

---

# Discovering Knowledge Deficiencies of Language Models on Massive Knowledge Base

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 Large language models (LLMs) possess impressive linguistic capabilities but often  
2 fail to faithfully retain factual knowledge, leading to hallucinations and unreliable  
3 outputs. Understanding LLMs’ knowledge deficiencies by exhaustively evaluating  
4 against full-scale knowledge bases is computationally prohibitive, especially for  
5 closed-weight models. We propose stochastic error ascent (SEA), a scalable and ef-  
6 ficient framework for discovering knowledge deficiencies (errors) in closed-weight  
7 LLMs under a strict query budget. Rather than naively probing all knowledge  
8 candidates, SEA formulates error discovery as a stochastic optimization process: it  
9 iteratively retrieves new high-error candidates by leveraging the semantic similarity  
10 to previously observed failures. To further enhance search efficiency and coverage,  
11 SEA employs hierarchical retrieval across document and paragraph levels, and  
12 constructs a relation directed acyclic graph to model error propagation and identify  
13 systematic failure modes. Empirically, SEA uncovers 40.7× more knowledge  
14 errors than Automated Capability Discovery and 26.7% more than AutoBench, while  
15 reducing the cost-per-error by 599× and 9×, respectively. Human evaluation  
16 confirms the high quality of generated questions, while ablation and convergence  
17 analyses validate the contribution of each component in SEA.

## 18 1 Introduction

19 Large language models (LLMs) are pretrained on vast corpora, including comprehensive knowledge  
20 sources such as Wikipedia. Despite this extensive exposure, LLMs often fail to retain or accurately  
21 reproduce factual information, resulting in misinformation and hallucinations (Gekhman et al.,  
22 2023; Zhang et al., 2023; Manakul et al., 2023; Jiang et al., 2024; Yu et al., 2024). For instance,  
23 LLM mistakes France’s capital as Berlin (Simhi et al., 2024), or may fabricate a plausible-looking  
24 citation by attributing a fictitious paper to a real researcher in the relevant domain (Merken, 2025).  
25 These knowledge deficiencies pose significant challenges for downstream applications, particularly in  
26 high-stakes domains like healthcare, law, and scientific research, where factual accuracy is paramount.

27 To this end, we propose stochastic error ascent, a scalable framework for uncovering knowledge  
28 deficiencies in LLMs under a query budget. Exhaustively querying a massive knowledge base is  
29 infeasible, so we iteratively select subsets that are likely to induce new model errors by formulating  
30 the task as a stochastic optimization problem. At each step, SEA approximates the per-step optimal  
31 subset by retrieving samples most similar to prior errors, leveraging the observation that model  
32 failures often exhibit shared characteristics (Li et al., 2024a). To enhance efficiency, we adopt a  
33 hierarchical strategy—finding the samples first at the document level and then at the paragraph  
34 level. We construct a relation directed acyclic graph (relation DAG) that traces source-target error  
35 dependencies and prunes low-impact nodes based on cumulative errors, helping discover systematic  
36 weaknesses.

37 We conduct extensive quantitative and qualitative evaluations of SEA over eight commonly used  
 38 LLMs. Compared to two baselines—Automated Capability Discovery (ACD, Lu et al. (2025)) and  
 39 AutoBench (Li et al., 2024a)—SEA uncovers 40.7× more errors than ACD and 26.7% more than  
 40 AutoBench, while reducing cost per error by 599× and 9×, respectively. Human evaluation over  
 41 1,000 randomly sampled questions confirms a 100% pass rate, i.e., all the model-generated questions  
 42 in SEA are reliable. SEA exhibits consistent error discovery across steps, with all components  
 43 contributing comparably, as shown in convergence and ablation studies.

## 44 2 Deficiency Discovery for Large Language Models

45 Consider we have a massive knowledge base  $\mathcal{K}$ , a closed-weight foundation model  $f_{\text{close}}$ , and a given  
 46 budget  $C$ .  $\mathcal{K}$  includes  $N$  documents, each with an abstract and  $M_i$  paragraphs  $p$ , i.e.,  $\mathcal{K} = \{p_j^{(i)} \mid i =$   
 47  $1, \dots, N; j = 1, \dots, M_i\}$ . **Our goal** is to find an optimal paragraph subset  $\hat{\mathcal{S}} = \{p_1, \dots, p_{|\hat{\mathcal{S}}|}\}$  from  
 48  $\mathcal{K}$  that can maximize the error  $T_{\hat{\mathcal{S}}}(f_{\text{close}})$  of the closed-weight language model  $f_{\text{close}}$  under the budget  
 49  $C$  for  $f_{\text{close}}$  (e.g., the price for all tokens or a number of API calls). Specifically, we aim to solve the  
 50 following optimization problem:

$$\arg \max_{\mathcal{S} \subset \mathcal{K}} T_{\mathcal{S}}(f_{\text{close}}) \quad \text{s.t.} \quad \sum_{|g(\mathcal{S})|} \text{cost}(f_{\text{close}}) < C, \quad (1)$$

51 where

$$T_{\mathcal{S}}(f_{\text{close}}) = \frac{1}{|g(\mathcal{S})|} \sum_{p_j^{(i)} \in \mathcal{S}} \left( \sum_{(x,y) \in g(p_j^{(i)})} \mathbb{1}_{[f_{\text{close}}(x) \neq y]} \right). \quad (2)$$

52 Eq. (2) denotes the average error of the closed-weight model  $f_{\text{close}}$ , where  $(x, y)$  is the multiple-choice  
 53 question-answer pairs generated from each  $p_j^{(i)}$  by prompting a question generator LLM  $g(\cdot)$ , and  
 54  $\mathbb{1}_{[\cdot]}$  is an indicator function. To avoid the accidental error triggered by the prompt sensitivity of  $f_{\text{close}}$ ,  
 55 we rephrase each question multiple times and generate semantically equivalent variants. To simplify,  
 56 we use  $T_{\mathcal{S}}$  for  $T_{\mathcal{S}}(f_{\text{close}})$  in the later sections.

### 57 2.1 Stochastic Error Ascent

58 To achieve our optimization objective in Eq. (1), we propose a multi-step stochastic error ascent  
 59 (SEA) algorithm. Generally, SEA iteratively performs error-based subset updates with hierarchical  
 60 error-related retrieval until we reach the budget  $C$ . During the error-related retrieval, we construct  
 61 a relation directed acyclic graph (relation DAG) with the source errors as nodes and error relations  
 62 across each timestamp as edges. We further prune the source error by tracking the quality of the  
 63 per-step source error. We present the overall algorithm in Alg. 1.

64 **Error-based subset update.** Given a large  $\mathcal{K}$ , it is nearly impossible to let  $f_{\text{close}}$  go through all  
 65 questions generated from  $p \in \mathcal{K}$  to find an optimal set of paragraphs  $\hat{\mathcal{S}}$  that can achieve the Eq.(1)  
 66 under a limited budget. Therefore, we consider updating  $\mathcal{S}$  iteratively. At timestamp  $t$ , we find a  
 67 batch of candidate paragraphs  $\mathcal{P}_t = \{p_t^{(1)}, \dots, p_t^{(|\mathcal{P}_t|)}\}$  from  $\mathcal{K}$  that can maximize the probability  
 68 of  $T_{\mathcal{S}_t \cup \mathcal{P}_t} > T_{\mathcal{S}_t}$ , which can be represented as:

$$\mathcal{S}_{t+1} = \mathcal{S}_t \cup \arg \max_{\mathcal{P}_t \subset \mathcal{K}} \Pr(T_{\mathcal{S}_t \cup \mathcal{P}_t} > T_{\mathcal{S}_t}). \quad (3)$$

69 We solve Eq.(3) by focusing on paragraphs that resemble the error-inducing examples. Specifically,  
 70 we target regions in  $\mathcal{K}$  that are likely to further challenge the model, resulting in retrieving an error-  
 71 related batch  $E$  that is semantically similar to a set of source error paragraphs:  $\mathcal{P}_{\text{source}}^{(t)} \subset \bigcup_{i=1}^t \mathcal{S}_i$ .  
 72 Here,  $\mathcal{P}_{\text{source}}^{(t)}$  comprises paragraphs for which the LLM exhibits a high error rate (i.e.,  $T_{\{p\}} > \xi$  for  
 73 all  $p \in \mathcal{P}_{\text{source}}^{(t)}$ , with  $\xi$  being a predefined error threshold). Practically, we retrieve the error-related  
 74 batch  $E$  by ranking candidates from  $\mathcal{K}$  based on their semantic similarity to  $\mathcal{P}_{\text{source}}^{(t)}$ , using tools  
 75 such as Sentence Transformers (Reimers & Gurevych, 2019), and selecting the top- $k$  candidates.

