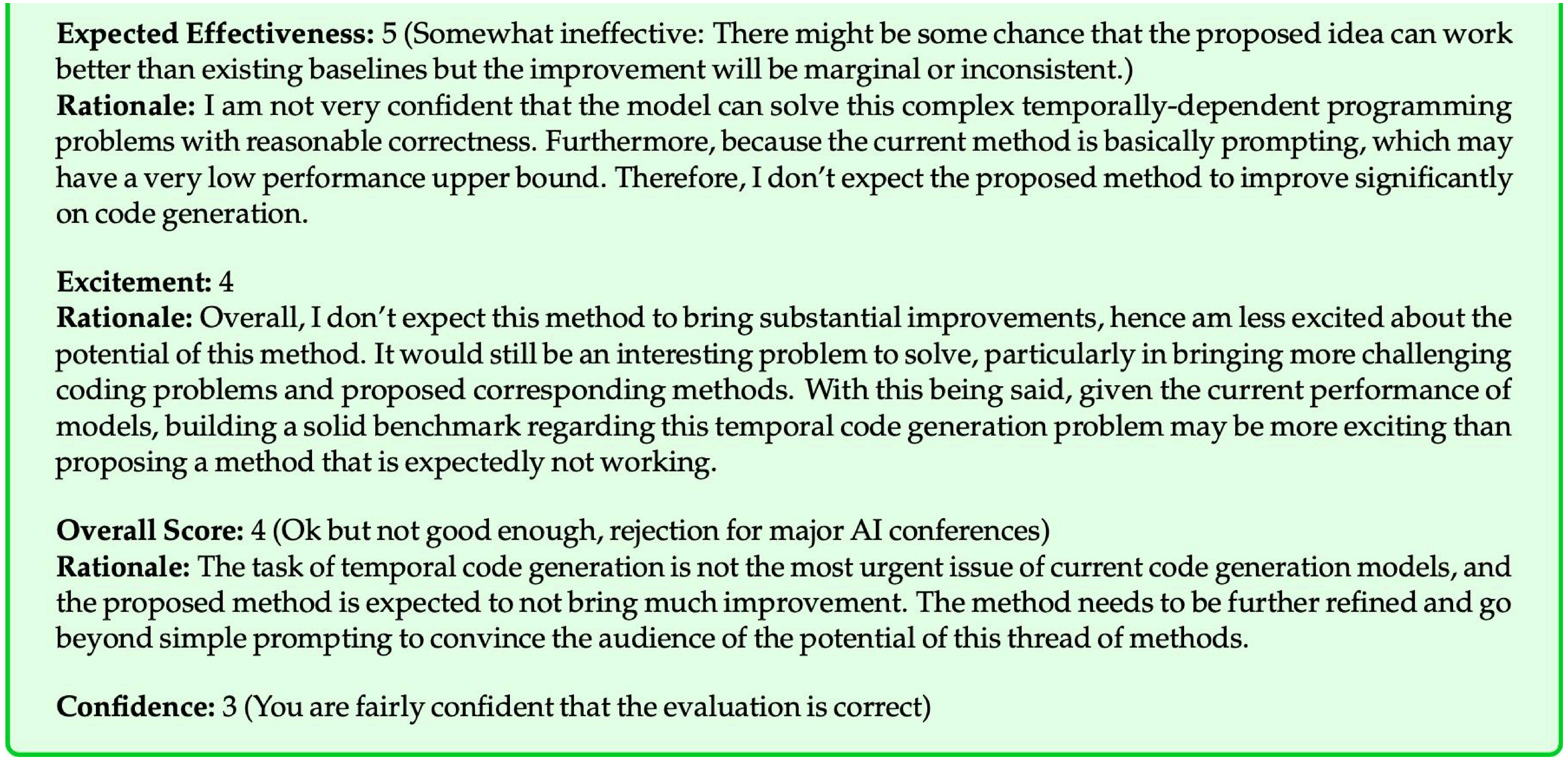
Novelty: 5 (somewhat novel - there are differences from existing ideas but not enough to turn into a new paper)
Rationale: Although I am not entirely familiar with the field of generating temporally adaptive programs, I suspect some similar ideas can be found in software engineering works (e.g., ICSE). More concretely on the method, it is rather similar to code generation with intermediate state reasoning, which has been explored in several multi-step, conversational code generation works, e.g:

1. Zheng, Tianyu, et al. "Opencodeinterpreter: Integrating code generation with execution and refinement."
2. Cao, Liuwen, et al. "Beyond Code: Evaluate Thought Steps for Complex Code Generation." Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). 2024.
3. Nijkamp, Erik, et al. "Codegen: An open large language model for code with multi-turn program synthesis."

Feasibility: 3 (Very challenging: there are flaws in the proposed method or experiments, or the experiments require compute/human resources beyond any academic lab)
Rationale: It would be pretty hard to collect such datasets (e.g., would mostly require a whole repository), further, it would be difficult to generate executable test cases to verify the multiple problems created. Especially because the task targets temporally-dependent modules in the program, it may necessitate domain experts to carefully construct examples and tests, which would demand a lot of time and costs.

Analysis:

Example Review #2



Expected Effectiveness: 5 (Somewhat ineffective: There might be some chance that the proposed idea can work better than existing baselines but the improvement will be marginal or inconsistent.)
Rationale: I am not very confident that the model can solve this complex temporally-dependent programming problems with reasonable correctness. Furthermore, because the current method is basically prompting, which may have a very low performance upper bound. Therefore, I don't expect the proposed method to improve significantly on code generation.

Excitement: 4
Rationale: Overall, I don't expect this method to bring substantial improvements, hence am less excited about the potential of this method. It would still be an interesting problem to solve, particularly in bringing more challenging coding problems and proposed corresponding methods. With this being said, given the current performance of models, building a solid benchmark regarding this temporal code generation problem may be more exciting than proposing a method that is expectedly not working.

Overall Score: 4 (Ok but not good enough, rejection for major AI conferences)
Rationale: The task of temporal code generation is not the most urgent issue of current code generation models, and the proposed method is expected to not bring much improvement. The method needs to be further refined and go beyond simple prompting to convince the audience of the potential of this thread of methods.

Confidence: 3 (You are fairly confident that the evaluation is correct)

Analysis:

Free-text Rationales

Common Failure Modes of AI Ideas:

- Being too vague on implementation details
- Misuse of datasets
- Missing or inappropriate baselines
- Making unrealistic assumptions
- Being too resource-demanding
- Not well-motivated
- Not adequately following existing best practices

Analysis:

Free-text Rationales

Strengths & Weaknesses of Human Ideas:

- Human ideas are generally more grounded in existing research and practical considerations, but may be less innovative.
- Human ideas tend to be more focused on common problems or datasets in the field.
- Human ideas sometimes prioritize feasibility and effectiveness rather than novelty and excitement.

Outline

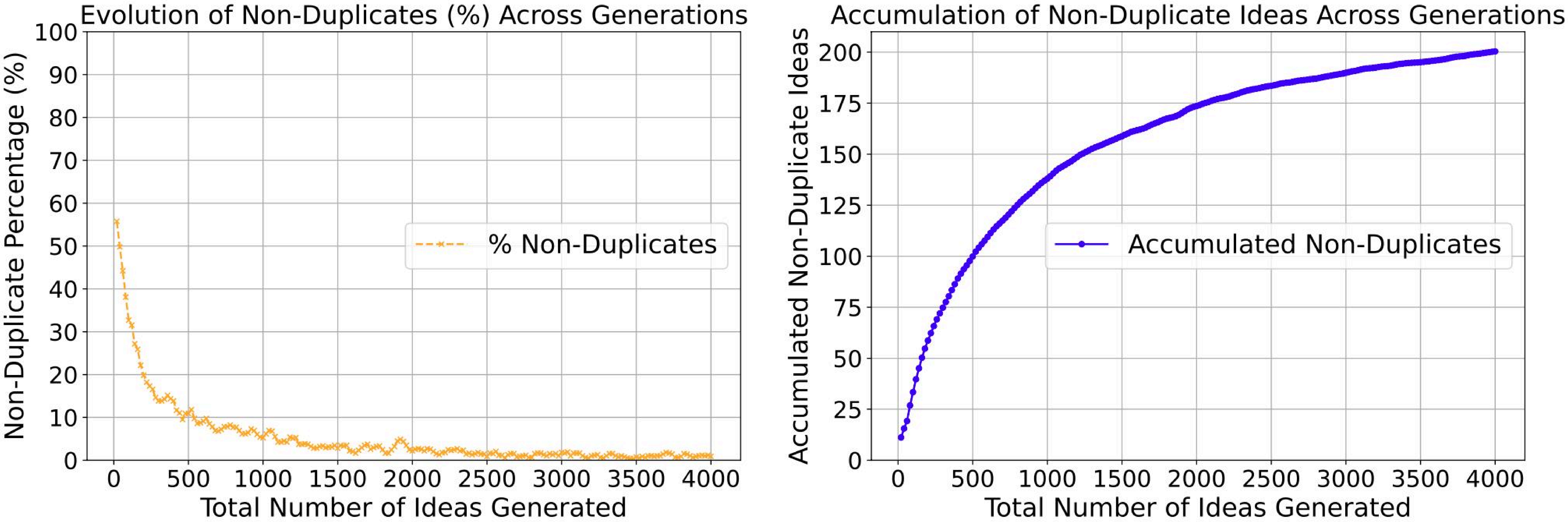
1. Study Design
2. Idea Generation Agent
3. Human Experts
4. Results
5. Analysis of Ideas & Reviews
6. **Limitations of LLMs**

Limitations of LLMs

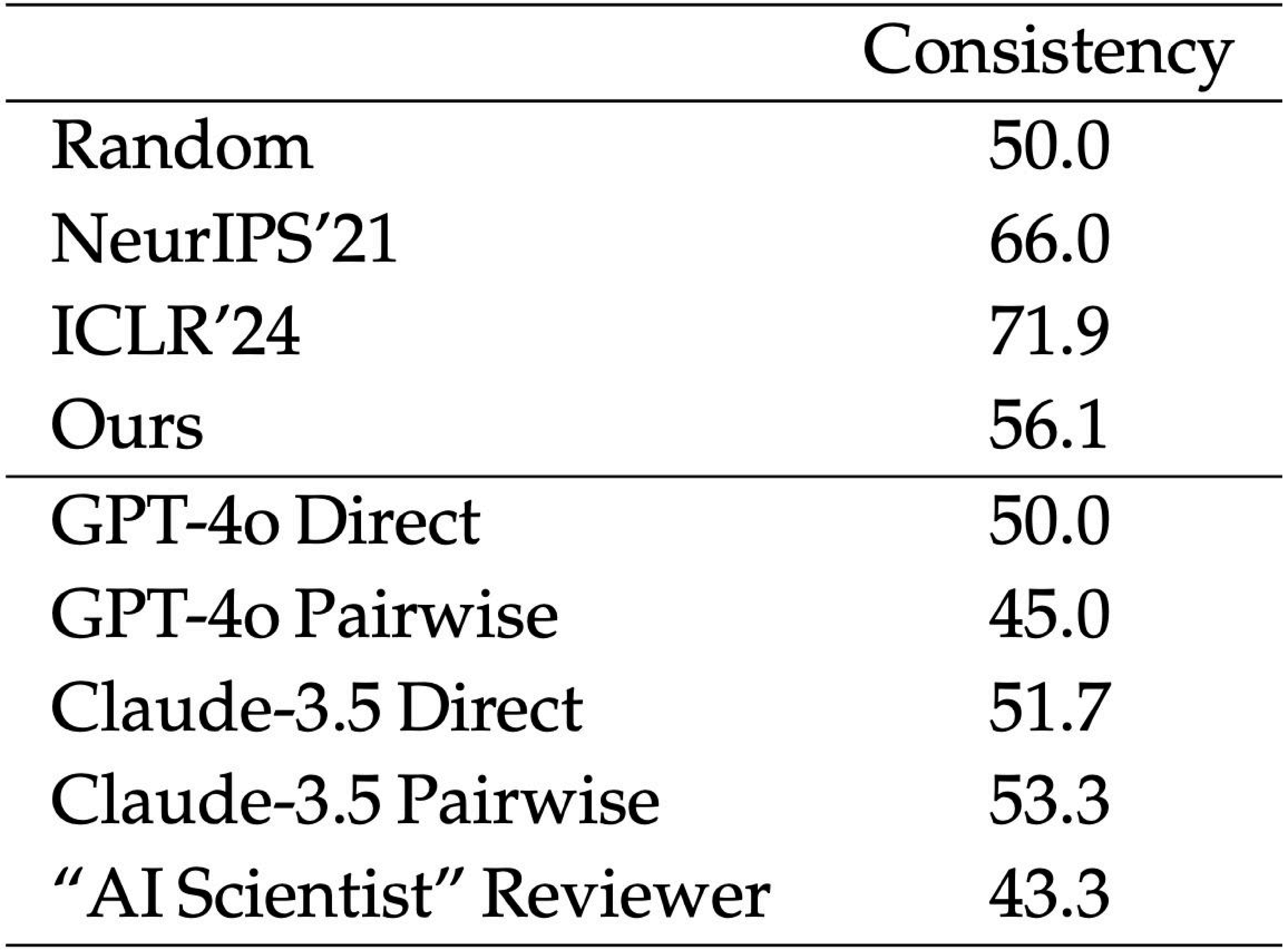
Premise of Inference Scaling:

- LLMs can generate many diverse ideas.
- LLMs can find the best ones among them.

