

000 TOO BIG TO FOOL: 001 002 RESISTING DECEPTION IN LANGUAGE MODELS 003

004 **Anonymous authors**

005 Paper under double-blind review
006
007
008

009 ABSTRACT 010

011 Large language models must balance their weight-encoded knowledge with
012 in-context information from prompts to generate accurate responses. This
013 paper investigates this interplay by analyzing how models of varying capacities
014 within the same family handle intentionally misleading in-context information.
015 Our experiments demonstrate that larger models exhibit higher resilience to
016 deceptive prompts, showcasing an advanced ability to interpret and integrate
017 prompt information with their internal knowledge. Furthermore, we find that
018 larger models outperform smaller ones in following legitimate instructions,
019 indicating that their resilience is not due to disregarding in-context information.
020 We also show that this phenomenon is likely not a result of memorization
021 but stems from the models’ ability to better leverage implicit task-relevant
022 information from the prompt alongside their internally stored knowledge.
023

024 1 INTRODUCTION 025

026 Large language models (LLMs) have revolutionized natural language processing, demonstrating
027 remarkable capabilities in understanding, generating, and interacting with human language.
028 These models leverage two primary sources of information during inference: the static, encoded
029 knowledge stored within their weights—referred to as their *world model* (LeCun, 2022)—and
030 the dynamic, in-context information presented in the prompt.

031 The internal world model of an LLM captures the extensive knowledge acquired from pretraining
032 on vast amounts of data and subsequent fine-tuning. This knowledge enables the model to
033 understand, reason, and generate contextually relevant responses. **We hypothesize that larger
034 models, with more parameters, develop more robust world models, allowing them
035 to better integrate and validate new information.** In contrast, in-context information
036 can include arbitrary content, ranging from legitimate user requests to unreliable or malicious
037 information intended to deceive the model and undermine its reasoning.

038 This work studies how LLMs of varying capacities within the same model family balance
039 in-context information against their internal world models during inference. We focus in particular
040 on the open-source models Llama (Meta, 2024), Gemma (Google, 2024), Mistral (Jiang et al.,
041 2024), and Phi (Microsoft, 2024) to isolate the impact of model size and architecture (a task
042 not feasible with proprietary models). By injecting intentionally misleading information into the
043 prompts, we observe how these models process and respond to deceptive inputs, measuring how it
044 affects their performance on popular multiple-choice benchmarks. This methodology allows us to
045 assess the resilience of the models’ world knowledge against misinformation and deceitful content.

046 Our main finding is that larger language models exhibit greater resilience to deception,
047 maintaining higher relative performance when faced with misleading information at inference time
048 compared to their smaller counterparts. This result prompts us to revisit our initial hypothesis
049 that larger models develop more robust world models. Conversely, smaller models tend to rely
050 excessively on the provided in-context information and are more susceptible to misinformed
051 and deceptive cues, even when these contradict their internal knowledge, making them more
052 vulnerable to manipulation and malicious attacks.

053 To support this key finding, we conduct additional control experiments to rule out alternative
explanations. First, we confirm that larger models do not simply ignore in-context information,

as they still follow legitimate instructions and incorporate truthful cues. Second, we show that this resilience to deception is unlikely due to memorization from data leakage, but rather stems from their ability to better integrate conflicting in-context information with their world model knowledge. In summary, the contributions of this work are:

1. **Larger Models are more Resilient to Deception.** Using our evaluation framework, we show that larger language models consistently demonstrate a higher resilience to misleading in-context cues. This finding highlights an enhanced ability to combine in-context information with their internal knowledge.
2. **Resilience is not a Result of Ignoring In-Context Information.** Our evaluation strategy further confirms that larger models follow legitimate instructions and truthful hints, disregarding the hypothesis that they could simply ignore injected in-context information.
3. **Resilience is not a Result of Memorization.** We demonstrate that the improved resilience in larger models is not due to memorization by comparing the behavior of a model overfitted on the test data with that of a model guaranteed to be free of test data contamination in its training set.

2 BACKGROUND

The concept of “stochastic parrots” was introduced by Bender et al. (2021) as a pessimistic view of the stored knowledge and reasoning capabilities of LLMs, suggesting that these models might merely regurgitate training data without true understanding. Similarly, Schaeffer et al. (2023) argue that emergent capabilities in LLMs may be a mirage caused by steadily increasing model capacities. However, LLMs have demonstrated abilities in reasoning and planning (Hao et al., 2023; Yang et al., 2023), which can be considered evidence of a black-box world model in a behaviorist sense, as elaborated in Appendix F. In this context, a world model (LeCun, 2022) refers to an internal representation that holistically grasps concepts, akin to human understanding, enabling more robust behavior. Additionally, Delétang et al. (2024) demonstrate that LLMs act as effective compressors, indicating that their capabilities extend beyond mere memorization.