76 Given a Sentence Transformer  $f_s : p \rightarrow \mathbb{R}^d$ , where  $d$  is the embedding dimension, we can define the  
 77 error-related batch as:

$$E = \text{FindSim}(\mathcal{K}, \mathcal{P}_{\text{source}}^{(t)}) = \left\{ p_c \mid p_c \in \bigcup_{p_s \in \mathcal{P}_{\text{source}}^{(t)}} \text{Top}_k \left( \frac{f_s(p_s) \cdot f_s(\mathcal{K})}{\|f_s(p_s)\| \|f_s(\mathcal{K})\|} \right) \right\}, \quad (4)$$

78 where  $f_s(\mathcal{K}) = [f_s(p_1), \dots, f_s(p_{|\mathcal{K}|})]^\top$ . To improve the efficiency of calculating Eq.(4), we perform  
 79 hierarchical retrieval from document to paragraph levels. We first pre-process the embeddings for the  
 80 abstract of all documents  $d_i \in \mathcal{K}$  as  $\mathcal{D}_{\text{abs}}$ , where  $d_i$  contains a set of paragraphs that can be represented  
 81 as  $\{p_{1:M_i}^{(i)}\}$ . We then retrieve a set of candidate error-related documents  $\mathcal{D}_c$  by performing Eq.(4)  
 82 between  $\mathcal{D}_{\text{abs}}$  and  $\mathcal{P}_{\text{source}}$ , i.e.,  $\mathcal{D}_c = \text{FindSim}(\mathcal{D}_{\text{abs}}, \mathcal{P}_{\text{source}})$ . Finally, we retrieve error-related batch  
 83  $E$  by comparing the paragraphs in document  $\mathcal{D}_c$  and source errors, i.e.,  $E = \text{FindSim}(\mathcal{D}_c, \mathcal{P}_{\text{source}})$ .  
 84 In this way, we finish the hierarchical retrieval.

85 **Relation DAG construction and source pruning.** To identify systematic issues in  $f_{\text{close}}$ , such  
 86 as flaws localized within specific regions of the knowledge base, we construct a relation DAG  
 87  $\mathcal{G} = (\mathcal{P}_{\text{source}}^{(t)}, \mathcal{E}(\mathcal{P}_{\text{source}}^{(t)}, \mathcal{P}_{\text{source}}^{(t+1)}))$ .  $\mathcal{G}$  is constructed by linking each paragraph  $p \in \mathcal{P}_{\text{source}}^{(t)}$  to its top  
 88 semantically similar error-inducing paragraphs in  $\mathcal{P}_{\text{source}}^{(t+1)}$ , based on the hierarchical error-related  
 89 retrieval described above, forming directed edges that represent potential error propagation paths. We  
 90 then assess the quality of  $\mathcal{P}_{\text{source}}$  based on *cumulative error*. We define the *cumulative error*  $\pi_{\mathcal{G}}(p)$   
 91 for paragraph  $p \in \mathcal{P}_{\text{source}}$  as the average error across its descendants’ error:

$$\pi_{\mathcal{G}}(p) = \frac{1}{|\text{Desc}_{\mathcal{E}}(p)|} \sum_{v \in \text{Desc}_{\mathcal{E}}(p)} T_{\{v\}}, \quad (5)$$

92 where  $\text{Desc}_{\mathcal{E}}(p)$  denotes all descendants of  $p$  that can be reached via the edge space  $\mathcal{E}$ . We **prevent**  
 93 **loops** in  $\mathcal{G}$  by removing new source errors in each step from  $\mathcal{K}$  (line 11 in Alg. 1). We then perform a  
 94 threshold filter according to  $\pi_{\mathcal{G}}(p)$  at each step to prune  $\mathcal{P}_{\text{source}}$ .

### 95 3 Comparing Stochastic Error Ascent with Baselines

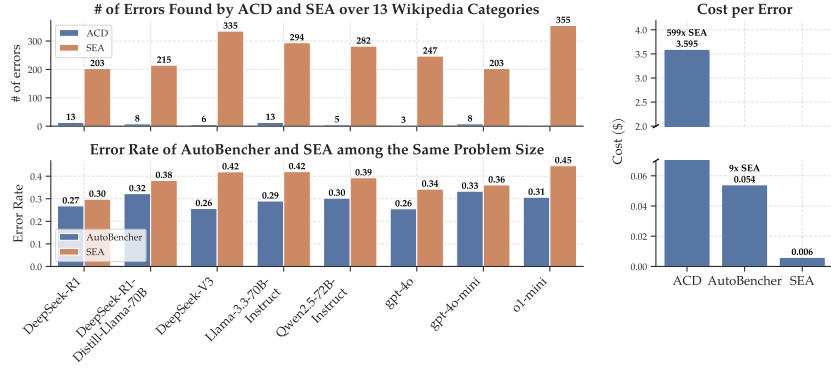
96 **Knowledge base details.** We collect a large-scale English-based knowledge base from  
 97 Wikipedia (Wikipedia contributors, 2004), comprising 7.1M documents and 28.8M paragraphs  
 98 across 13 top-level categories with hierarchical subcategories.

99 **Evaluation protocol.** We use an LLM generator (gpt-4o) to generate and rephrase questions for  
 100 each paragraph. Specifically, the LLM generator will take a paragraph as input and output a list of  
 101 questions in JSON format. We perform the human evaluation of the generated questions and put  
 102 the quality analysis in Sec. A. We compute accuracy for the questions answered by all models by  
 103 comparing the ground truth answer provided by the LLM generator and the answer from the testee  
 104 models. We formulate all the questions with the same template for all testee models.

105 **Implementation details.** For  $f_{\text{close}}$  selection, besides widely used closed-weight models, we also  
 106 simulate open-weight models in the closed-weight setting to expand the scope of evaluation. Specif-  
 107 ically, we evaluate GPT-series models: gpt-4o (OpenAI et al., 2024), gpt-4o-mini (OpenAI,  
 108 2024a), and o1-mini (OpenAI, 2024b), and DeepSeek-series models: Deepseek-V3 (Liu et al.,  
 109 2024), Deepseek-R1, and R1-Distill-Llama-70B (DeepSeek-AI et al., 2025). We also include  
 110 two widely used models hosted on DeepInfra<sup>1</sup>: Qwen2.5-72B-Instruct (Qwen et al., 2025) and  
 111 Llama-3.3-70B-Instruct (Grattafiori et al., 2024). We adopt mGTE (Zhang et al., 2024b) as our  
 112 Sentence Transformer model for hierarchical retrieval. We set the decoding parameters to temperature  
 113 0.1 and top- $p$  0.9 for deterministic responses. Thresholds for source error judgment and source error  
 114 pruning are set to  $\xi = \gamma = 0.5$ . We use the average accuracy across the 25 generated questions for  
 115 source error judgment. We set the  $k$  as 50 in Eq.(4) and randomly select 40 paragraphs from the  
 116 retrieved paragraphs as the error-related batch  $E$ . For the initial batch  $B$ , we uniformly retrieve 40  
 117 paragraphs from 13 predefined Wikipedia categories.

118 **Baselines and budget settings.** Research on automatically generating questions to self-discover  
 119 misinformation and vulnerabilities in LLMs remains limited. We select two closely related methods,

<sup>1</sup><https://deepinfra.com/>



**Figure 1:** Comparison of errors discovered by ACD, AutoBench, and SEA. We compare ACD with SEA among the same budget while comparing AutoBench among the same question size. For ACD, we summarized the number of failed tasks, and for SEA, we summarized the number of source errors. We let AutoBench create 13 benchmarks, each of which takes one of the Wikipedia categories as an interesting topic. We let SEA search the same number of questions according to each model. o1-mini failed on ACD due to the violation of the prompt usage policy from OpenAI.

120 Automated Capability Discovery (ACD, Lu et al. (2025)) and AutoBench (Li et al., 2024a), as  
 121 baselines to compare the ability of error discovery and cost. In our experiments, we use gpt-4o as  
 122 the generator model for task/benchmark generation for ACD and AutoBench. We let ACD take  
 123 one of the Wikipedia topics as seed tasks for task generation, and let AutoBench use all categories  
 124 as interest topics and generate 13 benchmarks, including 2,000 questions in total. We adopt two  
 125 different budget settings for SEA according to the features of ACD and AutoBench. Specifically,  
 126 we set the budget as 20,000 API calls for  $f_{\text{close}}$  plus the cost for QA generation when comparing with  
 127 ACD, and the budget as AutoBench to generate 13 benchmarks when comparing with AutoBench.