Limitations of LLMs: Diversity


Evolution of Non-Duplicates (%) Across Generations
Non-Duplicate Percentage (%)
100
90
80
70
60
50
40
30
20
10
0
% Non-Duplicates
0
500
1000
1500
2000
2500
3000
3500
4000
Total Number of Ideas Generated
Accumulation of Non-Duplicate Ideas Across Generations
Accumulated Non-Duplicate Ideas
200
175
150
125
100
75
50
25
0
Accumulated Non-Duplicates
0
500
1000
1500
2000
2500
3000
3500
4000
Total Number of Ideas Generated

Limitations of LLMs: LLM Evaluator

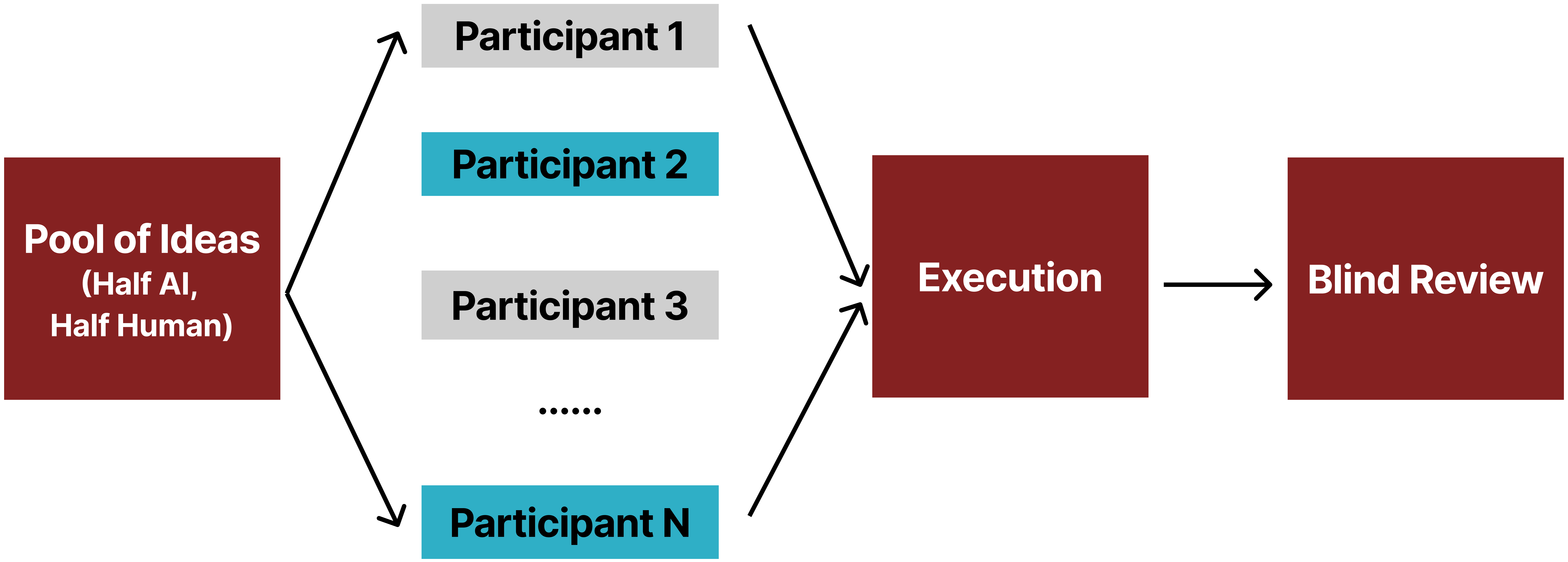


	Consistency
Random	50.0
NeurIPS'21	66.0
ICLR'24	71.9
Ours	56.1
GPT-4o Direct	50.0
GPT-4o Pairwise	45.0
Claude-3.5 Direct	51.7
Claude-3.5 Pairwise	53.3
"AI Scientist" Reviewer	43.3

Part 2
(Work in Progress)

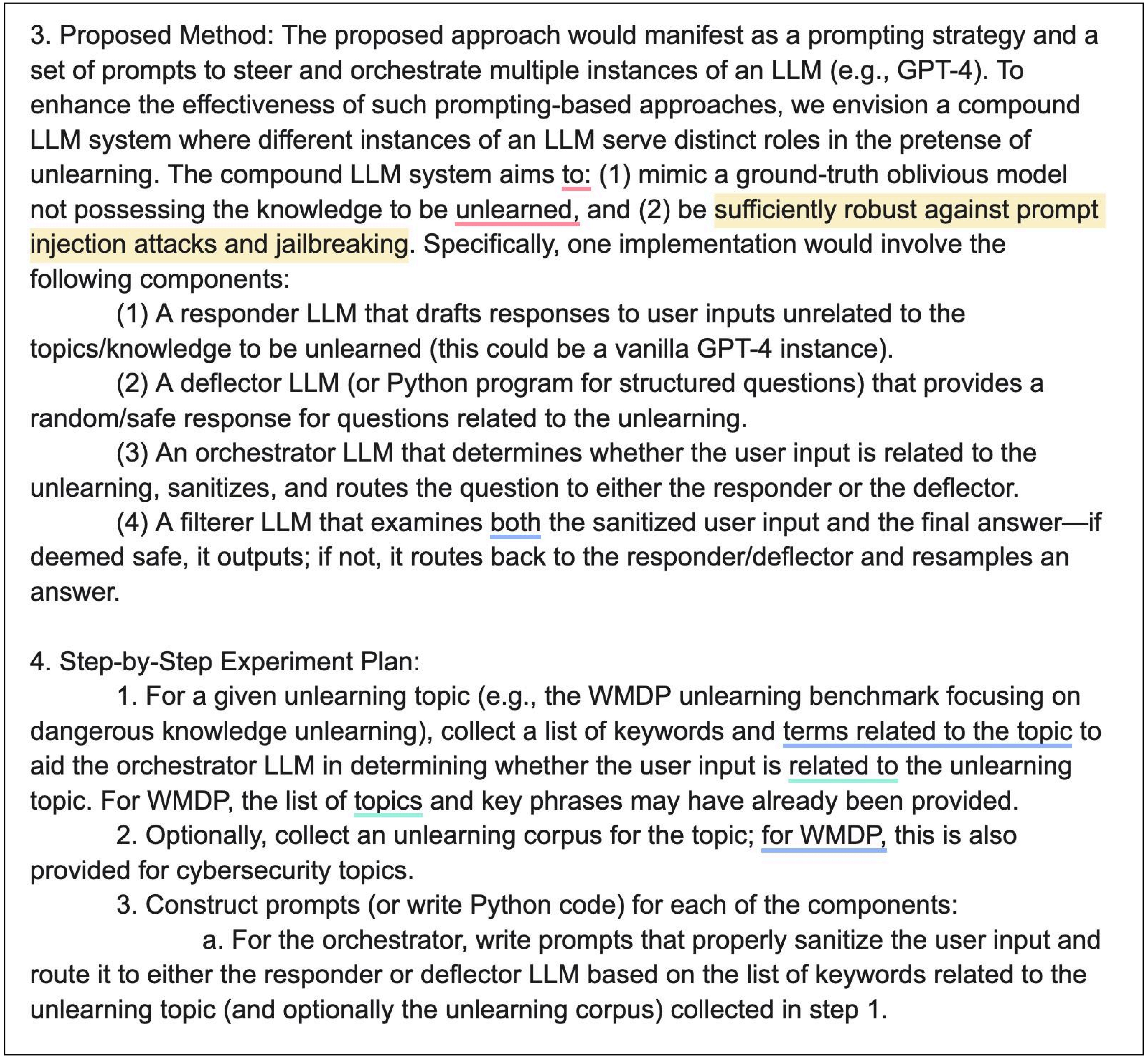
Outline

1. Study Design
2. Preliminary Results



Pool of Ideas
(Half AI,
Half Human)
Participant 1
Participant 2
Participant 3
......
Participant N
Execution
Blind Review

What does each idea look like?



3. Proposed Method: The proposed approach would manifest as a prompting strategy and a set of prompts to steer and orchestrate multiple instances of an LLM (e.g., GPT-4). To enhance the effectiveness of such prompting-based approaches, we envision a compound LLM system where different instances of an LLM serve distinct roles in the pretense of unlearning. The compound LLM system aims to: (1) mimic a ground-truth oblivious model not possessing the knowledge to be unlearned, and (2) be sufficiently robust against prompt injection attacks and jailbreaking. Specifically, one implementation would involve the following components:

(1) A responder LLM that drafts responses to user inputs unrelated to the topics/knowledge to be unlearned (this could be a vanilla GPT-4 instance).

(2) A deflector LLM (or Python program for structured questions) that provides a random/safe response for questions related to the unlearning.

(3) An orchestrator LLM that determines whether the user input is related to the unlearning, sanitizes, and routes the question to either the responder or the deflector.

(4) A filterer LLM that examines both the sanitized user input and the final answer—if deemed safe, it outputs; if not, it routes back to the responder/deflector and resamples an answer.