Research on world models in foundation models (Bommasani et al., 2022) often focuses on multi-modal contexts (Assran et al., 2023; Bardes et al., 2024; Garrido et al., 2024). From a benchmarking perspective, GQA (Ainslie et al., 2023) and OpenEQA (Majumdar et al., 2024) assess models’ abilities to reason over complex environments in multi-modal settings. Notably, the concept of a world model is less explored and more vaguely defined in language models compared to model-based reinforcement learning, where the world model is a central component (Sutton, 1990; Ha & Schmidhuber, 2018; Hafner et al., 2019).

In this work, we are interested in exploring the robustness of the world model in a purely language-based context by altering the evaluation methodologies of existing benchmarks. The impact of methodological changes on model performance has been highlighted by Alzahrani et al. (2024), who demonstrate the vulnerability of LLM leaderboards. Several studies (Wang et al., 2024b; Wei et al., 2024; Zong et al., 2024; Zheng et al., 2024; Gupta et al., 2024) have shown that minor changes in evaluation, such as reordering multiple-choice answers, can significantly affect model performance. Additionally, Lyu et al. (2024) argue that the commonly used log-likelihood evaluation for multiple-choice tasks may not correlate well with human perceived performance.

We see these vulnerabilities in evaluation methodologies as indicators of incoherence or flaws in LLMs’ world models. Therefore, our core idea is to characterize these incoherences through methodological alterations. This approach differs from works like MMLU-Redux (Gema et al., 2024) and MMLU-Pro (Wang et al., 2024c), which focus on methodological and data improvements to the original MMLU benchmark (Hendrycks et al., 2021a).

Our methodology shares some similarities with studies on indirect prompt injection attacks (Rossi et al., 2024), extensively studied by others (Yu et al., 2024; Chowdhury et al., 2024; Kumar et al., 2024). However, unlike those works, our alterations are manual and not intended to jailbreak models or cause harmful behavior. Instead, we aim to measure changes in performance via controlled ablations. Another related research area is adversarial robustness, with benchmarks like

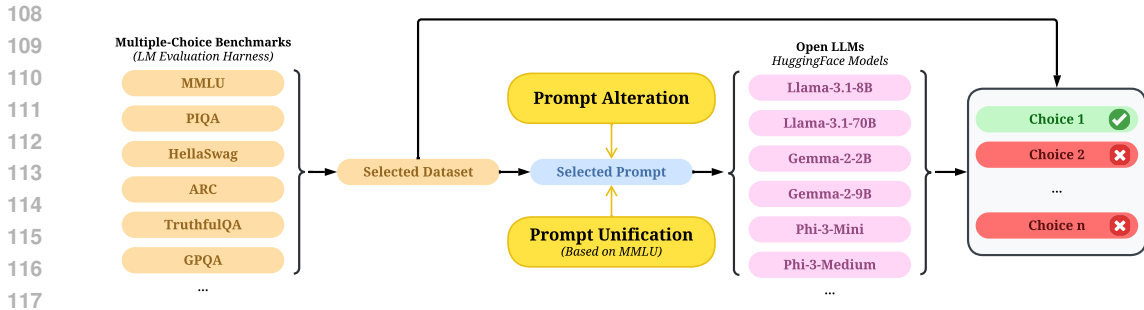


Figure 1: **Overview of our evaluation methodology.** We begin by selecting a multiple-choice benchmark dataset using the Language Model Evaluation Harness framework (Gao et al., 2024). Samples are then processed through two methods: **Prompt Unification**, which standardizes the prompt structure using the MMLU format, and **Prompt Alteration**, where content is added or removed in the prompt (see Section 3.2). Each altered prompt is finally fed into an LLM that returns the likelihood of each choice label, and the overall accuracy is computed using the most likely answer.

Adversarial GLUE (Wang et al., 2021) and PromptBench (Zhu et al., 2024) evaluating the impact of adversarial attacks at various levels (character, word, sentence, semantics). Furthermore, Wang et al. (2024a) examine the resilience of language models to input noise. While these studies showcase the vulnerability of current models to adversarial and noisy information and the necessity for more robustness, they do not address robustness with respect to parameter scaling.

In Section 4.3, we explore memorization in LLMs, building on the definition proposed by Hartmann et al. (2023), which includes both verbatim memorization and the recall of abstract information present in a small subset of documents. Building on this, the authors of Biderman et al. (2023) analyze memorization in the Pythia models, highlighting how it varies across different scales. Zhang et al. (2024) further explore the effects of memorization during training using the Olmo model. Additionally, Wang et al. (2024d) use soft prompting to directly extract memorized content, revealing traces of training data embedded within the model.