### 128 3.1 Results

129 **Comparing with ACD** We first compare the number of errors ACD and SEA discovered among  
 130 the same budget, including question generation and testee inference. Specifically, we let ACD  
 131 start searching from a free-style handwritten task conditioning on one of the 13 general Wikipedia  
 132 categories and summarize the error tasks discovered by ACD. Error tasks and source errors both reflect  
 133 LLM misinformation in a category. Therefore, we summarize the comparison results between error  
 134 tasks and source errors found by ACD and SEA, respectively, for each model across 13 categories in  
 135 Fig. 1. We can observe a significant gap between the number of errors found by SEA and ACD. SEA  
 136 can discover at most 55.83 times as many as the errors of ACD on DeepSeek-V3 model.

137 **Comparing with AutoBench** We compare the error rates of questions generated by SEA and  
 138 AutoBench. Specifically, we let AutoBench generate 13 benchmarks, each corresponding to a  
 139 distinct Wikipedia category as the input topic. We then concatenate all benchmarks as one (resulting  
 140 in 2,000 questions) and evaluate all models based on it. We compare the error rate between the  
 141 AutoBench benchmark and the error rate on an equal number of questions generated by SEA. The  
 142 results are summarized in Fig. 1. As shown in Fig. 1, SEA outperforms AutoBench across all  
 143 evaluated models in terms of error rate. The average error rate across all models is 0.38 for SEA  
 144 versus 0.30 for AutoBench, reflecting a 26.7% increase in error detection efficiency.

## 145 4 Conclusion

146 In this work, we introduced stochastic error ascent (SEA) that can discover knowledge deficiencies  
 147 of language models on a massive knowledge base. SEA identify knowledge deficiencies in closed-  
 148 weight LLMs by framing it as a budget-constrained stochastic optimization process. SEA surpass  
 149 previous baselines, including ACD and AutoBench, by uncovering 40.72 times and 26.7% more  
 150 errors, respectively, at 599 and 9 times lower cost per error. SEA achieves a 100% human pass rate  
 151 on generated questions, exposes distinct error clusters across models such as gpt-4o, DeepSeek-V3,  
 152 and o1-mini, and delivers critical insights for enhancing model reliability.

153 **References**

- 154 David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine*  
155 *Learning research*, 3(Jan):993–1022, 2003.
- 156 Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared  
157 Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri,  
158 Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan,  
159 Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian,  
160 Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios  
161 Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino,  
162 Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders,  
163 Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa,  
164 Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob  
165 McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating  
166 large language models trained on code, 2021. URL <https://arxiv.org/abs/2107.03374>.
- 167 DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu,  
168 Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu,  
169 Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao  
170 Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan,  
171 Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao,  
172 Guantang Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding,  
173 Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang  
174 Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong,  
175 Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao,  
176 Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang,  
177 Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang,  
178 Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L.  
179 Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang,  
180 Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng  
181 Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanbiao Zhao, Wen Liu, Wenfeng  
182 Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan  
183 Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang,  
184 Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen,  
185 Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li,  
186 Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang,  
187 Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan,  
188 Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia  
189 He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong  
190 Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha,  
191 Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang,  
192 Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li,  
193 Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen  
194 Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.  
195 URL <https://arxiv.org/abs/2501.12948>.
- 196 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep  
197 bidirectional transformers for language understanding, 2019. URL <https://arxiv.org/abs/1810.04805>.
- 199 Zorik Gekhman, Jonathan Herzig, Roei Aharoni, Chen Elkind, and Idan Szpektor. Trueteacher:  
200 Learning factual consistency evaluation with large language models, 2023. URL <https://arxiv.org/abs/2305.11171>.
- 202 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad  
203 Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan,  
204 Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev,  
205 Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru,  
206 Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak,  
207 Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu,

208 Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle  
209 Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego  
210 Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova,  
211 Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel  
212 Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon,  
213 Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan  
214 Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet,  
215 Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde,  
216 Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie  
217 Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua  
218 Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak,  
219 Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley  
220 Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yearly, Laurens van der Maaten, Lawrence  
221 Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas  
222 Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri,  
223 Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie  
224 Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes  
225 Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne,  
226 Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajwal  
227 Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong,  
228 Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic,  
229 Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie  
230 Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana  
231 Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie,  
232 Sharan Narang, Sharath Rapparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon  
233 Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan,  
234 Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas  
235 Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami,  
236 Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti,  
237 Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier  
238 Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao  
239 Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song,  
240 Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe  
241 Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya  
242 Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei  
243 Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu,  
244 Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit  
245 Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury,  
246 Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer,  
247 Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu,  
248 Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido,  
249 Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu  
250 Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer,  
251 Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu,  
252 Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc  
253 Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily  
254 Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers,  
255 Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank  
256 Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee,  
257 Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan,  
258 Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph,  
259 Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog,  
260 Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James  
261 Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny  
262 Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings,  
263 Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai  
264 Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik  
265 Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle  
266 Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng

- 267 Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish  
268 Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim  
269 Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle  
270 Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang,  
271 Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam,  
272 Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier,  
273 Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia  
274 Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro  
275 Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani,  
276 Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy,  
277 Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin  
278 Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu,  
279 Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh  
280 Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay,  
281 Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang,  
282 Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie  
283 Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta,  
284 Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman,  
285 Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun  
286 Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria  
287 Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru,  
288 Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz,  
289 Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv  
290 Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi,  
291 Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait,  
292 Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The  
293 llama 3 herd of models, 2024. URL <https://arxiv.org/abs/2407.21783>.
- 294 Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob  
295 Steinhardt. Measuring massive multitask language understanding. In *International Conference on*  
296 *Learning Representations*, 2021a. URL <https://openreview.net/forum?id=d7KBjmI3GmQ>.
- 297 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,  
298 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*,  
299 2021b.
- 300 Yue Huang, Chujie Gao, Siyuan Wu, Haoran Wang, Xiangqi Wang, Yujun Zhou, Yanbo Wang,  
301 Jiayi Ye, Jiawen Shi, Qihui Zhang, Yuan Li, Han Bao, Zhaoyi Liu, Tianrui Guan, Dongping  
302 Chen, Ruoxi Chen, Kehan Guo, Andy Zou, Bryan Hooi Kuen-Yew, Caiming Xiong, Elias Stengel-  
303 Eskin, Hongyang Zhang, Hongzhi Yin, Huan Zhang, Huaxiu Yao, Jaehong Yoon, Jieyu Zhang,  
304 Kai Shu, Kaijie Zhu, Ranjay Krishna, Swabha Swayamdipta, Taiwei Shi, Weijia Shi, Xiang Li,  
305 Yiwei Li, Yuexing Hao, Yuexing Hao, Zhihao Jia, Zhize Li, Xiuying Chen, Zhengzhong Tu,  
306 Xiyang Hu, Tianyi Zhou, Jieyu Zhao, Lichao Sun, Furong Huang, Or Cohen Sasson, Prasanna  
307 Sattigeri, Anka Reuel, Max Lamparth, Yue Zhao, Nouha Dziri, Yu Su, Huan Sun, Heng Ji, Chaowei  
308 Xiao, Mohit Bansal, Nitesh V. Chawla, Jian Pei, Jianfeng Gao, Michael Backes, Philip S. Yu,  
309 Neil Zhenqiang Gong, Pin-Yu Chen, Bo Li, and Xiangliang Zhang. On the trustworthiness  
310 of generative foundation models: Guideline, assessment, and perspective, 2025. URL <https://arxiv.org/abs/2502.14296>.
- 312 Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando  
313 Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free  
314 evaluation of large language models for code, 2024. URL <https://arxiv.org/abs/2403.07974>.
- 316 Che Jiang, Biqing Qi, Xiangyu Hong, Dayuan Fu, Yang Cheng, Fandong Meng, Mo Yu, Bowen  
317 Zhou, and Jie Zhou. On large language models’ hallucination with regard to known facts, 2024.  
318 URL <https://arxiv.org/abs/2403.20009>.
- 319 Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language  
320 models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