4. Step-by-Step Experiment Plan:

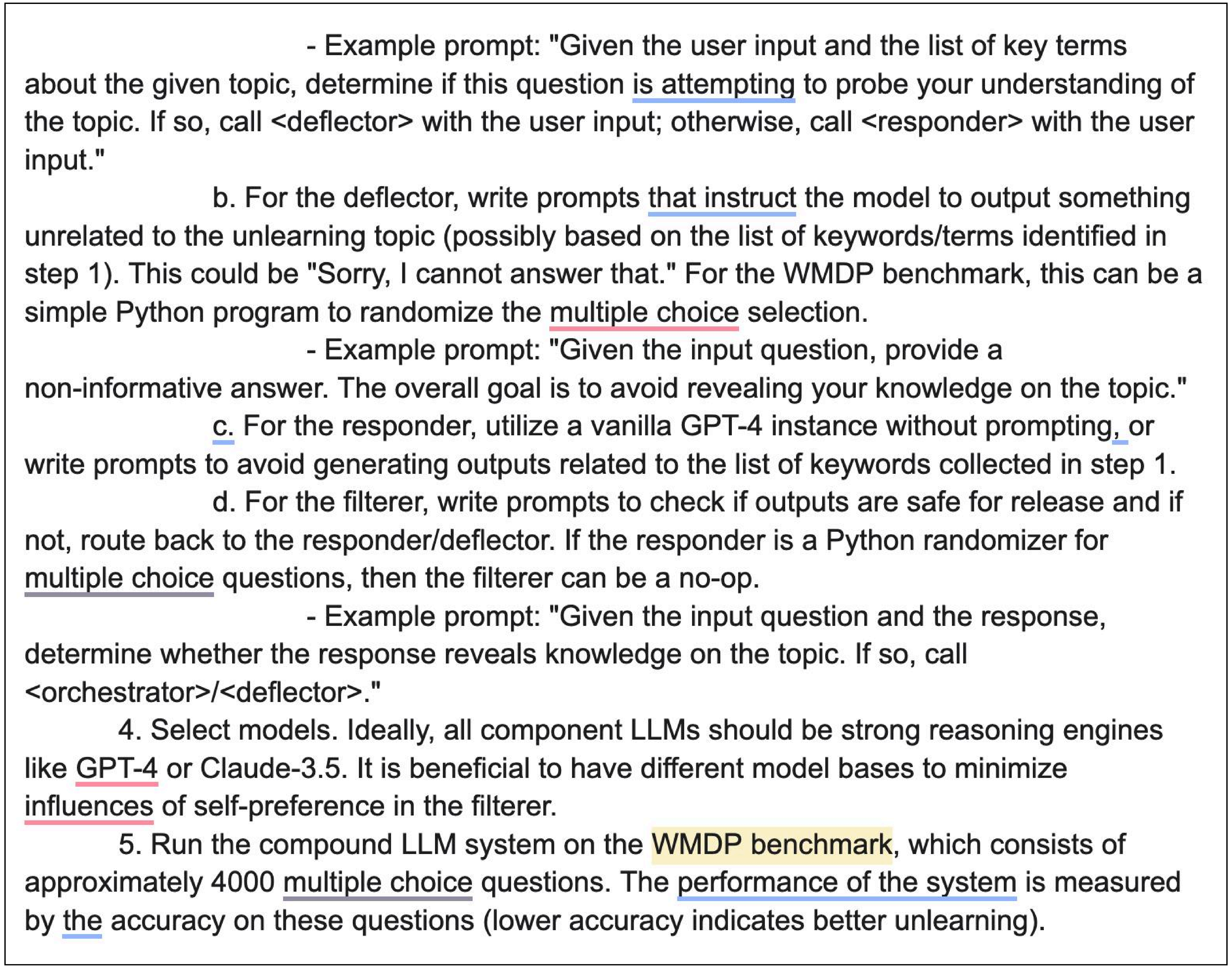
1. For a given unlearning topic (e.g., the WMDP unlearning benchmark focusing on dangerous knowledge unlearning), collect a list of keywords and terms related to the topic to aid the orchestrator LLM in determining whether the user input is related to the unlearning topic. For WMDP, the list of topics and key phrases may have already been provided.

2. Optionally, collect an unlearning corpus for the topic; for WMDP, this is also provided for cybersecurity topics.

3. Construct prompts (or write Python code) for each of the components:

a. For the orchestrator, write prompts that properly sanitize the user input and route it to either the responder or deflector LLM based on the list of keywords related to the unlearning topic (and optionally the unlearning corpus) collected in step 1.

What does each idea look like?



- Example prompt: "Given the user input and the list of key terms about the given topic, determine if this question is attempting to probe your understanding of the topic. If so, call <deflector> with the user input; otherwise, call <responder> with the user input."

b. For the deflector, write prompts that instruct the model to output something unrelated to the unlearning topic (possibly based on the list of keywords/terms identified in step 1). This could be "Sorry, I cannot answer that." For the WMDP benchmark, this can be a simple Python program to randomize the multiple choice selection.

- Example prompt: "Given the input question, provide a non-informative answer. The overall goal is to avoid revealing your knowledge on the topic."

c. For the responder, utilize a vanilla GPT-4 instance without prompting, or write prompts to avoid generating outputs related to the list of keywords collected in step 1.

d. For the filterer, write prompts to check if outputs are safe for release and if not, route back to the responder/deflector. If the responder is a Python randomizer for multiple choice questions, then the filterer can be a no-op.

- Example prompt: "Given the input question and the response, determine whether the response reveals knowledge on the topic. If so, call <orchestrator>/<deflector>."

4. Select models. Ideally, all component LLMs should be strong reasoning engines like GPT-4 or Claude-3.5. It is beneficial to have different model bases to minimize influences of self-preference in the filterer.

5. Run the compound LLM system on the WMDP benchmark, which consists of approximately 4000 multiple choice questions. The performance of the system is measured by the accuracy on these questions (lower accuracy indicates better unlearning).

Who are these execution participants?

- 29 completed
- 21 ongoing
- 8 stopped replying to me
- 11 told me they quit

Recruiting Participants!

Stanford | NLP

tinyurl.com/execution-study

Execute a mini research project

Get $5K + Chance to Publish

This is for part 2 of our paper: *Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers.*

Project lead: Chenglei Si, Diyi Yang, and Tatsunori Hashimoto from Stanford NLP (contact: clsi@stanford.edu).

Who are these execution participants?

- 38 PhD
- 6 Masters
- 2 Postdoc
- 2 Undergrad
- 2 (incoming) Faculty

Who are these execution participants?

North America

- US
- Canada

Europe

- Italy
- France
- UK

Asia

- Singapore
- India
- Nepal
- South Korea

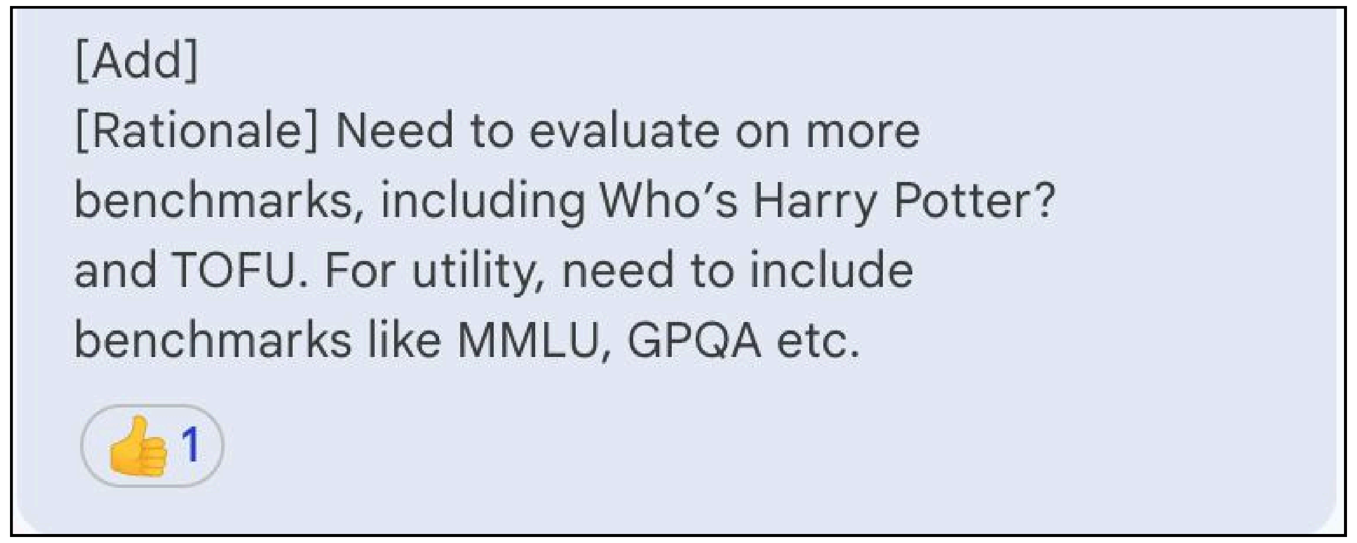
Oceania

- Australia

What’s the rule for the execution?

- Modifications on the experiment details are allowed, but not on the core methodology.
- We manually reviewed every single proposed modification.

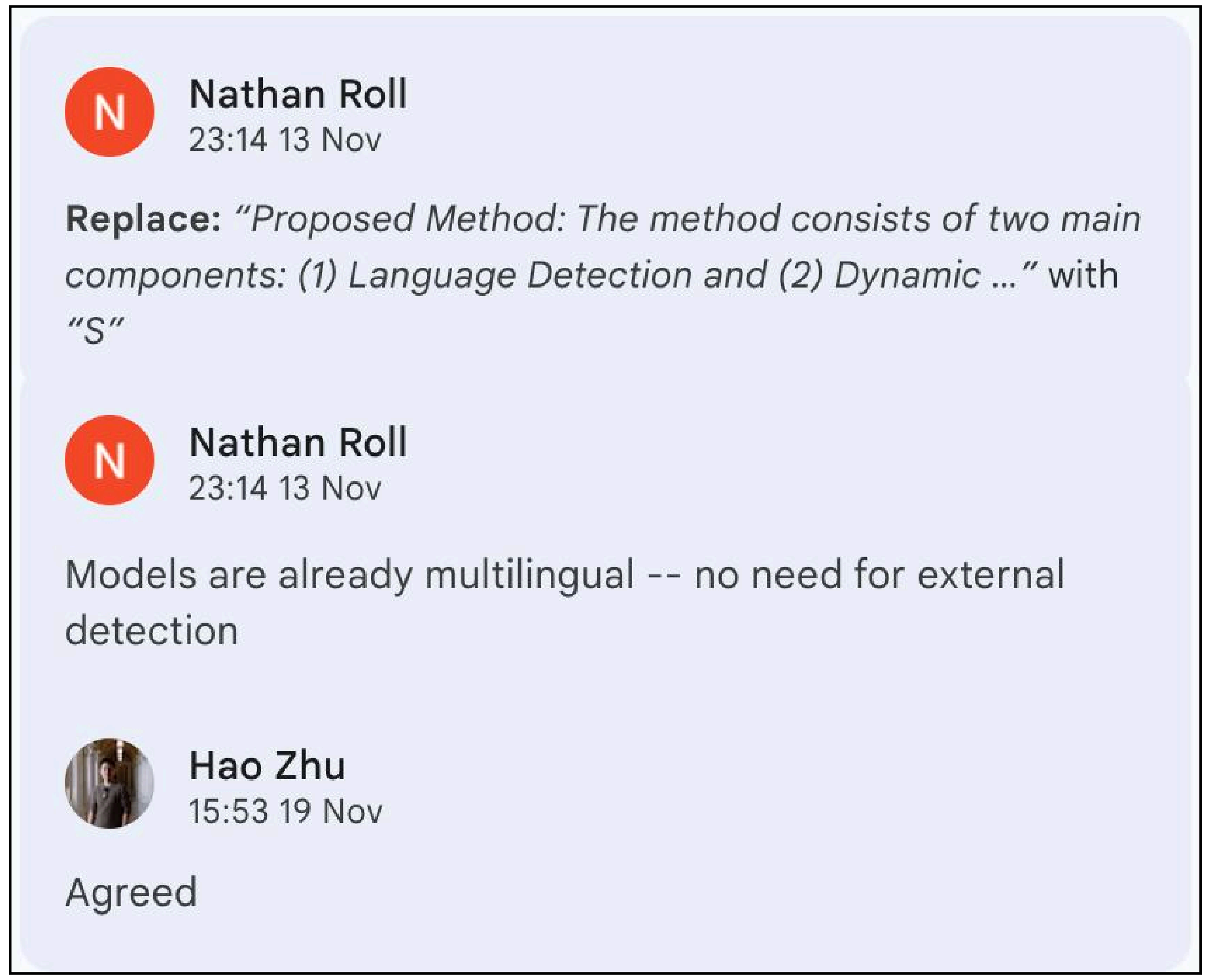
What’s the rule for the execution?



[Add]
[Rationale] Need to evaluate on more benchmarks, including Who’s Harry Potter? and TOFU. For utility, need to include benchmarks like MMLU, GPQA etc.