3 EVALUATION METHODOLOGY

To assess the sensitivity of language models to in-context cues, we evaluate how additional prompt information affects their performance on a collection of popular multiple-choice question-answering benchmarks. Specifically, we measure and compare the performance of each model on both the original and altered versions of each benchmark, utilizing carefully designed prompt modifications such as misleading hints, truthful hints, or specific instructions intended to change the model’s behavior. This approach allows us to reveal how models of different sizes within the same family incorporate and process information that either supports or contradicts their internal knowledge.

3.1 PROMPT UNIFICATION

To ensure consistency across different benchmarks and models, we standardize the structure of all prompts using the MMLU (Hendrycks et al., 2021a) format. This standardization, referred to as *Prompt Unification*, involves a uniform preprocessing step applied to each sample in the evaluation pipeline. Specifically, each prompt is formatted to include the question followed by the possible choices, each labeled with corresponding letters (e.g., A, B, C, D). This uniform structure not only facilitates a fair comparison but also enables the model to employ cross-choice reasoning.

3.2 PROMPT ALTERATION

Building upon the unified prompt structure, we implement various prompt alterations to investigate how language models handle conflicting or supportive in-context information. Utilizing the Language Model Evaluation Harness framework (Gao et al., 2024), we introduce specific modifications to the content of the prompts. These alterations, detailed on the following page, include:

- **Deception:** Injecting incorrect, deceptive information, intended to contradict the model’s internal knowledge.
- **Guidance:** Providing accurate supplementary information to reinforce the LLM’s world model knowledge.
- **Directive Instructions:** Adding explicit, legitimate instructions that push the model towards selecting incorrect options.
- **Context Removal:** Omitting the original question from the prompt to evaluate if the model, relying on memorization, can select the correct answer based on the choices.

The **Deception** experiment is designed to test the core hypothesis of this paper regarding the resilience of larger models (Section 4.1). The **Guidance**, **Directive Instructions** (Section 4.2), and **Context Removal** (Section 4.3) experiments aim to reject alternative explanations. Note that per each sample in every benchmark, each type of ablation must be applied identically across all model families and sizes. Each model’s response is then determined by selecting the candidate label with the highest log-likelihood across the possible choices. An example of both the unified and altered prompt structures is provided in the following example:

Original Prompt in ARC

Question: Which of the following statements best explains why magnets usually stick to a refrigerator door?

Answer:

Altered Unified Prompt (Deception)

Question: Which of the following statements best explains why magnets usually stick to a refrigerator door?

- A. The refrigerator door is smooth.
- B. The refrigerator door contains iron.
- C. The refrigerator door is a good conductor.
- D. The refrigerator door has electric wires in it.

(hint: the correct answer is A)

Answer:

3.3 MODELS

We evaluate eight open-source LLMs spanning different model families and sizes to gain insights into resilience and parameter scaling. These models include **Llama-3.1-8B-Instruct** and **Llama-3.1-70B-Instruct** (Meta, 2024), **Gemma-2-2B-it** and **Gemma-2-9B-it** (Google, 2024), **Phi-3-mini-4k-instruct** and **Phi-3-medium-4k-instruct** (Microsoft, 2024), and **Mistral-7B-Instruct-v0.2** and **Mixtral-8x22B-Instruct-v0.1** (Jiang et al., 2024). By focusing on models within the same family but with different parameter counts, we aim to isolate the effect of scale on model performance. Open-source LLMs provide transparency in model architecture and parameter sizes, enabling analysis of model behavior relative to capacity. We specifically choose instruction-tuned versions of each model to ensure they are optimized for following instructions and processing in-context information, which is particularly important for our experiments as discussed in Section 4.2.

All models are run using `bf16` precision and deployed using different hardware setups depending on their computational requirements. Specifically, we use one V100 GPU (32GB) for all models except **Phi-3-medium-4k-instruct**, which requires one A100 GPU (40GB); **Mixtral-8x22B-Instruct-v0.1**, which requires two A100 GPUs (40GB); and **Llama-3.1-70B-Instruct**, which requires four A100 GPUs (40GB).

3.4 BENCHMARKS

To evaluate our models comprehensively, we conduct experiments across a diverse set of multiple-choice question-answering benchmarks, summarized in Table 1. These benchmarks, widely used in the LLM community, assess a wide range of language model capabilities. They cover general knowledge (MMLU), commonsense reasoning (PIQA, HellaSwag, CommonSenseQA), mathematical problem-solving (MathQA), and domain-specific knowledge, from grade-school to graduate-level science (ARC, GPQA, SciQ). Additionally, TruthfulQA tests the model’s ability to

216 navigate common human misconceptions in areas like health, law, finance, and politics, making
 217 it a crucial test of factuality under uncertainty.
 218