- 321 Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie  
322 Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. Dynabench: Rethinking bench-  
323 marking in nlp. *arXiv preprint arXiv:2104.14337*, 2021.
- 324 Xiang Lisa Li, Evan Zheran Liu, Percy Liang, and Tatsunori Hashimoto. Autobencher: Creating  
325 salient, novel, difficult datasets for language models. *arXiv preprint arXiv:2407.08351*, 2024a.
- 326 Yuangang Li, Jiaqi Li, Zhuo Xiao, Tiankai Yang, Yi Nian, Xiyang Hu, and Yue Zhao. Nlp-adbench:  
327 Nlp anomaly detection benchmark. *arXiv preprint arXiv:2412.04784*, 2024b.
- 328 Renjie Liang, Li Li, Chongzhi Zhang, Jing Wang, Xizhou Zhu, and Aixin Sun. Tvr-ranking: A  
329 dataset for ranked video moment retrieval with imprecise queries, 2024. URL <https://arxiv.org/abs/2407.06597>.  
330
- 331 Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao,  
332 Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint*  
333 *arXiv:2412.19437*, 2024.
- 334 Cong Lu, Shengran Hu, and Jeff Clune. Automated capability discovery via model self-exploration.  
335 *arXiv preprint arXiv:2502.07577*, 2025.
- 336 Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Selfcheckgpt: Zero-resource black-box  
337 hallucination detection for generative large language models, 2023. URL <https://arxiv.org/abs/2303.08896>.  
338
- 339 Sara Merken. Ai 'hallucinations' in court papers spell trouble for lawyers, 2025.  
340 URL <https://www.reuters.com/technology/artificial-intelligence/ai-hallucinations-court-papers-spell-trouble-lawyers-2025-02-18/>. Accessed:  
341 2025-03-21.  
342
- 343 OpenAI. Gpt-4o mini: advancing cost-efficient intelligence. <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>, 2024a.  
344
- 345 OpenAI. Openai o1 system card, 2024b. URL <https://cdn.openai.com/o1-system-card.pdf>.  
346
- 347 OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni  
348 Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor  
349 Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian,  
350 Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny  
351 Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks,  
352 Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea  
353 Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen,  
354 Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung,  
355 Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch,  
356 Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty  
357 Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte,  
358 Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel  
359 Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua  
360 Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike  
361 Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon  
362 Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne  
363 Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo  
364 Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar,  
365 Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik  
366 Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich,  
367 Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy  
368 Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie  
369 Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini,  
370 Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne,  
371 Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David  
372 Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie



373 Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély,  
374 Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo  
375 Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano,  
376 Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng,  
377 Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto,  
378 Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power,  
379 Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis  
380 Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted  
381 Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel  
382 Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon  
383 Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky,  
384 Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie  
385 Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng,  
386 Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun  
387 Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang,  
388 Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian  
389 Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren  
390 Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming  
391 Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao  
392 Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL  
393 <https://arxiv.org/abs/2303.08774>.

394 Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller,  
395 and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*,  
396 2019.

397 Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin  
398 Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity’s last exam. *arXiv preprint*  
399 *arXiv:2501.14249*, 2025.

400 Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan  
401 Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang,  
402 Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin  
403 Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi  
404 Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan,  
405 Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL  
406 <https://arxiv.org/abs/2412.15115>.

407 Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks.  
408 *arXiv preprint arXiv:1908.10084*, 2019.

409 Taiwei Shi, Kai Chen, and Jieyu Zhao. Safer-instruct: Aligning language models with automated  
410 preference data, 2024. URL <https://arxiv.org/abs/2311.08685>.

411 Taiwei Shi, Zhuoer Wang, Longqi Yang, Ying-Chun Lin, Zexue He, Mengting Wan, Pei Zhou, Sujay  
412 Jauhar, Sihao Chen, Shan Xia, Hongfei Zhang, Jieyu Zhao, Xiaofeng Xu, Xia Song, and Jennifer  
413 Neville. Wildfeedback: Aligning llms with in-situ user interactions and feedback, 2025. URL  
414 <https://arxiv.org/abs/2408.15549>.

415 Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt:  
416 Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint*  
417 *arXiv:2010.15980*, 2020.

418 Adi Simhi, Jonathan Herzig, Idan Szpektor, and Yonatan Belinkov. Distinguishing ignorance from  
419 error in llm hallucinations. *arXiv preprint arXiv:2410.22071*, 2024.

420 Linxin Song, Jieyu Zhang, Lechao Cheng, Pengyuan Zhou, Tianyi Zhou, and Irene Li. Nlpbench:  
421 Evaluating large language models on solving nlp problems. *arXiv preprint arXiv:2309.15630*,  
422 2023.

423 Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and  
424 Yu Cheng. Conflictbank: A benchmark for evaluating the influence of knowledge conflicts in llm,  
425 2024. URL <https://arxiv.org/abs/2408.12076>.

- 426 Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine*  
427 *learning research*, 9(11), 2008.
- 428 Siyuan Wang, Zhuohan Long, Zhihao Fan, Zhongyu Wei, and Xuanjing Huang. Benchmark self-  
429 evolving: A multi-agent framework for dynamic llm evaluation, 2024a. URL <https://arxiv.org/abs/2402.11443>.
- 431 Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R.  
432 Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level  
433 scientific problem-solving abilities of large language models, 2024b. URL <https://arxiv.org/abs/2307.10635>.
- 435 Wikipedia contributors. Wikipedia, the free encyclopedia, 2004. URL <https://www.wikipedia.org/>. [Online; accessed 22-July-2004].
- 437 Siyuan Wu, Yue Huang, Chujie Gao, Dongping Chen, Qihui Zhang, Yao Wan, Tianyi Zhou, Xian-  
438 gliang Zhang, Jianfeng Gao, Chaowei Xiao, and Lichao Sun. Unigen: A unified framework for  
439 textual dataset generation using large language models, 2024. URL <https://arxiv.org/abs/2406.18966>.
- 441 Tiankai Yang, Yi Nian, Shawn Li, Ruiyao Xu, Yuangang Li, Jiaqi Li, Zhuo Xiao, Xiyang Hu, Ryan  
442 Rossi, Kaize Ding, et al. Ad-llm: Benchmarking large language models for anomaly detection.  
443 *arXiv preprint arXiv:2412.11142*, 2024.
- 444 Lei Yu, Meng Cao, Jackie CK Cheung, and Yue Dong. Mechanistic understanding and mitigation  
445 of language model non-factual hallucinations. In Yaser Al-Onaizan, Mohit Bansal, and Yun-  
446 Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*,  
447 pp. 7943–7956, Miami, Florida, USA, November 2024. Association for Computational Linguis-  
448 tics. doi: 10.18653/v1/2024.findings-emnlp.466. URL <https://aclanthology.org/2024.findings-emnlp.466/>.
- 450 Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu  
451 Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal under-  
452 standing and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on*  
453 *Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024.
- 454 Zhiyuan Zeng, Yizhong Wang, Hannaneh Hajishirzi, and Pang Wei Koh. Evaltree: Profiling language  
455 model weaknesses via hierarchical capability trees, 2025. URL <https://arxiv.org/abs/2503.08893>.
- 457 Jieyu Zhang, Weikai Huang, Zixian Ma, Oscar Michel, Dong He, Tanmay Gupta, Wei-Chiu Ma,  
458 Ali Farhadi, Aniruddha Kembhavi, and Ranjay Krishna. Task me anything. *arXiv preprint*  
459 *arXiv:2406.11775*, 2024a.
- 460 Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong  
461 Yang, Pengjun Xie, Fei Huang, et al. mgte: Generalized long-context text representation and  
462 reranking models for multilingual text retrieval. *arXiv preprint arXiv:2407.19669*, 2024b.
- 463 Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemaoy Liu, Tingchen Fu, Xinting Huang, Enbo Zhao,  
464 Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi.  
465 Siren’s song in the ai ocean: A survey on hallucination in large language models, 2023. URL  
466 <https://arxiv.org/abs/2309.01219>.
- 467 Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli  
468 He, Kam-Fai Wong, and Pasquale Minervini. Steering knowledge selection behaviours in llms via  
469 sae-based representation engineering, 2025. URL <https://arxiv.org/abs/2410.15999>.
- 470 Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. Dyval:  
471 Dynamic evaluation of large language models for reasoning tasks, 2024a. URL <https://arxiv.org/abs/2309.17167>.
- 473 Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. Dynamic evaluation of large  
474 language models by meta probing agents, 2024b. URL <https://arxiv.org/abs/2402.14865>.

---

**Algorithm 1:** Stochastic Error Ascent

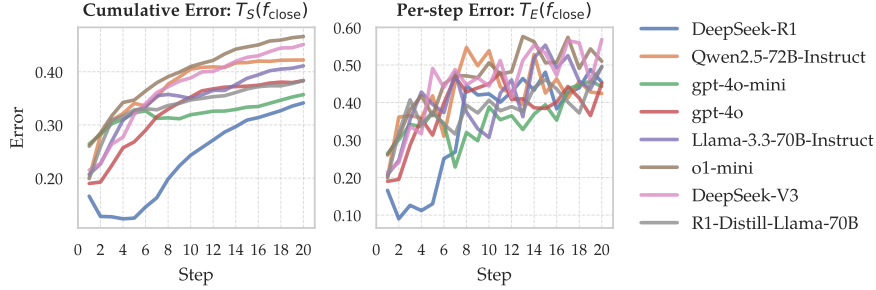
---

**Input:** Knowledge base  $\mathcal{K}$ , closed-weight model  $f_{\text{close}}$ , budget  $C$ , random initial paragraph set  $B$ , threshold  $\xi$  and  $\gamma$ .