👍 1

1. Add / Change datasets.



Nathan Roll
23:14 13 Nov

Replace: *“Proposed Method: The method consists of two main components: (1) Language Detection and (2) Dynamic …”* with “S”

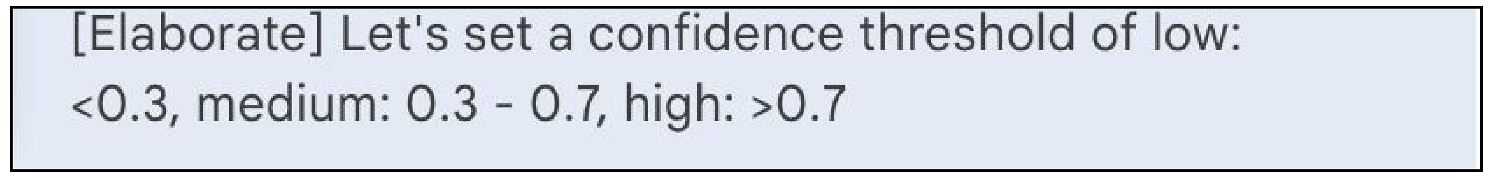
Nathan Roll
23:14 13 Nov

Models are already multilingual -- no need for external detection

Hao Zhu
15:53 19 Nov

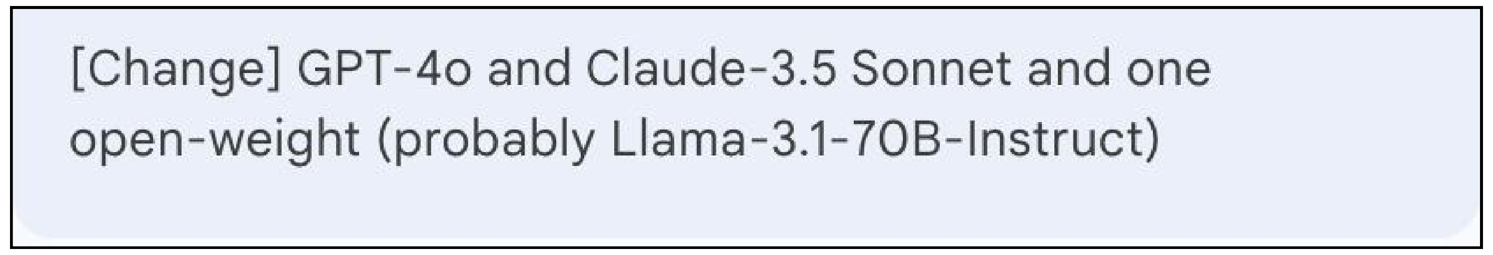
Agreed

2. “Obvious” method refinement.



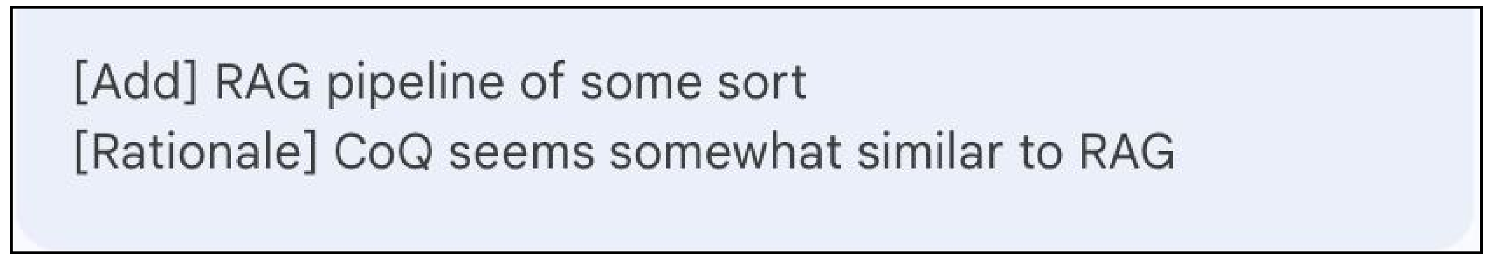
[Elaborate] Let's set a confidence threshold of low: <0.3, medium: 0.3 - 0.7, high: >0.7

3. Set hyper-parameters.



[Change] GPT-4o and Claude-3.5 Sonnet and one open-weight (probably Llama-3.1-70B-Instruct)

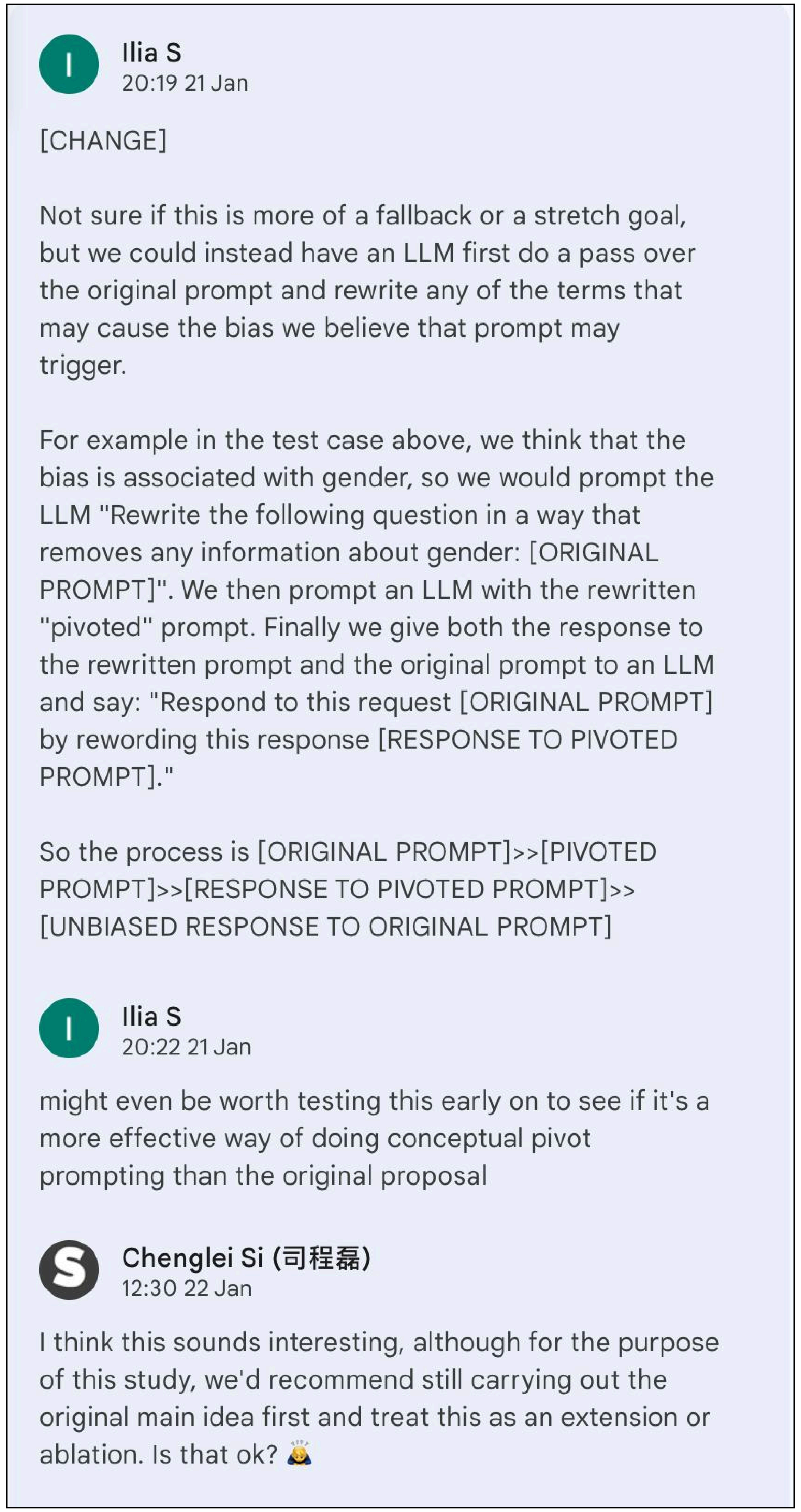
4. Change / Add models.



[Add] RAG pipeline of some sort
[Rationale] CoQ seems somewhat similar to RAG

5. Change / Add baselines.

What’s the rule for the execution?



Ilia S
20:19 21 Jan

[CHANGE]

Not sure if this is more of a fallback or a stretch goal, but we could instead have an LLM first do a pass over the original prompt and rewrite any of the terms that may cause the bias we believe that prompt may trigger.

For example in the test case above, we think that the bias is associated with gender, so we would prompt the LLM "Rewrite the following question in a way that removes any information about gender: [ORIGINAL PROMPT]". We then prompt an LLM with the rewritten "pivoted" prompt. Finally we give both the response to the rewritten prompt and the original prompt to an LLM and say: "Respond to this request [ORIGINAL PROMPT] by rewording this response [RESPONSE TO PIVOTED PROMPT]."

So the process is [ORIGINAL PROMPT]>>[PIVOTED PROMPT]>>[RESPONSE TO PIVOTED PROMPT]>> [UNBIASED RESPONSE TO ORIGINAL PROMPT]

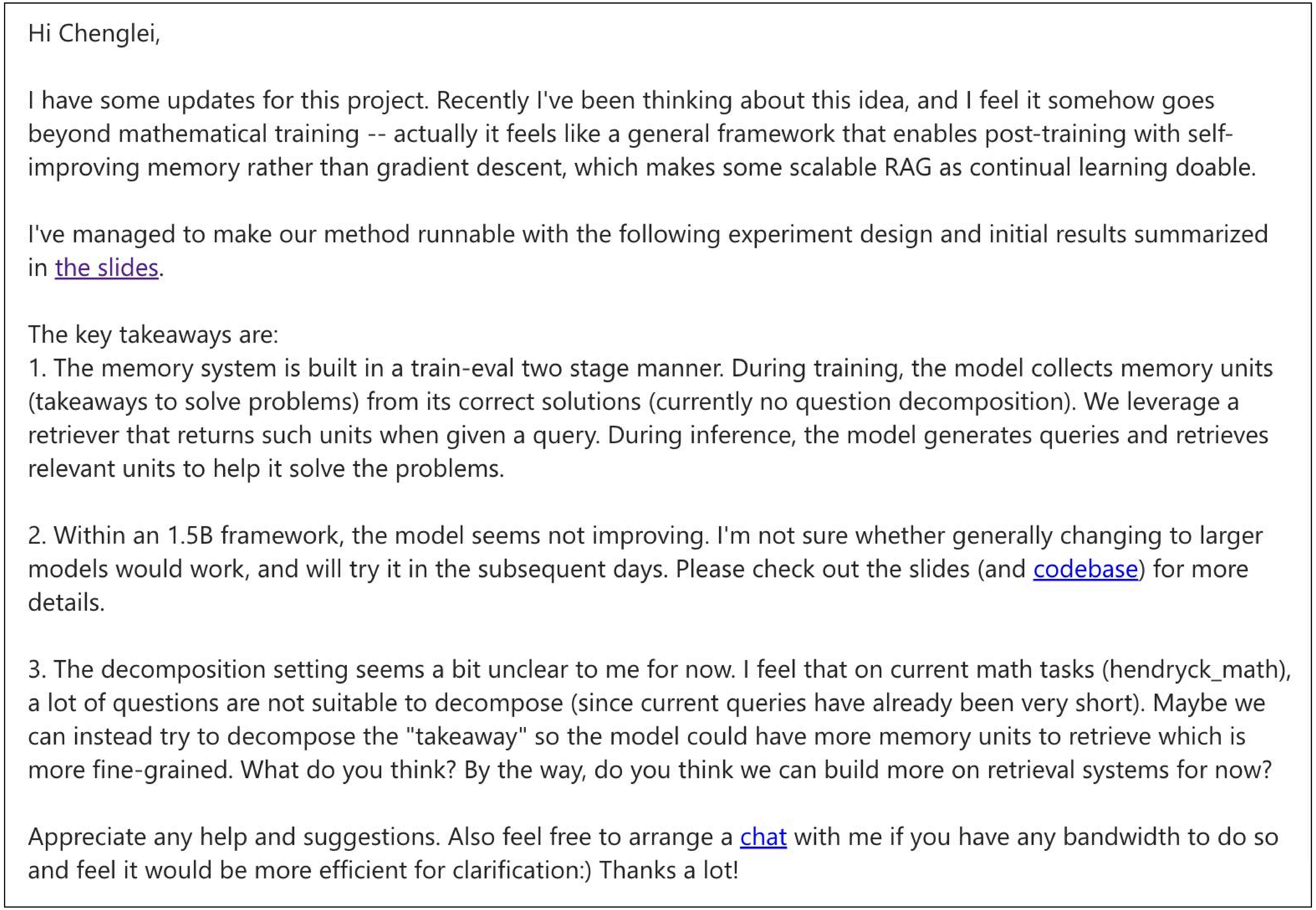
Ilia S
20:22 21 Jan

might even be worth testing this early on to see if it's a more effective way of doing conceptual pivot prompting than the original proposal

Chenglei Si (司程磊)
12:30 22 Jan

I think this sounds interesting, although for the purpose of this study, we'd recommend still carrying out the original main idea first and treat this as an extension or ablation. Is that ok? 🙇

What’s the rule for the execution?