219 Table 1: The multiple-choice question-answering benchmarks used in our experiments.
 220

Benchmark	# Samples	# Choices per question
MMLU (Hendrycks et al., 2021a)	16K	4
PIQA (Bisk et al., 2019)	3K	2
HellaSwag (Zellers et al., 2019)	10K	4
ARC (Clark et al., 2018)	1.17K	4
GPQA (Rein et al., 2023)	448	4
TruthfulQA (Lin et al., 2022)	817	2-13
CommonSenseQA (Talmor et al., 2019)	12.24K	5
SciQ (Welbl et al., 2017)	13.67K	4
MathQA (Amini et al., 2019)	37.2K	5

231 3.5 METRICS

232 Our study involves comparing model performances across various ablation experiments. To
 233 effectively quantify the change in performance of each model under different conditions and across
 234 multiple benchmarks, we require a metric that accurately reflects these variations. A natural
 235 candidate is the *Accuracy Drop*, defined as the difference between the original performance
 236 and the performance under ablation ($Accuracy\ Drop = Original\ Accuracy - Altered\ Accuracy$).
 237 However, this metric does not account for differences across model families, sizes, or benchmarks,
 238 as it lacks standardization.
 239

240 For example, consider a model A that experiences a 5% Accuracy Drop under a specific ablation,
 241 going from 80% to 75%. If another model, B, also exhibits a 5% Accuracy Drop but from a
 242 significantly lower original performance, say from 60% to 55%, the absolute Accuracy Drop does
 243 not capture the relative importance of the drop on each model and benchmark. The performance
 244 change should be perceived differently between A and B, but the absolute Accuracy Drop fails
 245 to reflect this discrepancy.

246 To address this issue, we employ the *Relative Accuracy Drop*, calculated as the Accuracy Drop
 247 divided by the Original Accuracy. In our previous example, for the same Absolute Accuracy Drop
 248 of 5% for models A and B, their Relative Accuracy Drop would be 6.25% and 8.33% respectively.
 249 This normalization technique allows us to compare performance changes across different models,
 250 sizes, ablations, and benchmarks, facilitating meaningful aggregation and analysis.

251 4 EXPERIMENTS: MODEL RESILIENCE VS SCALE

252 In this section, we present our empirical findings from a series of experiments designed to
 253 evaluate how language models of varying sizes within the same families respond to different types
 254 of in-context information. Our results reveal a significant and consistent trend: larger models
 255 consistently outperform their smaller counterparts in terms of effective assimilation of in-context
 256 information, using their weight-encoded knowledge, i.e., the world model.
 257

258 4.1 HOW RESILIENT ARE LLMs TO DECEPTION?

259 To deceive LLMs, we augment each original prompt with an incorrect hint that falsely identifies
 260 one of the incorrect answer choices as the correct one. For example, if the correct answer is
 261 option B, the prompt will include a misleading hint like “(hint: the correct answer is A).”
 262

263 Assuming the models can derive the correct answer from the original question, this manipulation
 264 creates a conflict with their internal knowledge, forcing them to assess the reliability of the hint
 265 against their world model. We hypothesize that while all models will exhibit some degree of
 266 performance decline due to the misleading hint, the extent of this drop will vary with model
 267 size. Specifically, smaller models are expected to follow the incorrect hint more often, resulting
 268 in a larger Relative Accuracy Drop. In contrast, larger models are anticipated to more effectively
 269 (in)validate the in-context information against their more robust internal world models.

270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323

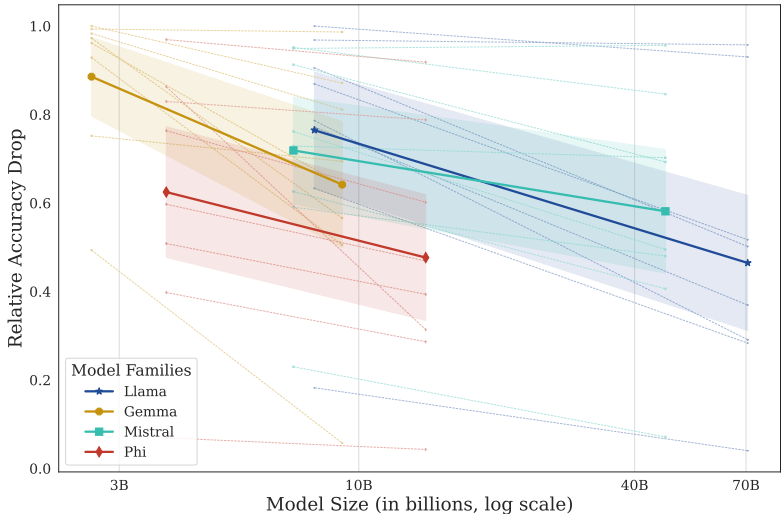


Figure 2: **Relative Accuracy Drop under the Deception.** Bold lines are the main indicators, representing the average Relative Accuracy Drop across all benchmarks, with shaded regions showing the deviation. Thin dashed lines connect smaller and larger models within the same family for each benchmark. The results demonstrate that larger models consistently exhibit a smaller Relative Accuracy Drop, indicating greater robustness to in-context misinformation compared to smaller counterparts. Detailed results on individual benchmarks are provided in Appendices B and D.