**Output:** Optimal paragraph set  $\hat{\mathcal{S}}$

```
1 Initialization:  $t \leftarrow 1$ ,  $\text{cost} \leftarrow 0$ ,  $\mathcal{S}_t \leftarrow \emptyset$ ,  $\mathcal{P}_{\text{source}}^{(t)} \leftarrow \emptyset$ 
2 while  $\text{cost} < C$  do
3   if  $t = 1$  then
4      $E \leftarrow B$  ▷ No source error at step 1; use random initial batch.
5   else
6      $E \leftarrow \text{FindSim}(\mathcal{K}, \mathcal{P}_{\text{source}}^{(t)})$  ▷ Eq.(4): Sample an error related candidate batch.
7      $\mathcal{S}_{t+1} = \mathcal{S}_t \cup E$  ▷ Update the target subset.
8      $\mathcal{P}_{\text{source}}^{(t+1)} \leftarrow \mathcal{P}_{\text{source}}^{(t)} \cup \{p \in E \mid T_{\{p\}} > \xi\}$  ▷ Update source error set.
9      $\mathcal{K} \leftarrow \mathcal{K} \setminus \{p \in E \mid T_{\{p\}} > \xi\}$  ▷ Remove new source error from  $\mathcal{K}$  to avoid loop.
10     $\mathcal{G}_{t+1} \leftarrow (\mathcal{P}_{\text{source}}^{(t+1)}, \mathcal{E}(\mathcal{P}_{\text{source}}^{(t)}, \mathcal{P}_{\text{source}}^{(t+1)}))$  ▷ Update relation DAG.
11     $\mathcal{P}_{\text{source}}^{(t+1)} \leftarrow \mathcal{P}_{\text{source}}^{(t+1)} \setminus \{p \in \mathcal{P}_{\text{source}}^{(t+1)} \mid \pi_{\mathcal{G}_{t+1}}(p) < \gamma\}$  ▷ Eq.(5): Pruning  $\mathcal{P}_{\text{source}}$ .
12     $\text{cost} \leftarrow \text{cost}(T_E(f_{\text{close}}))$  ▷ Update cost. We did one-time inference on  $E$ .
13     $t \leftarrow t + 1$ 
14 return  $\hat{\mathcal{S}}$ 
```

---

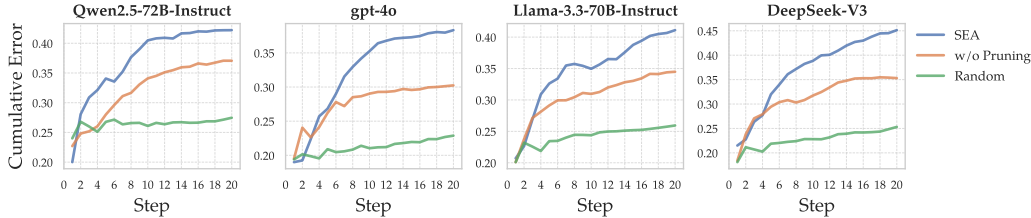


**Figure 2:** Per-step error  $T_E(f_{\text{close}})$  and cumulative error  $T_S(f_{\text{close}})$  for each model. We observe that the errors of all models are positively related to step, indicating SEA can gradually and continually find the model’s knowledge deficiencies from the knowledge base.

## 475 A Analyzing Stochastic Error Ascent

476 **Validating with human evaluation.** In this study, we utilize gpt-4o as the question-answer generator.  
477 We assess the quality of its outputs through a human evaluation involving five college-level students,  
478 who verify the truthfulness of the generated answers by cross-referencing them with corresponding  
479 paragraphs in the knowledge base, mirroring the input provided to the generator. From 1,000  
480 randomly selected questions generated across 20 steps, our evaluation achieved a 100% human pass  
481 rate, confirming that all answers were both present and correct within the associated paragraphs,  
482 consistent with the results in Li et al. (2024a), which also uses LLM as a question generator with  
483 Wikipedia documents.

484 **Convergence analysis.** To analyze the convergence behavior of SEA, we conduct an empirical  
485 study, given the inaccessibility of internal activations in closed-weight models during the search  
486 process. We track both the cumulative error  $T_S(f_{\text{close}})$  and per-step error  $T_E(f_{\text{close}})$  over 20 iterations  
487 of SEA (Fig. 2). For cumulative error (left), we observe a steep initial increase followed by a plateau  
488 across all models, indicating that SEA rapidly identifies high-impact errors early in the search, then  
489 gradually uncovers subtler or less frequent failure modes. Notably, models such as o1-mini and  
490 R1-Distill-Llama-70B exhibit higher peak cumulative errors, while DeepSeek-R1 shows a more  
491 gradual ascent. Per-step error (right) further highlights the adaptability of SEA, as it consistently



**Figure 3:** Ablation studies on the component contribution of SEA. We compare SEA with its two variants: without source pruning (i.e., pass the lines 10 and 11 in Alg. 1) and random selection (i.e., pass the lines 9, 10, and 11 in Alg. 1). We observe that each component contributes equally to SEA.

Model Cost	DeepSeek-R1	R1-Distill-Llama-70B	o1-mini	DeepSeek-V3	Llama-3.3-70B	Qwen2.5-72B	gpt-4o-mini	gpt-4o
Generation Cost (US \$)	28.163	28.660	31.094	30.208	29.776	29.542	28.243	32.897
Inference Cost (US \$)	48.360	7.888	39.708	1.261	0.868	0.37	0.347	7.905
Inference Output Tokens	19,608,736	10,836,882	8,507,015	380,099	1,024,942	272,566	125,117	308,145

**Table 1:** Question generation cost, inference cost, and output tokens at inference time across 20 steps (results in Fig. 2; 20,000 questions in total). We can see a significant gap between reasoning models (DeepSeek-R1, R1-Distill-Llama-70B, and o1-mini) and other non-reasoning models.

492 uncovers challenging inputs. The slope of per-step error varies across models: gpt-4o-mini shows  
 493 a relatively flat trajectory, while o1-mini and DeepSeek-V3 show steeper climbs.

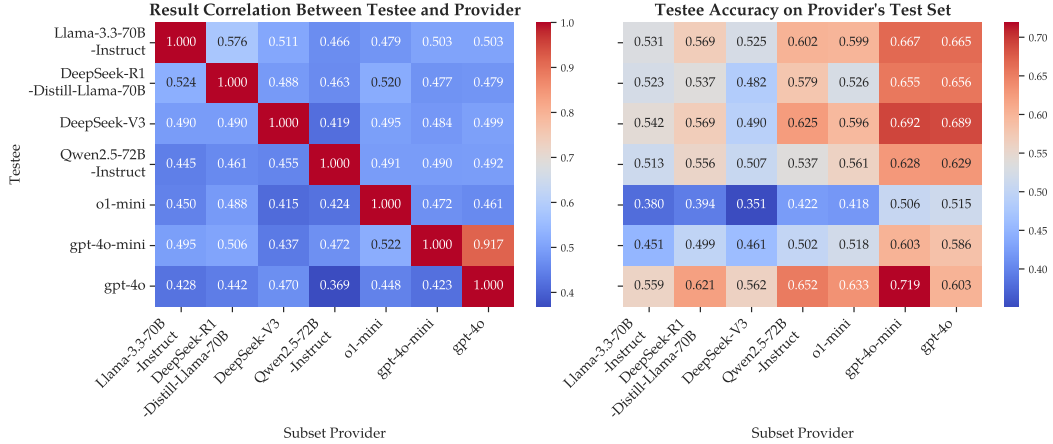
494 **Ablation studies.** SEA has two significant procedures: (1) collecting and updating the source error  
 495 set (line 9 in Alg. 1) and (2) directed graph construction and source pruning (lines 10 and 11 in  
 496 Alg. 1). Procedure Two relies on the result from Procedure One. To analyze the contribution to both  
 497 procedures, we conduct two ablation experiments on four selected LLMs (Qwen2.5-72B-Instruct,  
 498 gpt-4o, Llama-3.3-70B-Instruct, and DeepSeek-V3) by removing procedure one and both  
 499 procedures to create two variants of SEA: (i) w/o pruning and (ii) random selection. For the  
 500 random selection, we change all error-related batches into random batches where we randomly select  
 501 paragraphs from the knowledge base. We summarize the results in Fig. 3, from which we observe  
 502 that the contribution of each procedure to SEA is linearly increasing. The cumulative error of random  
 503 selection barely increases, while the gap between SEA and w/o pruning variant starts increasing after  
 504 a few steps, indicating that low-quality sources that haven’t been pruned by the cumulative error start  
 505 negatively affecting SEA.