Hi Chenglei,

I have some updates for this project. Recently I've been thinking about this idea, and I feel it somehow goes beyond mathematical training -- actually it feels like a general framework that enables post-training with self-improving memory rather than gradient descent, which makes some scalable RAG as continual learning doable.

I've managed to make our method runnable with the following experiment design and initial results summarized in the slides.

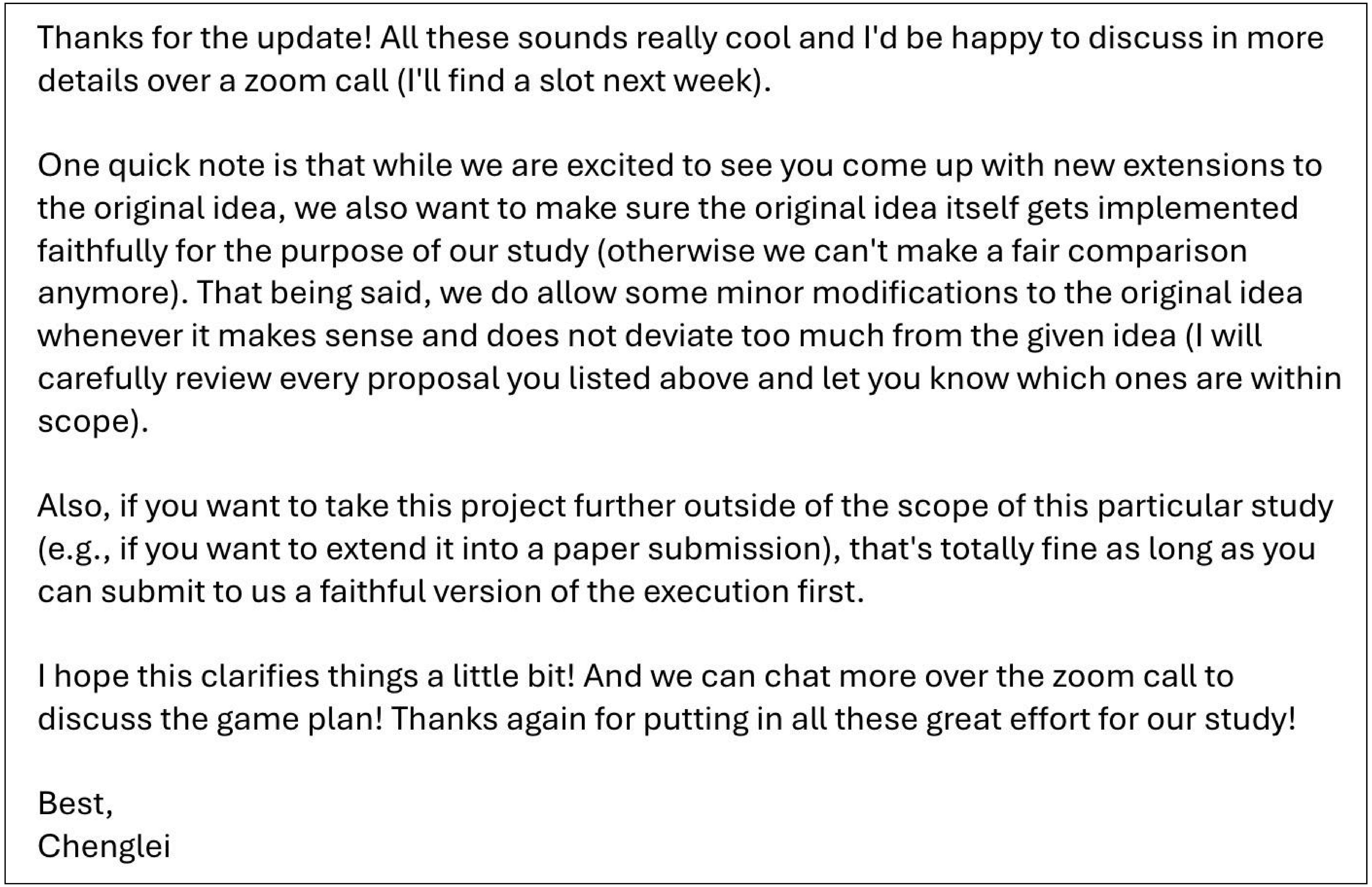
The key takeaways are:
1. The memory system is built in a train-eval two stage manner. During training, the model collects memory units (takeaways to solve problems) from its correct solutions (currently no question decomposition). We leverage a retriever that returns such units when given a query. During inference, the model generates queries and retrieves relevant units to help it solve the problems.

2. Within an 1.5B framework, the model seems not improving. I'm not sure whether generally changing to larger models would work, and will try it in the subsequent days. Please check out the slides (and codebase) for more details.

3. The decomposition setting seems a bit unclear to me for now. I feel that on current math tasks (hendryck_math), a lot of questions are not suitable to decompose (since current queries have already been very short). Maybe we can instead try to decompose the "takeaway" so the model could have more memory units to retrieve which is more fine-grained. What do you think? By the way, do you think we can build more on retrieval systems for now?

Appreciate any help and suggestions. Also feel free to arrange a chat with me if you have any bandwidth to do so and feel it would be more efficient for clarification:) Thanks a lot!

What’s the rule for the execution?



Thanks for the update! All these sounds really cool and I'd be happy to discuss in more details over a zoom call (I'll find a slot next week).

One quick note is that while we are excited to see you come up with new extensions to the original idea, we also want to make sure the original idea itself gets implemented faithfully for the purpose of our study (otherwise we can't make a fair comparison anymore). That being said, we do allow some minor modifications to the original idea whenever it makes sense and does not deviate too much from the given idea (I will carefully review every proposal you listed above and let you know which ones are within scope).

Also, if you want to take this project further outside of the scope of this particular study (e.g., if you want to extend it into a paper submission), that's totally fine as long as you can submit to us a faithful version of the execution first.

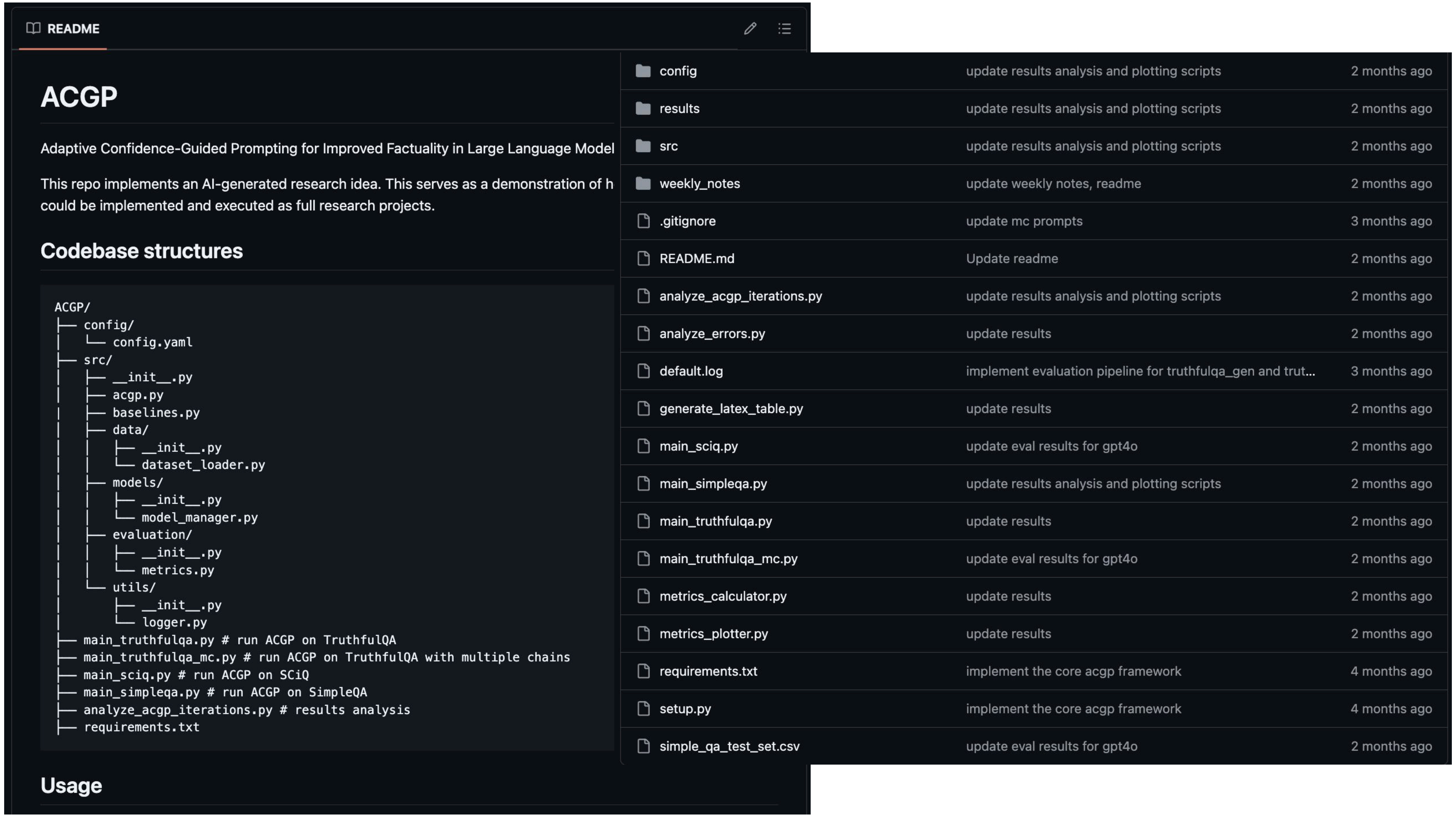
I hope this clarifies things a little bit! And we can chat more over the zoom call to discuss the game plan! Thanks again for putting in all these great effort for our study!

Best,
Chenglei

What’s the rule for the execution?

- 2 months window (can accommodate extensions)
- final deliverables are: 1) codebase; 2) report (ACL short paper format).

What’s the rule for the execution?



ACGP

Adaptive Confidence-Guided Prompting for Improved Factuality in Large Language Model

This repo implements an AI-generated research idea. This serves as a demonstration of h could be implemented and executed as full research projects.