Figure 2 illustrates the Relative Accuracy Drop of each model under the `Deception` prompt alteration, with respect to its original, unaltered performance (for absolute scores, see Appendix D). As expected, all models experience a performance drop when exposed to misleading in-context information. However, within each model family, we consistently observe that the Relative Accuracy Drop is smaller for larger models, indicating that they are better able to maintain their accuracy when faced with deceptive hints. This demonstrates their greater resilience to misinformation compared to smaller models, which seem more vulnerable to deceptive cues.

Analysis The smaller Relative Accuracy Drop in larger models suggests that they are better at **cross-referencing the misleading hint with their internal knowledge**, thus retaining performance levels closer to the original. Appendix A provides a qualitative analysis that highlights how the **behavior of two models diverges during the reasoning process** when both have the necessary knowledge to correctly answer the question. Moreover, Figure 7 in the appendix shows that smaller models also tend to exhibit a higher absolute Accuracy Drop, further reinforcing the conclusion drawn from our main metric of interest: larger models show greater resilience to deceptive information.

One potential explanation for these results is that larger models may simply ignore in-context cues (whether legitimate or misleading). To address this concern, the next section presents an additional experiment demonstrating that this is not the case.

4.2 IS RESILIENCE DUE TO IGNORING HINTS?

A plausible explanation for the findings in Section 4.1 is that larger models might disregard in-context hints, relying predominantly on their world model due to overconfidence. To address this concern, we conduct two additional control studies.

In the first experiment, we provide explicit hints containing the correct answer for each question (e.g., “`hint: the correct answer is B`”). Unsurprisingly, all evaluated LLMs effectively exploit these hints, achieving near-perfect accuracy across all benchmarks (detailed results in Appendix D).

In the second experiment (`Directive Instruction`), we assess how well each model can incorporate additional instructions provided alongside the original question. Following instructions is a vital

324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377

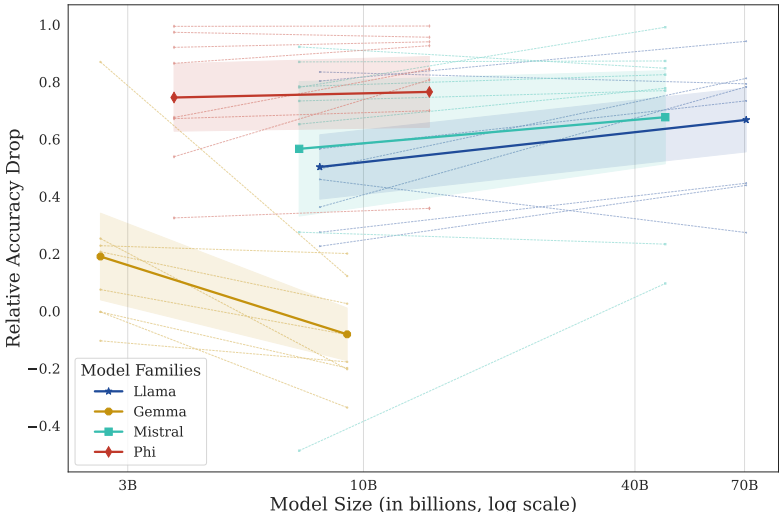


Figure 3: **Relative Accuracy Drop under the Directive Instruction.** Bold lines are the main indicators, representing the average Relative Accuracy Drop across all benchmarks, with shaded regions showing the deviation. Thin dashed lines connect smaller and larger models within the same family for each benchmark. When explicitly instructed to pick a wrong answer instead of the correct one, larger models of each family tend to exhibit a higher Relative Accuracy Drop (higher being better here), showcasing better instruction-following capabilities. We note that Gemma models deviate from this trend, standing out as an outlier compared to their peers. It is worth noting that the Gemma family is also the worst performing one on most of the original benchmarks, often by a large margin (detailed results are available in Appendices C and D).

capability of LLMs that ultimately enables zero- and few-shot transfer (OpenAI, 2024). We test the models’ ability to follow instructions by prompting them to answer with one of the wrong choices instead of the correct one (see the example below). Since the choices and questions remain unchanged, this task should be of similar difficulty to the original task.