506 **Cost Analysis** We summarize all the model’s costs for Fig. 2 results in Tab. 1. We use  
 507 cloud API served by DeepSeek and DeepInfra for all open-sourced models (DeepSeek-R1,  
 508 R1-Distill-Llama-70B, DeepSeek-V3, Llama-3.3-70B, and Qwen2.5-72B), and our cost calcula-  
 509 tion is based on their token price per million tokens. We observe that the variance for the generation  
 510 cost is low, while that for the inference cost is high. A significant difference can be discovered by  
 511 looking into the inference cost between reasoning models (DeepSeek-R1, R1-Distill-Llama-70B,  
 512 and o1-mini) and other non-reasoning models. DeepSeek-R1 has extremely long inference token  
 513 length even for the multiple choice questions, which causes the highest cost, though its price-per-token  
 514 is lower than o1-mini and gpt-4o.

## 515 B Analyzing LLMs from the Discovered Deficiencies

516 **Query 1: How does the model perform on other LLMs’ optimal subset?** In this study, we search  
 517 for eight LLMs’ deficiencies and consequently create eight unique optimal subsets according to  
 518 Eq. 1. We conduct the cross-validation by testing each model (testee) on the other models’ (providers)  
 519 subsets, with correlation and accuracy results summarized in Fig. 4<sup>2</sup>. Our analysis revealed an asym-  
 520 metrical correlation pattern, where the direction of testing significantly influenced outcomes—for  
 521 instance, gpt-4o-mini as the testee exhibited a high correlation (0.917) with gpt-4o as the provider,  
 522 yet the reverse yielded a notably lower correlation (0.423), suggesting that gpt-4o-mini may be a dis-  
 523 tilled variant optimized to emulate gpt-4o’s behavior, while its subset highlights divergent behaviors.  
 524 Models from similar families, such as gpt-4o and gpt-4o-mini, demonstrated higher correlations,

<sup>2</sup>We omit the DeepSeek-R1 due to the budget limitation.



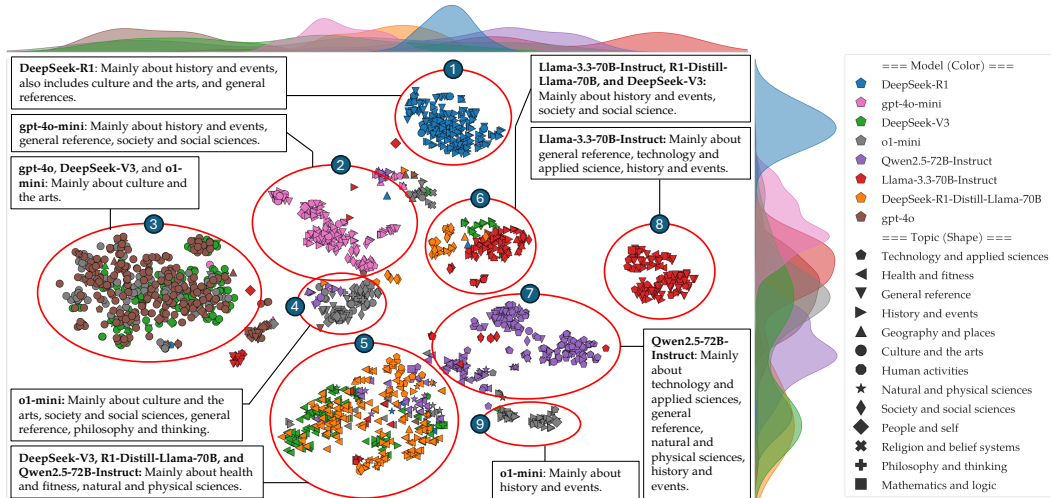
**Figure 4:** Comparison of cross-validation between each model. X-axis indicates the subset provider (i.e.,  $\hat{S}$  provider; sourced from experiments in Fig. 2), and Y-axis denotes the testee. We summarize two results: (1) correlation between testee result and provider result, and (2) accuracy of testee on each provider’s results. The higher the correlation, the more similar the answers of the two models are. Similarly, the higher the testee’s accuracy, the more challenging the provider’s question.

Cluster ID	Models	Main Categories	Error Pattern
3	gpt-4o DeepSeek-V3 o1-mini	Culture and the arts	(1) Challenging in Chronological Analysis (2) Unfamiliar with Locational Details (3) Issues in Pattern Recognition (4) Inaccurate Data Synthesis (5) Collaborative and Relational Patterns
5	Qwen2.5-72B-Instruct Llama-3.3-70B-Instruct R1-Distill-Llama-70B	Health and fitness Natural and physics science	(1) Challenges with Chronological and Historical Data (2) Issues with Contextual and Performance-Related Information (3) Inaccurate Interpretation of Patterns and Trends (4) Over-reliance on Assumptions and Generalizations (5) Difficulty with Contextual Associations and Identifications

**Table 2:** Error patterns for models in cluster 3 and 5 in Fig. 5. We aggregate error patterns from the question level to the paragraph level and finally to the model level.

525 indicative of shared output patterns, whereas cross-family comparisons, like Llama-3.3-70B versus  
 526 gpt-4o, showed moderate correlations (0.4-0.5). The accuracy matrix illustrates model performance  
 527 across subsets generated by different providers, with gpt-4o and gpt-4o-mini’s subsets being less  
 528 challenging, yielding higher accuracies from testee models, while DeepSeek-V3’s subset posed a  
 529 greater difficulty, and o1-mini consistently underperformed, underscoring varying model capabilities  
 530 and subset complexities.

531 **Query 2: What kind of knowledge do the models lack?** To investigate this, we visualize the  
 532 source error  $p \in \mathcal{P}_{\text{source}}$  in  $\hat{S}$  of each model by compressing the embedding of  $\mathcal{P}_{\text{source}}$  with t-  
 533 SNE (Van der Maaten & Hinton, 2008) in Fig. 5. We mark different models in different colors, and  
 534 different Wikipedia categories in different marker shapes. We can see that gpt-4o, DeepSeek-V3,  
 535 and o1-mini overlap highly, with errors concentrated in culture and the arts. gpt-4o-mini and  
 536 DeepSeek-R1 have unique clusters, less overlapping with other models, while both weaknesses  
 537 include history and events, and society and social science. We also notice that DeepSeek-V3,  
 538 R1-Distill-Llama-70B, and Qwen2.5-72B-Instruct have error overlap on health and fitness,  
 539 and natural and physical sciences, while Qwen2.5-72B-Instruct also has its only cluster on tech-  
 540 nology and applied sciences, general reference, natural and physical sciences, and history and events.  
 541 Errors from Llama-3.3-70B-Instruct have overlap with DeepSeek-R1-Distill-Llama-70B  
 542 on history and events, indicating the possible inherent relationship between these two models. We  
 543 observe that a wide range (5 out of 13) of categories can trigger o1-mini’s error, including culture  
 544 and the arts, society and social sciences, general reference, philosophy and thinking, and history  
 545 and events, indicating a significant gap of training data and training strategies between o1-mini and  
 546 other models.



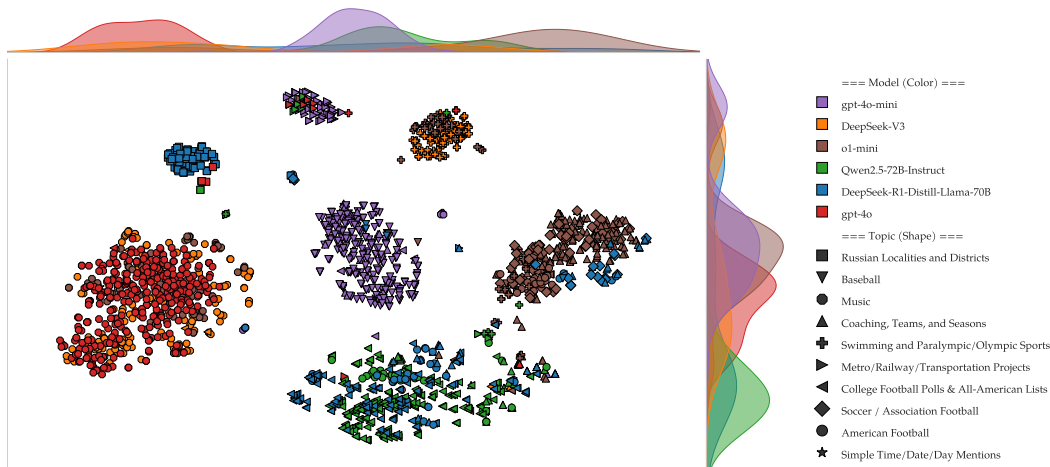
**Figure 5:** Error distribution of each testee model. We search with the **same initial set** from thirteen categories of Wikipedia. We visualize the results by t-SNE without a clustering algorithm. Each point in the figure denotes the corresponding model’s source error  $p \in \mathcal{P}_{\text{source}}$ . Different colors denote different models, and different markers denote different categories. We can observe natural clusters of each model discovered by SEA according to their knowledge omission areas.