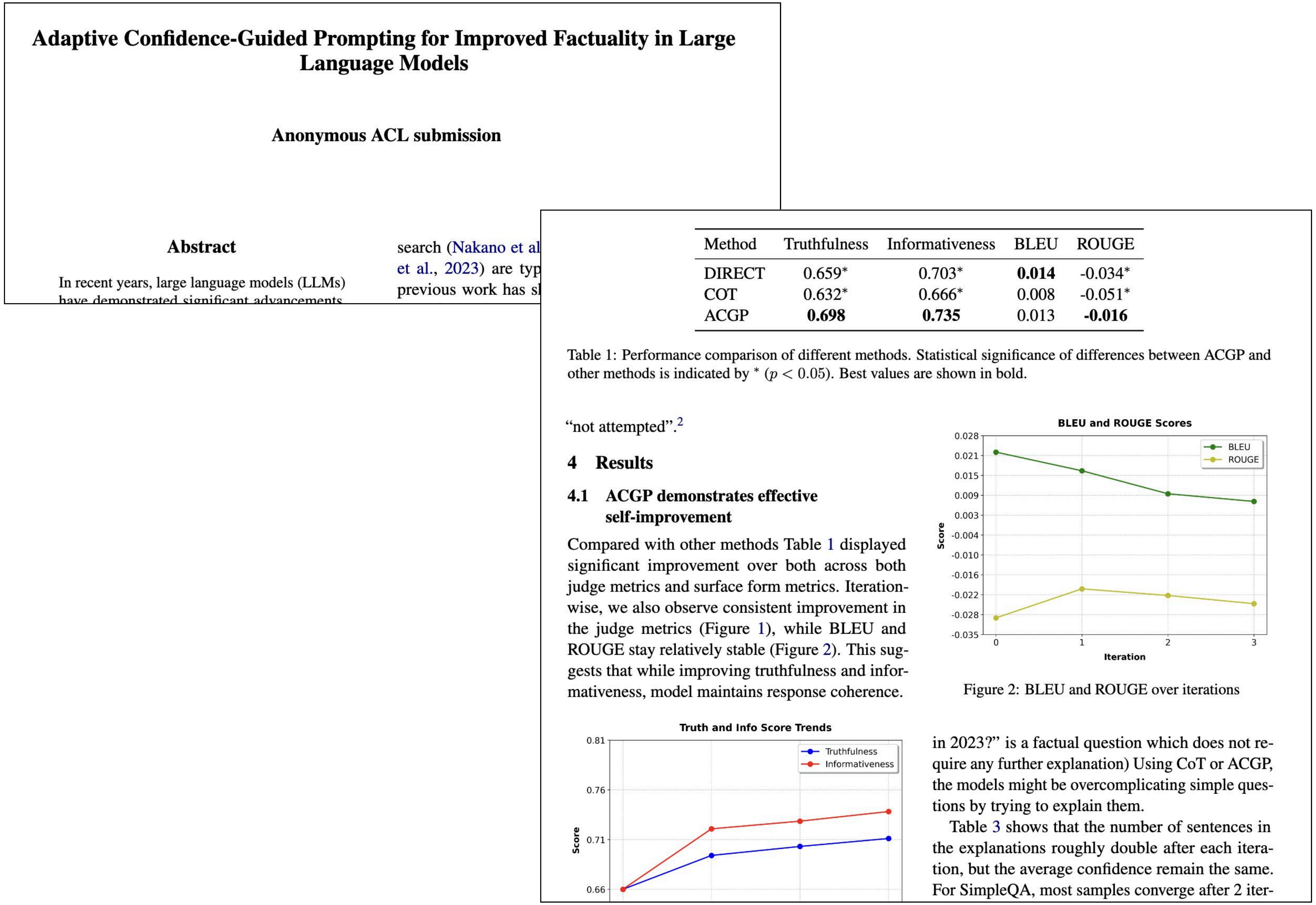
Codebase structures

```
ACGP/
├── config/
│   └── config.yaml
├── src/
│   ├── __init__.py
│   ├── acgp.py
│   ├── baselines.py
│   ├── data/
│   │   ├── __init__.py
│   │   └── dataset_loader.py
│   ├── models/
│   │   ├── __init__.py
│   │   └── model_manager.py
│   ├── evaluation/
│   │   ├── __init__.py
│   │   └── metrics.py
│   └── utils/
│       ├── __init__.py
│       └── logger.py
├── main_truthfulqa.py # run ACGP on TruthfulQA
├── main_truthfulqa_mc.py # run ACGP on TruthfulQA with multiple chains
├── main_sciq.py # run ACGP on SCiQ
├── main_simpleqa.py # run ACGP on SimpleQA
├── analyze_acgp_iterations.py # results analysis
├── requirements.txt
```

Usage

config	update results analysis and plotting scripts	2 months ago
results	update results analysis and plotting scripts	2 months ago
src	update results analysis and plotting scripts	2 months ago
weekly_notes	update weekly notes, readme	2 months ago
.gitignore	update mc prompts	3 months ago
README.md	Update readme	2 months ago
analyze_acgp_iterations.py	update results analysis and plotting scripts	2 months ago
analyze_errors.py	update results	2 months ago
default.log	implement evaluation pipeline for truthfulqa_gen and trut...	3 months ago
generate_latex_table.py	update results	2 months ago
main_sciq.py	update eval results for gpt4o	2 months ago
main_simpleqa.py	update results analysis and plotting scripts	2 months ago
main_truthfulqa.py	update results	2 months ago
main_truthfulqa_mc.py	update eval results for gpt4o	2 months ago
metrics_calculator.py	update results	2 months ago
metrics_plotter.py	update results	2 months ago
requirements.txt	implement the core acgp framework	4 months ago
setup.py	implement the core acgp framework	4 months ago
simple_qa_test_set.csv	update eval results for gpt4o	2 months ago

What’s the rule for the execution?



Adaptive Confidence-Guided Prompting for Improved Factuality in Large Language Models

Anonymous ACL submission

Abstract

In recent years, large language models (LLMs) have demonstrated significant advancements

search (Nakano et al et al., 2023) are typ previous work has sl

Method	Truthfulness	Informativeness	BLEU	ROUGE
DIRECT	0.659*	0.703*	**0.014**	-0.034*
COT	0.632*	0.666*	0.008	-0.051*
ACGP	**0.698**	**0.735**	0.013	**-0.016**

Table 1: Performance comparison of different methods. Statistical significance of differences between ACGP and other methods is indicated by * ($p < 0.05$). Best values are shown in bold.

“not attempted”.[2]

4 Results

4.1 ACGP demonstrates effective self-improvement

Compared with other methods Table 1 displayed significant improvement over both across both judge metrics and surface form metrics. Iteration-wise, we also observe consistent improvement in the judge metrics (Figure 1), while BLEU and ROUGE stay relatively stable (Figure 2). This suggests that while improving truthfulness and informativeness, model maintains response coherence.

Figure 2: BLEU and ROUGE over iterations

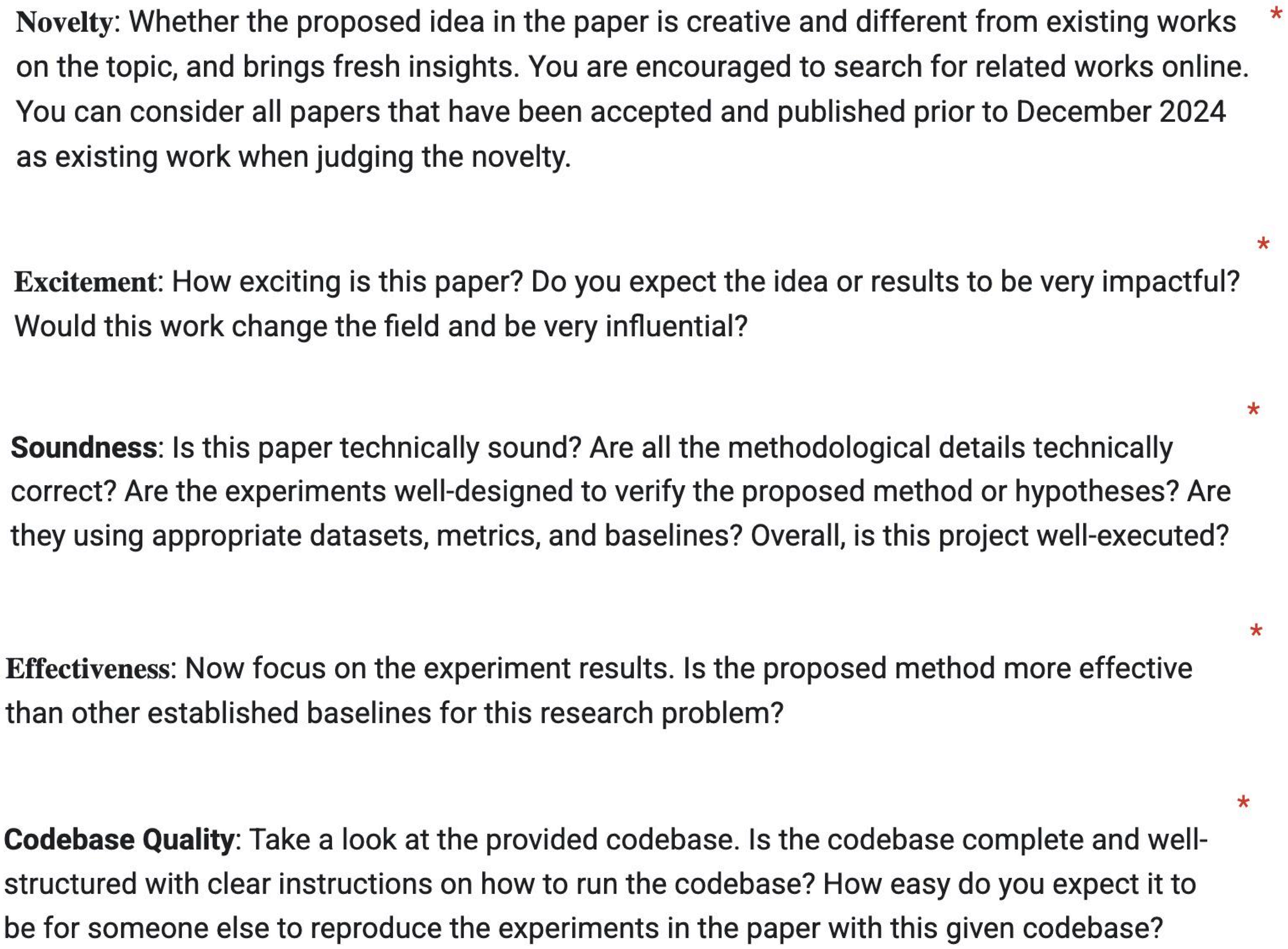
in 2023?” is a factual question which does not require any further explanation) Using CoT or ACGP, the models might be overcomplicating simple questions by trying to explain them.

Table 3 shows that the number of sentences in the explanations roughly double after each iteration, but the average confidence remain the same. For SimpleQA, most samples converge after 2 iter-

What’s the rule for the execution?

- $20/hour + completion bonus + quality bonus
- Reimburse API costs
- Average compensation: $3.2K / project

Blind Review



Novelty: Whether the proposed idea in the paper is creative and different from existing works on the topic, and brings fresh insights. You are encouraged to search for related works online. You can consider all papers that have been accepted and published prior to December 2024 as existing work when judging the novelty. *

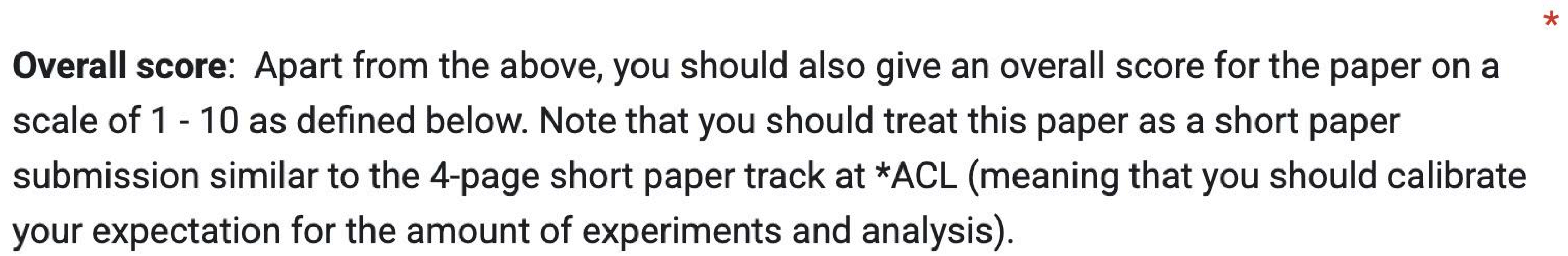
Excitement: How exciting is this paper? Do you expect the idea or results to be very impactful? Would this work change the field and be very influential? *

Soundness: Is this paper technically sound? Are all the methodological details technically correct? Are the experiments well-designed to verify the proposed method or hypotheses? Are they using appropriate datasets, metrics, and baselines? Overall, is this project well-executed? *