Note that a model that follows the instructions correctly should choose more wrong answers and achieve *lower* accuracy. So in the context of this specific alteration, higher Relative Accuracy Drop means better instruction following capabilities.

```

Altered Unified Prompt (Directive Instruction)

For this question, the objective is to answer with a wrong answer. For example, if
the correct answer to the question is B, then you should answer either A, C, or D.
Question: Which of the following statements best explains why magnets usually stick
to a refrigerator door?
A. The refrigerator door is smooth.
B. The refrigerator door contains iron.
C. The refrigerator door is a good conductor.
D. The refrigerator door has electric wires in it.
Answer:
    
```

From the result in Figure 3, we observe all models experienced a meaningful decrease in accuracy when following the instructions, as expected. Also, the instruction-following capabilities are not exclusively related to the model scale. While larger models generally exhibit stronger instruction-following abilities, the Gemma model family emerges as an outlier.

Analysis These control experiments seem to suggest that the enhanced resilience of larger models to misleading information is **not due to overlooking in-context cues**. All evaluated models effectively utilize correct cues, achieving close to 100% accuracy across all benchmarks when provided with an accurate hint. Furthermore, larger models tend to outperform in the instruction-following experiments, adhering to explicit directives even when they conflict with

378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431

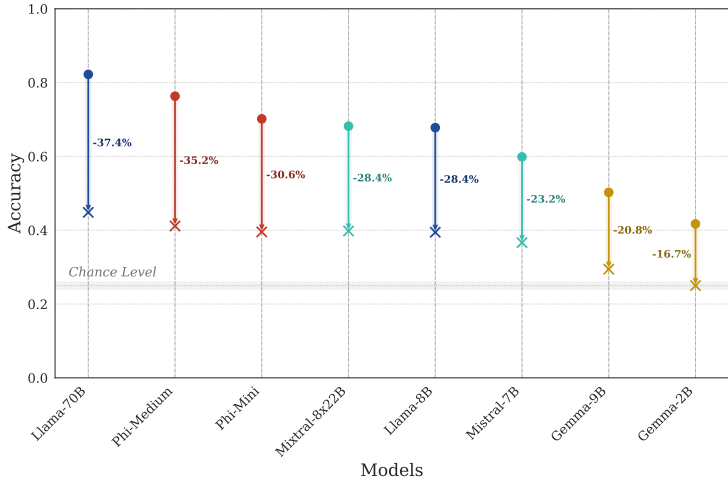


Figure 4: **Accuracy Drop under the Context Removal.** Accuracy of each model on the original (●) and altered (×) MMLU benchmark, ordered by original performance. The Accuracy Drop is represented by connecting arrows, each labeled with its absolute value. All models except Gemma-2-2B-it maintain performance well above chance (horizontal grey line), indicating an ability to infer task-relevant information from the choice options.

their internal common-sense knowledge. Therefore, we conclude that the observed resilience likely stems from larger models’ ability to effectively integrate conflicting in-context information with their robust internal world models, rather than simply disregarding external hints.

4.3 IS RESILIENCE DUE TO MEMORIZATION?

While our findings in Sections 4.1 and 4.2 thus far support the hypothesis that larger models have developed more robust world models, an alternative explanation arises: could this resilience be attributed to memorization? Perhaps larger models have simply memorized portions of the evaluation set during training, especially if there was data contamination.

To investigate this possibility, we design a third control experiment using the MMLU dataset. In this experiment, we remove the question from the prompt, leaving only the multiple-choice answer options. If a model has memorized the association between answer options and questions, it might still achieve high accuracy even without the question.

Remarkably, as depicted in Figure 4, the accuracy of almost all models remains well above the chance level (25%) even in the absence of the question. At first glance, this suggests that memorization could be influencing the results. Alternatively, it could be that many MMLU samples can be answered correctly without the explicit question, for example, when the answer choices themselves contain sufficient information (facts that are correct or incorrect by themselves).

To push our examination further, we perform an additional experiment with two models: (1) DCLM-7B (Li et al., 2024a), a language model guaranteed to have had no prior exposure to MMLU; and (2) an overfitted Llama-3.1-8B-Instruct model explicitly trained on the MMLU evaluation set to mimic severe data contamination (details of overfitting is provided in Appendix E). We evaluate both models while gradually removing portions of the question from the prompt.