547 **Query 3: What pattern causes a model to fail in a specific category?** To understand the behavior  
 548 of a model’s error, we aggregate the question-level error into paragraph-level errors and further  
 549 summarize them as the model’s error behavior. We choose clusters 3 and 5 in Fig. 5 and perform the  
 550 multi-level aggregation by prompting an LLM to retrieve question-level error pattern, paragraph-level  
 551 pattern, and finally the model and cluster-level error pattern. We only analyze clusters 3 and 5 for  
 552 budget reasons and summarize the error patterns in Tab. 2. The models in cluster 3 appear to have  
 553 difficulties with tasks requiring historical context, spatial awareness, and relational reasoning in the  
 554 domain of culture and arts. The models in cluster 5 seem to have broader issues with contextual  
 555 understanding and precision, particularly in domains requiring empirical rigor. Both clusters exhibit  
 556 challenges with chronological analysis and pattern recognition, indicating that these might be common  
 557 limitations across various LLMs when dealing with complex domains.

## 558 C Related work

559 **Dynamic benchmarking.** Static benchmarks (Chen et al., 2021; Hendrycks et al., 2021b,a; Song  
 560 et al., 2023; Liang et al., 2024; Yue et al., 2024; Li et al., 2024b; Yang et al., 2024; Wang et al., 2024b;  
 561 Jain et al., 2024; Shi et al., 2024, 2025; Phan et al., 2025) suffer from data leakage and high building  
 562 costs. Dynamic frameworks address this via automated, evolving data generation. Dynabench (Kiela  
 563 et al., 2021) integrates human-model interaction, while Task Me Anything (Zhang et al., 2024a)  
 564 enables scalable, user-driven evaluation. Recent work—DyVal (Zhu et al., 2024a,b), UniGen (Wu  
 565 et al., 2024), and Benchmark Self-Evolving (Wang et al., 2024a)—employs probing agents and multi-  
 566 agent systems for iterative refinement. AutoBencher (Li et al., 2024a) casts benchmark construction  
 567 as optimization to surface model flaws but remains limited by static templates or annotated data  
 568 (Huang et al., 2025). In contrast, SEA adopts a fully adaptive, error-driven probing strategy using  
 569 relation DAG to uncover failures.

570 **Model behavior understanding.** We investigate factual knowledge modeling in LLMs, focusing  
 571 on distinguishing correct from plausible-but-false outputs—a challenge rooted in their latent fact-  
 572 ual encoding (Petroni et al., 2019). Techniques such as gradient-guided prompt generation (Shin  
 573 et al., 2020), systematic prompt design (Jiang et al., 2020), and tools like Automated Capability  
 574 Discovery (Lu et al., 2025) further expand the evaluation scope. EvalTree (Zeng et al., 2025) further  
 575 reveals vulnerabilities via hierarchical capability trees. However, reliance on hand-crafted prompts or  
 576 constrained queries limits their efficacy in surfacing nuanced misinformation.



**Figure 6:** Error distribution of each testee model. We search with the **same random initial set** from Wikipedia **without specifying specific topics**. We visualize the results by t-SNE without a clustering algorithm. Each point in the figure denotes the corresponding model’s source error  $p \in \mathcal{P}_{\text{source}}$ . Different colors denote different models, and different markers denote different categories.

## 577 D Extra Analysis and Case Studies

578 **Query 4: Will LLM produce misinformation on its unknown knowledge?** In order to investigate  
 579 this question, we randomly sampled 5 questions from the gpt-4o optimal subset, specifically selecting  
 580 those where SEA previously identified factual errors or vulnerabilities in LLMs, representing unknown  
 581 knowledge. In this experiment, we modify these questions into a free-response format using gpt-4o  
 582 and retest them on gpt-4o. Errors such as Tab. 3 (incorrect doctoral year), Tab. 4 (misidentified  
 583 event), Tab. 5 (wrong attribution of artist and medium), Tab. 6 (misattributed venue), and Tab. 7  
 584 (erroneous exhibition location) clearly show that when LLM encounter unknown knowledge, they  
 585 may produce misinformation, as highlighted by the underlined errors in each example.

586 **Query 5: Will LLM exhibit memory-context conflict when answering questions related to**  
 587 **the detected deficiencies?** Memory-context conflicts are known as conflicts between pre-trained  
 588 parametric knowledge and retrieved information (Su et al., 2024; Zhao et al., 2025). To evaluate  
 589 whether LLM exhibits such a memory-context conflict when addressing questions probing their  
 590 deficiencies, we randomly sampled 1000 incorrect questions from the gpt-4o QA set and augmented  
 591 each with its corresponding retrieved factual context, testing on gpt-4o. Despite this external  
 592 supplementation, the accuracy of gpt-4o improved only to 28.6%, indicating that the model adopts  
 593 the provided context in merely about one-third of cases while predominantly relying on its pre-trained  
 594 internal knowledge in the remaining instances. This outcome indicates that even essential external-  
 595 augmented information may not sufficiently override LLM’s entrenched memory. Therefore, our  
 596 findings mean that LLM do exhibit a clear memory-context conflict. The relevant testee prompt is  
 597 listed in Tab. 12.

598 **Query 6: What deficiencies can be discovered from a random initial set without category**  
 599 **constraints?** As described in Sec. 3, our random initial batch is uniformly sampled across 13  
 600 Wikipedia categories. However, the downstream task may lack well-defined categories. To investigate  
 601 the solution, we test six models (gpt-4o, gpt-4o-mini, o1-mini, Qwen2.5-72B-Instruct,  
 602 DeepSeek-R1-Distill-Llama-70B, and DeepSeek-V3) by adopting a complete random initial  
 603 batch without any topic constraint when performing SEA. We search each model for 20 steps with the  
 604 same setting as described in Sec. 3, summarizing the topic from each model by LDA (Blei et al., 2003)  
 605 and aggregating the topic from all models into 10 general topics, including: Baseball, American  
 606 Football, Metro/Railway/Transportation Projects, Swimming/Paralympic/Olympic  
 607 Sports, Music, Soccer/Association Football, Simple Time/Date/Day Mentions,  
 608 College Football Polls/All-American Lists, Coaching/Teams/Seasons, and Russian  
 609 Localities/Districts. These topics are mainly about sports and health, identifying sys-

### Example 1

**Title:**

James B. Stump/Career

**Original question:**

What year did James B. Stump receive his doctoral degree from Boston University?

**Ground true answer:**

2000

**Modified question:**

Discuss the year in which James B. Stump was awarded his doctoral degree from Boston University.

**Misinformation:**

James B. Stump was awarded his doctoral degree from Boston University in 1998.

**Table 3:** Example 1 for query 4. The correct doctoral year is "2000", but the misinformation incorrectly states "1998". The incorrect information has been highlighted using underlines.

610 thematic failure patterns of different LLMs in this area. We further visualize the result of the  
611 source error in Fig. 6. We first observe a similar distribution as in Fig. 5, where gpt-4o,  
612 DeepSeek-V3, and o1-mini share similar failure patterns, while Qwen2.5-72B-Instruct and  
613 DeepSeek-R1-Distill-Llama-70B share similar failure patterns. We notice a large volume of  
614 DeepSeek-R1-Distill-Llama-70B and o1-mini aligns with the observation in Fig. 5. We also  
615 observe that gpt-4o, DeepSeek-V3, and o1-mini mainly fail in music-related paragraphs, while  
616 gpt-4o-mini mainly fails in Baseball and Transportation project-related topics.

617 **Query 7: How can we extend the searching scope?** Following the settings in Zhang et al. (2024a),  
618 we try fitting a BERT (Devlin et al., 2019) model to identify if a paragraph from a knowledge base  
619 can trigger an LLM's error. We first collect 4,402 retrieved paragraphs from 50 rounds of SEA  
620 searching process on gpt-4o. We annotate the paragraphs as 0 if the average accuracy across the  
621 generated questions is lower than 0.5, and 1 otherwise. The collected paragraphs are split into  
622 training, validation, and test sets, respectively, with the ratio 8:1:1. We adopt early-stopping to  
623 prevent overfitting according to the validation performance. We tried the bert-base-uncased and  
624 bert-large-uncased respectively. The bert-base-uncased achieves 66.22% average accuracy  
625 on the test set, while bert-large-uncased achieves 67.85% average accuracy. These results suggest  
626 that larger BERT models can capture the subtle semantic cues that differentiate paragraphs likely to  
627 mislead an LLM. However, the overall performance indicates that this is a challenging classification  
628 task, potentially due to the noisy or indirect relationship between paragraph content and downstream  
629 model behavior.