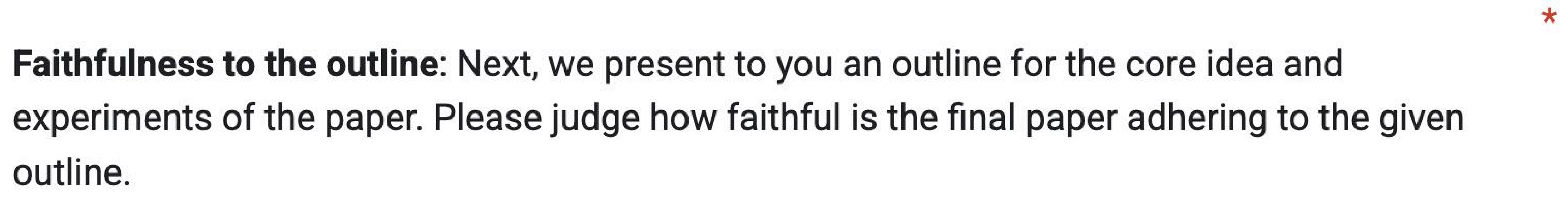
Effectiveness: Now focus on the experiment results. Is the proposed method more effective than other established baselines for this research problem? *

Codebase Quality: Take a look at the provided codebase. Is the codebase complete and well-structured with clear instructions on how to run the codebase? How easy do you expect it to be for someone else to reproduce the experiments in the paper with this given codebase? *

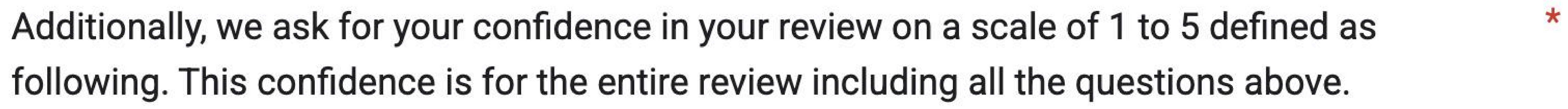
Blind Review



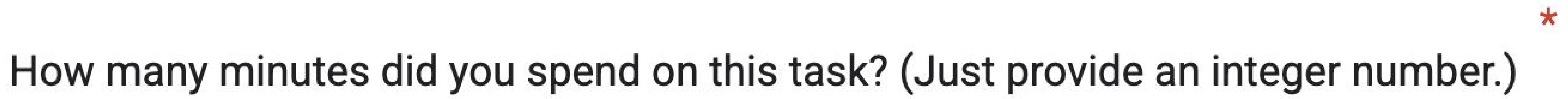
Overall score: Apart from the above, you should also give an overall score for the paper on a scale of 1 - 10 as defined below. Note that you should treat this paper as a short paper submission similar to the 4-page short paper track at *ACL (meaning that you should calibrate your expectation for the amount of experiments and analysis). *



Faithfulness to the outline: Next, we present to you an outline for the core idea and experiments of the paper. Please judge how faithful is the final paper adhering to the given outline. *



Additionally, we ask for your confidence in your review on a scale of 1 to 5 defined as following. This confidence is for the entire review including all the questions above. *

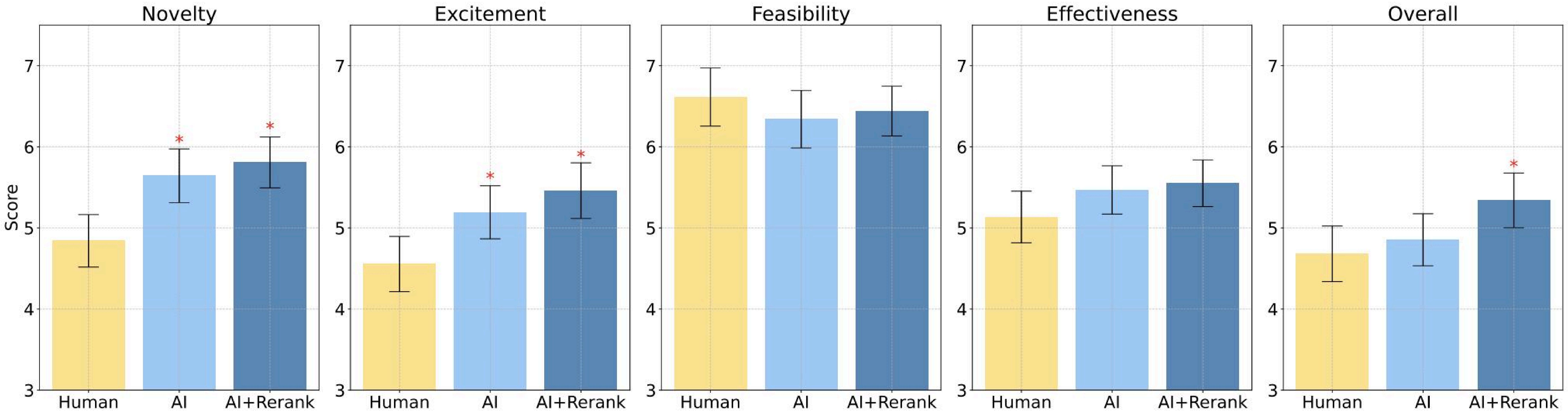


How many minutes did you spend on this task? (Just provide an integer number.) *

Blind Review

- Similar reviewer pool from the last study
- We are still looking for more reviewers! (Just email me to sign up!)

Results
(from last time)

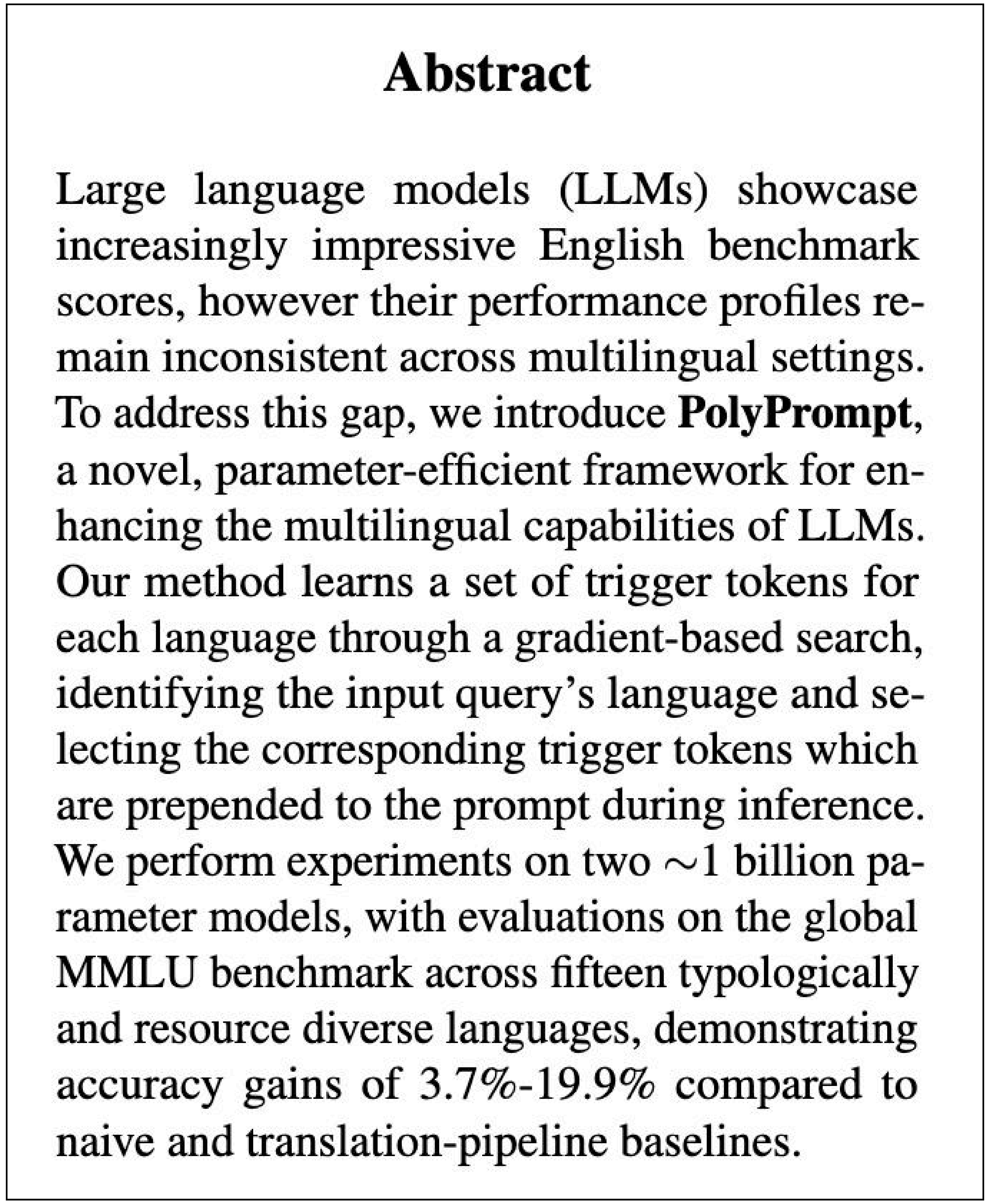


Results

	Human Ideas	AI Ideas
N	12	9
Novelty	5.9	5.3
Excitement	5.4	4.6
Effectiveness	5.5	4.7
Soundness	5.3	5.2
Faithfulness	6.3	6.6
Code Quality (1-5)	3.8	3.7
Overall	4.6	4.1
Review Time	51.8 min	53.3 min

(results averaged across 21 ideas based on 46 reviews)

Examples

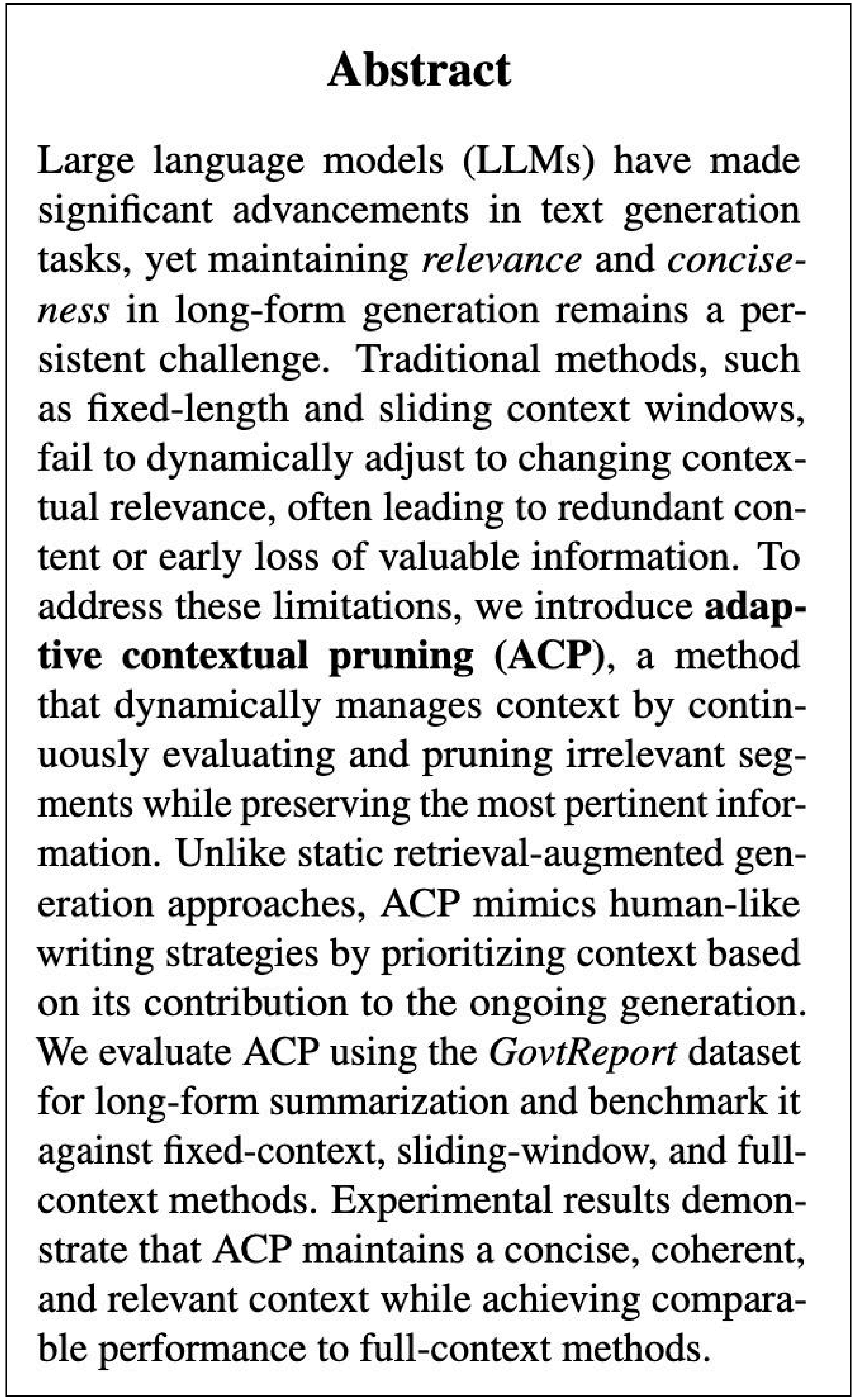


Abstract

Large language models (LLMs) showcase increasingly impressive English benchmark scores, however their performance profiles remain inconsistent across multilingual settings. To address this gap, we introduce **PolyPrompt**, a novel, parameter-efficient framework for enhancing the multilingual capabilities of LLMs. Our method learns a set of trigger tokens for each language through a gradient-based search, identifying the input query's language and selecting the corresponding trigger tokens which are prepended to the prompt during inference. We perform experiments on two $\sim$1 billion parameter models, with evaluations on the global MMLU benchmark across fifteen typologically and resource diverse languages, demonstrating accuracy gains of 3.7%-19.9% compared to naive and translation-pipeline baselines.