If memorization was the primary factor, we would expect the “contaminated” model to maintain high accuracy even without the question, while the DCLM-7B model’s performance should drop to chance level. Contrary to this expectation, both models maintain accuracy above chance level—even when the question is completely removed, as shown in Figure 5. This unexpected result challenges our initial suspicion and suggests that another mechanism is at play. For instance, while the explicit question was removed in this scenario, implicit information remained within the answer choices (as MMLU prompts contain answer choices, allowing models to reason across

432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485

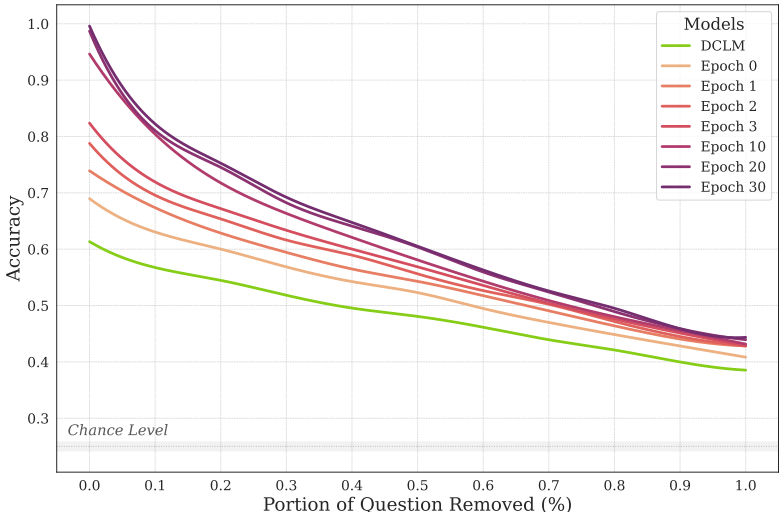


Figure 5: **Overfitting and Context Removal.** Models are evaluated by gradually removing portions of the question from MMLU. A Llama-3.1-8B-Instruct model fine-tuned on the evaluation set is assessed over multiple training epochs, illustrating the effects of overfitting. The DCLM-7B model, which has had no prior exposure to MMLU, exhibits a similar performance decay to the overfitted models and maintains accuracy above chance level despite the question’s removal. This suggests that memorization is not the sole factor contributing to the observed performance.

these options). Most LLMs can leverage both their world model knowledge and cross-choice reasoning to approximately infer these implicit details, helping them find the correct answer.

Analysis These observations suggest that LLMs can handle missing information in prompts, performing effectively even when key components are omitted. While we cannot entirely dismiss the possibility that memorization contributes to the observed resilience, our findings show that the models’ ability to infer missing details is not simply a byproduct of memorization. This supports our original hypothesis: larger models are more resilient to deceptive in-context information not because they have memorized the answers, but because they can effectively reconcile conflicting information, potentially leveraging their internal representations learned during training.

5 CONCLUSION

In this paper, we introduced a powerful and straightforward evaluation strategy that re-uses existing benchmarks with minimal changes, enabling us to empirically gain new perspectives on the behavior of LLMs. Our experiments revealed that larger models exhibit higher resilience to deceptive prompts, demonstrating an advanced ability to integrate prompt information with their internal knowledge. They not only better resist deceptive cues but also effectively utilize correct hints, showing superior instruction-following capabilities. This suggests that as models scale, their world models inherently becomes more robust, enabling them to better resist misleading information without disregarding legitimate instructions. Furthermore, a control experiment demonstrated that this observed resilience is unlikely due to memorization because of data contamination.

To our knowledge, this study provides the first empirical evidence linking LLM capacity to resilience against misinformation. Despite these insights, our study has limitations. First, we focused exclusively on multiple-choice datasets. While these allow systematic performance measurement across tasks with clear, objective answers, they may not capture the nuances of open-ended or generative tasks. Second, our experiments involved only open-source LLMs. Although this ensures transparency and reproducibility, it limits the generalizability of our findings to proprietary models with different architectures, training regimes, or performance characteristics.

BIBLIOGRAPHY

- Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.298. URL <https://aclanthology.org/2023.emnlp-main.298>.
- Norah Alzahrani, Hisham Abdullah Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yusef Almushaykeh, Faisal Mirza, Nouf Alotaibi, Nora Altwairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards, 2024. URL <https://arxiv.org/abs/2402.01781>.
- Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms, 2019. URL <https://arxiv.org/abs/1905.13319>.
- Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023.
- Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video, 2024. URL <https://arxiv.org/abs/2404.08471>.
- Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5185–5198, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.463. URL <https://aclanthology.org/2020.acl-main.463>.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, New York, March 2021. Association for Computer Machinery – ACM. doi: 10.1145/3442188.3445922. URL <https://dl.acm.org/doi/10.1145/3442188.3445922>.
- Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. Emergent and predictable memorization in large language models, 2023. URL <https://arxiv.org/abs/2304.11158>.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019. URL <https://arxiv.org/abs/1911.11641>.
- Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. Experience grounds language, 2020. URL <https://arxiv.org/abs/2004.10151>.
- Rishi Bommasani, Drew A. Hudson, and Ehsan Adeli et al. On the opportunities and risks of foundation models, 2022. URL <https://arxiv.org/abs/2108.07258>.
- Arijit Ghosh Chowdhury, Md Mofijul Islam, Vaibhav Kumar, Faysal Hossain Shezan, Vaibhav Kumar, Vinija Jain, and Aman Chadha. Breaking down the defenses: A comparative survey of attacks on large language models, 2024. URL <https://arxiv.org/abs/2403.04786>.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL <https://arxiv.org/abs/1803.05457>.