## 630 E Prompt of SEA

631 This section is supplemented with some additional details when implementing the pipeline of SEA,  
632 which is introduced in Section 2.1. To be more specific, Tab 8, Tab 9, Tab 10, Tab 11, and Tab 12  
633 are prompts for multiple choice question generation, question rephrasing, analyzing error pattern,  
634 SEA testee model, and testee model in query 5, respectively.



## Example 2

---

**Title:**

Mimi Cherono Ng'ok/Group exhibitions

**Original question:**

In which event was the series "Do You Miss Me? Sometimes, Not Always" by Mimi Cherono Ng'ok included?

**Ground true answer:**

African Photography Encounters

**Modified question:**

Discuss the event in which Mimi Cherono Ng'ok's series "Do You Miss Me? Sometimes, Not Always" was included. Provide details about the event and its significance.

**Misinformation:**

Mimi Cherono Ng'ok's photographic series "Do You Miss Me? Sometimes, Not Always" was included in the 2019 Whitney Biennial, a prestigious contemporary art exhibition held at the Whitney Museum of American Art in New York City. The Whitney Biennial is one of the most significant events in the art world, known for showcasing the latest trends and emerging voices in American art. It serves as a barometer of contemporary art in the United States, often highlighting diverse perspectives and innovative practices. The 2019 edition of the Biennial was particularly noted for its focus on issues of identity, social justice, and the political climate, reflecting the broader cultural and social dynamics of the time. Mimi Cherono Ng'ok's work, which often explores themes of memory, displacement, and emotional landscapes, fit well within this context. Her series "Do You Miss Me? Sometimes, Not Always" delves into personal and collective narratives, capturing the nuances of absence and longing through evocative imagery. The inclusion of Ng'ok's work in the Whitney Biennial not only elevated her profile as an artist but also contributed to the broader discourse on contemporary African art and its place within the global art scene. Her participation underscored the Biennial's commitment to presenting diverse artistic voices and addressing complex themes that resonate with audiences worldwide.

**Table 4:** Example 2 for query 4. The proper event is "African Photography Encounters," yet the misinformation erroneously identifies it as the "2019 Whitney Biennial". The incorrect information has been highlighted using underlines.

### Example 3

**Title:**

Museo Nacional Centro de Arte Reina Sofía/Gallery

**Original question:**

Identify the artist and medium for the 1923 painting 'Portrait of Tristan Tzara'.

**Ground true answer:**

Robert Delaunay, oil on cardboard

**Modified question:**

Describe the artist and the medium used for the painting 'Portrait of Tristan Tzara' created in 1923.

**Misinformation:**

The painting 'Portrait of Tristan Tzara' was created by the Romanian avant-garde artist Marcel Janco in 1923. Marcel Janco was a key figure in the Dada movement, which was known for its anti-establishment and anti-art sentiments. The medium used for this painting is oil on canvas. Janco's work often incorporated elements of abstraction and cubism, reflecting the experimental and revolutionary spirit of the Dada movement. Tristan Tzara, the subject of the portrait, was a prominent Dada poet and one of the movement's founders, making this painting a significant piece in the context of early 20th-century avant-garde art.

**Table 5:** Example 3 for query 4. It shows that the true artist and medium are "Robert Delaunay, oil on cardboard", while the misinformation wrongly lists "Marcel Janco" and "oil on canvas". The incorrect information has been highlighted using underlines.

### Example 4

**Title:**

Paa Joe/Exhibitions/Solo and group exhibitions

**Original question:**

Identify the venue that hosted the "Six Feet Under" exhibition during the years 2007 and 2008.

**Ground true answer:**

Deutsches Hygiene-Museum, Dresden

**Modified question:**

Describe the venue that hosted the "Six Feet Under" exhibition during the years 2007 and 2008.

**Misinformation:**

The "Six Feet Under" exhibition was hosted at the Kunstmuseum Bern in Switzerland during the years 2007 and 2008. The Kunstmuseum Bern is renowned for its extensive collection of fine art, ranging from the Middle Ages to contemporary works. The museum is situated in the Swiss capital, Bern, and is known for its modern architectural design that complements its historical setting. The "Six Feet Under" exhibition explored themes related to death and the afterlife, featuring a diverse array of artworks that engaged with these profound topics.

**Table 6:** Example 4 for query 4. In Example 4, the accurate venue is "Deutsches Hygiene-Museum, Dresden", but the misinformation mistakenly mentions "Kunstmuseum Bern". The incorrect information has been highlighted using underlines.

### Example 5

---

**Title:**

Clare Kenny/Selected exhibitions

**Original question:**

Which art space featured Clare Kenny's "If I was a Rich Girl" in 2019?

**Ground true answer:**

Kunst Raum Riehen

**Modified question:**

Identify the art space that showcased Clare Kenny's exhibition titled "If I was a Rich Girl" in the year 2019. Provide details about the venue and its significance in the art community if possible.

**Misinformation:**

Clare Kenny's exhibition titled "If I was a Rich Girl" was showcased at VITRINE in 2019. VITRINE is a contemporary art gallery located in London, known for its unique exhibition format that utilizes both a window gallery space and an interior gallery. This allows for art to be viewed from the street, making it accessible to a broader audience. VITRINE has gained a reputation for supporting emerging and mid-career artists, providing them with a platform to present innovative and experimental works. The gallery's commitment to showcasing diverse and thought-provoking art has made it a significant player in the contemporary art scene.

**Table 7:** Example 5 for query 4. Original testing process correctly names the venue as "Kunst Raum Riehen", in contrast to the misinformation's incorrect attribution to "VITRINE". The incorrect information has been highlighted using underlines.

### **Prompt for Multiple Choice Question-answer Pair Generation**

According to the following title and context, reply with {num\_of\_qa} multiple-choice questions and a statement based on the question.

## Title  
{title}

## Context (Only you can see this context)  
{context}

#### ## Requirement

- Each question should include 4 options.
- The context is only for you to generate the question. Therefore, your question should not redirect to this context, like "according to the provided context...".
- Your question should include enough information that can help the testee to understand the background.
- Your reply should follow the given JSON format.

#### ## Reply format

```
[  
  {  
    "question": "QUESTION CONTENT",  
    "options": ["A: ...", "B: ...", "C: ...", "D: ..."],  
    "statement": "STATEMENT OF THIS QUESTION.",  
    "answer": "CHOICE FROM THE OPTIONS. For example, A"  
  },  
  ...  
]
```

**Table 8:** The prompt for multiple choice question generation.

### **Prompt for Question Rephrasing**

According to the following title, context, and question, reply with {num\_of\_qa} rephrased questions and the corresponding answers. Your question should provide sufficient content to avoid ambiguity. You should reply with JSON format as follows:

## Title  
{title}

## Context (Only you can see this context)  
{context}

## Question  
{question}

#### ## Requirement

- Your question should have the same meaning as the provided question, only rephrased.
- The context is only for you to generate the question. Therefore, your question should not redirect to this context, like “according to the provided context..”.
- Your question should include enough information that can help the testee to understand the background.
- Your reply should follow the given JSON format.

#### ## Reply format

```
[  
  {  
    "question": "QUESTION CONTENT",  
    "options": ["A: ...", "B: ...", "C: ...", "D: ..."],  
    "statement": "STATEMENT OF THIS QUESTION.",  
    "answer": "CHOICE FROM THE OPTIONS. For example, A"  
  },  
  ...  
]
```

**Table 9:** The prompt for Question Rephrasing.

### Prompt for Analyzing Error Pattern

Given a context, a question, and its corresponding incorrect solution, generate a gerund phrase that thoroughly and precisely describes the **specific** skill or capability lacking that causes the error.

## Context  
{context}

## Question  
{question}

## Correct Solution  
{answer}

## Incorrect Solution  
{llm\_answer}

#### ## Requirement

- The incorrect Solution is provided by a testee who cannot access the context. Your answer should not mention that the skill is related to context information retrieval.
- The skill description should be an action-oriented gerund phrase that is **informative** and **detailed**.
- The phrase should refer to a **specific** skill or capability that comprehensively covers the key aspects of the solution, without including any context or specifics from the question or solution.
- Avoid unnecessary elements unrelated to the core capability.
- Please output **only a gerund phrase** describing the skill, with NO additional text.

**Table 10:** The prompt for Analyzing Error Pattern.

### Prompt for Testee Model

Given the topic: {topic}, answer the following question by choosing one option in Options:

Question:  
{que}

Options:  
{opts}

Your Answer (put your answer in \box{ }):

**Table 11:** The prompt for Testee Model.

**Prompt for Testee Model in Query 5**

Given the topic: {topic}, answer the following question by choosing one option from Options below:

Question:  
{question}

Retrieved Fact:  
{input}

Options:  
{options}

Answer:

**Table 12:** The prompt for Testee Model in Query 5.