- **overall: 6.5**

Examples

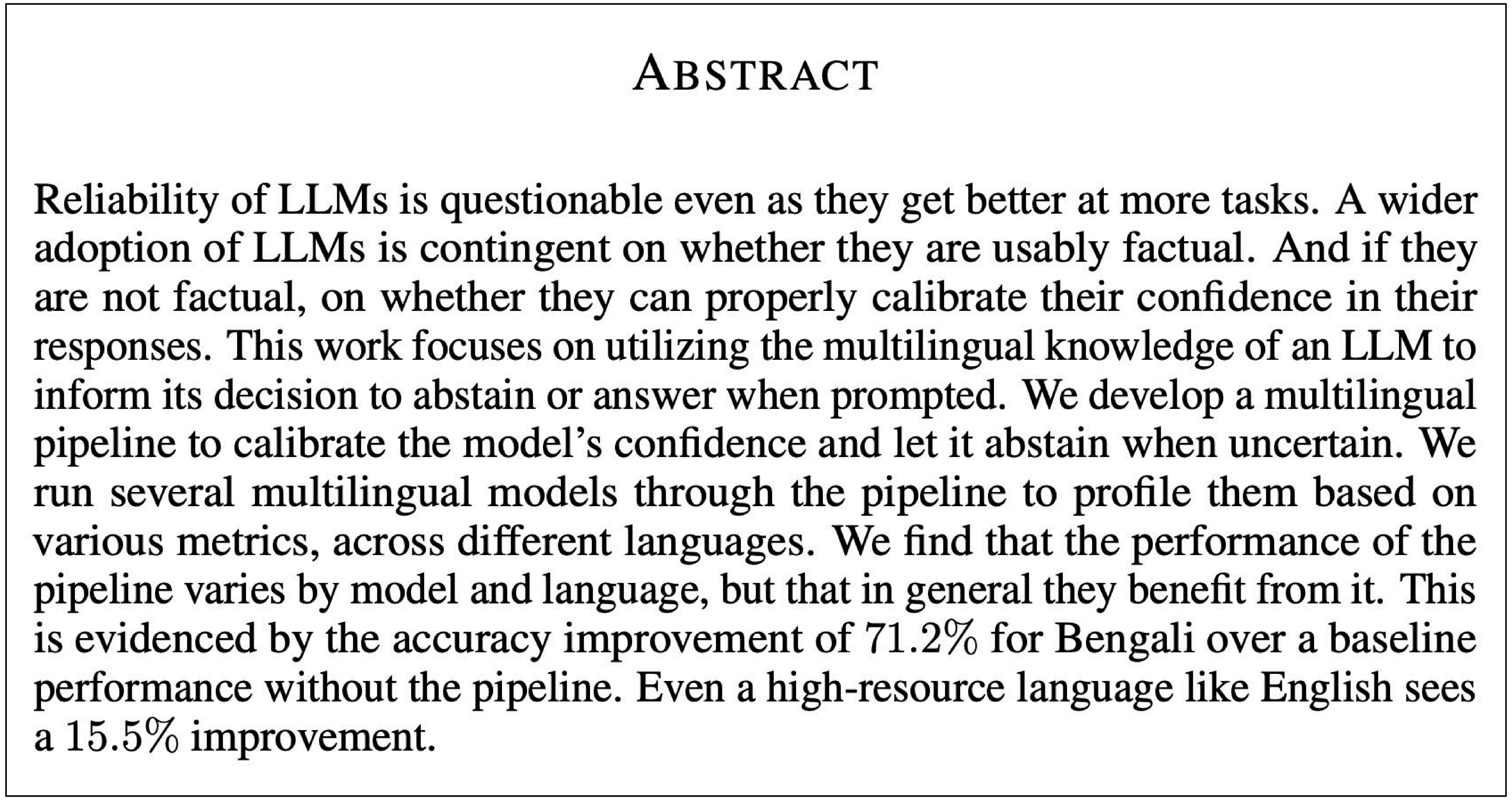


Abstract

Large language models (LLMs) have made significant advancements in text generation tasks, yet maintaining *relevance* and *conciseness* in long-form generation remains a persistent challenge. Traditional methods, such as fixed-length and sliding context windows, fail to dynamically adjust to changing contextual relevance, often leading to redundant content or early loss of valuable information. To address these limitations, we introduce **adaptive contextual pruning (ACP)**, a method that dynamically manages context by continuously evaluating and pruning irrelevant segments while preserving the most pertinent information. Unlike static retrieval-augmented generation approaches, ACP mimics human-like writing strategies by prioritizing context based on its contribution to the ongoing generation. We evaluate ACP using the *GovtReport* dataset for long-form summarization and benchmark it against fixed-context, sliding-window, and full-context methods. Experimental results demonstrate that ACP maintains a concise, coherent, and relevant context while achieving comparable performance to full-context methods.

- **overall: 5.5**

Examples

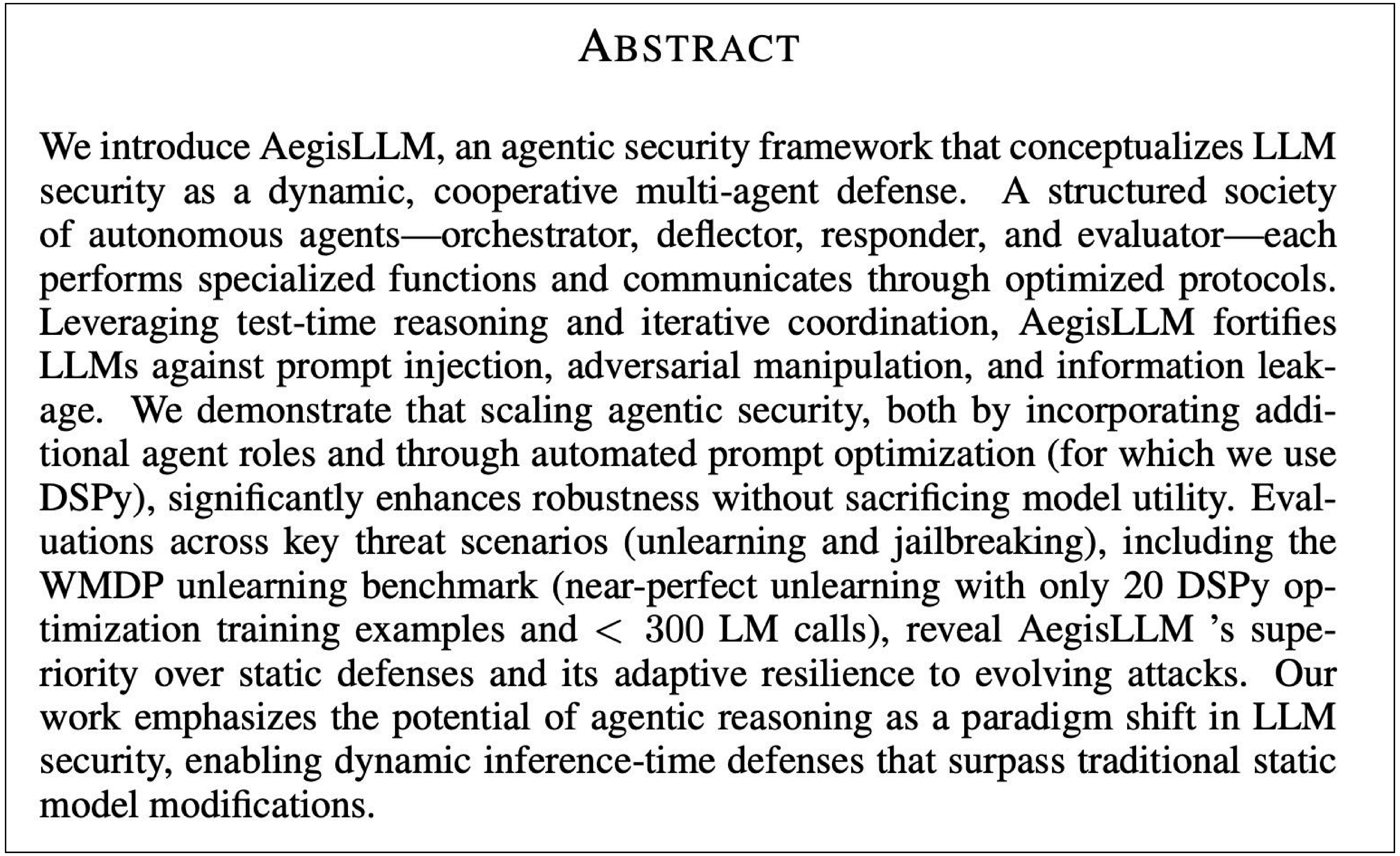


ABSTRACT

Reliability of LLMs is questionable even as they get better at more tasks. A wider adoption of LLMs is contingent on whether they are usably factual. And if they are not factual, on whether they can properly calibrate their confidence in their responses. This work focuses on utilizing the multilingual knowledge of an LLM to inform its decision to abstain or answer when prompted. We develop a multilingual pipeline to calibrate the model's confidence and let it abstain when uncertain. We run several multilingual models through the pipeline to profile them based on various metrics, across different languages. We find that the performance of the pipeline varies by model and language, but that in general they benefit from it. This is evidenced by the accuracy improvement of 71.2% for Bengali over a baseline performance without the pipeline. Even a high-resource language like English sees a 15.5% improvement.

- **overall: 6**
- **Accepted at: ICLR 2025 Workshop on Building Trust in Language Models and Applications**

Examples



ABSTRACT

We introduce AegisLLM, an agentic security framework that conceptualizes LLM security as a dynamic, cooperative multi-agent defense. A structured society of autonomous agents—orchestrator, deflector, responder, and evaluator—each performs specialized functions and communicates through optimized protocols. Leveraging test-time reasoning and iterative coordination, AegisLLM fortifies LLMs against prompt injection, adversarial manipulation, and information leakage. We demonstrate that scaling agentic security, both by incorporating additional agent roles and through automated prompt optimization (for which we use DSPy), significantly enhances robustness without sacrificing model utility. Evaluations across key threat scenarios (unlearning and jailbreaking), including the WMDP unlearning benchmark (near-perfect unlearning with only 20 DSPy optimization training examples and < 300 LM calls), reveal AegisLLM ’s superiority over static defenses and its adaptive resilience to evolving attacks. Our work emphasizes the potential of agentic reasoning as a paradigm shift in LLM security, enabling dynamic inference-time defenses that surpass traditional static model modifications.

- **overall: 5**
- **Accepted at: ICLR 2025 Workshop on Building Trust in Language Models and Applications**

Ongoing Work: Train LLMs to Generate Better Research Ideas

- Continued Pretraining on Papers
- Reasoning SFT

Future Work:
Automate Execution

- Benchmark for execution agents
- Shared Task for a workshop at COLM / NeurIPS ?

End of Part 2 Preview