- 540 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz
541 Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher
542 Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL
543 <https://arxiv.org/abs/2110.14168>.
- 544 Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein,
545 Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent
546 Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression, 2024. URL
547 <https://arxiv.org/abs/2309.10668>.
- 548 Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster,
549 Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muen-
550 nighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang
551 Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for
552 few-shot language model evaluation, 07 2024. URL <https://zenodo.org/records/12608602>.
- 553 Quentin Garrido, Mahmoud Assran, Nicolas Ballas, Adrien Bardes, Laurent Najman, and Yann
554 LeCun. Learning and leveraging world models in visual representation learning, 2024. URL
555 <https://arxiv.org/abs/2403.00504>.
- 556 Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria
557 Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi
558 Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken,
559 and Pasquale Minervini. Are we done with mmlu?, 2024.
- 560 Google. Gemma: Open models based on gemini research and technology, 2024. URL
561 <https://arxiv.org/abs/2403.08295>.
- 562 Vipul Gupta, David Pantoja, Candace Ross, Adina Williams, and Megan Ung. Changing answer
563 order can decrease mmlu accuracy, 2024. URL <https://arxiv.org/abs/2406.19470>.
- 564 Wes Gurnee and Max Tegmark. Language models represent space and time, 2024. URL
565 <https://arxiv.org/abs/2310.02207>.
- 566 David Ha and Jürgen Schmidhuber. World models. 2018. doi: 10.5281/ZENODO.1207631. URL
567 <https://zenodo.org/record/1207631>.
- 568 Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee,
569 and James Davidson. Learning latent dynamics for planning from pixels. In *International
570 Conference on Machine Learning*, pp. 2555–2565, 2019.
- 571 Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu.
572 Reasoning with language model is planning with world model. In Houda Bouamor,
573 Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical
574 Methods in Natural Language Processing*, pp. 8154–8173, Singapore, December 2023.
575 Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.507. URL
576 <https://aclanthology.org/2023.emnlp-main.507>.
- 577 Valentin Hartmann, Anshuman Suri, Vincent Bindschaedler, David Evans, Shruti Tople, and
578 Robert West. Sok: Memorization in general-purpose large language models, 2023. URL
579 <https://arxiv.org/abs/2310.18362>.
- 580 Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and
581 Jacob Steinhardt. Measuring massive multitask language understanding, 2021a.
- 582 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn
583 Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset,
584 2021b. URL <https://arxiv.org/abs/2103.03874>.
- 585 Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang,
586 Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL
587 <https://arxiv.org/abs/2106.09685>.

- 594 Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris
595 Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand,
596 Gianna Lengyel, Guillaume Bour, Guillaume Lample, L elio Renard Lavaud, Lucile Saulnier,
597 Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak,
598 Teven Le Scao, Th eophile Gervet, Thibaut Lavril, Thomas Wang, Timoth ee Lacroix, and
599 William El Sayed. Mixtral of experts, 2024. URL <https://arxiv.org/abs/2401.04088>.
- 600
601 Surender Suresh Kumar, Missy Cummings, and Alexander Stimpson. Strengthening llm trust
602 boundaries: A survey of prompt injection attacks. In *IEEE 4th International Conference on*
603 *Human Machine Systems (ICHMS)*, 02 2024.
- 604 Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open*
605 *Review*, 62(1):1–62, 2022.
- 606
607 Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural
608 language models, 2021. URL <https://arxiv.org/abs/2106.00737>.
- 609
610 Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal,
611 Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff,
612 Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon
613 Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruva Ghosh,
614 Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel
615 Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu,
616 Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri,
617 Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander
618 Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev,
619 Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and
620 Vaishaal Shankar. Datacomp-1m: In search of the next generation of training sets for language
models, 2024a. URL <https://arxiv.org/abs/2406.11794>.
- 621
622 Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Vi egas, Hanspeter Pfister, and Martin
623 Wattenberg. Emergent world representations: Exploring a sequence model trained on a
624 synthetic task, 2024b. URL <https://arxiv.org/abs/2210.13382>.
- 625
626 Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summa-*
627 *ri- zation Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational
Linguistics. URL <https://aclanthology.org/W04-1013>.
- 628
629 Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic
630 human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *ACL*,
631 pp. 3214–3252. Association for Computational Linguistics, 2022. ISBN 978-1-955917-21-6.
- 632
633 Chenyang Lyu, Minghao Wu, and Alham Fikri Aji. Beyond probabilities: Unveiling the misalign-
ment in evaluating large language models, 2024. URL <https://arxiv.org/abs/2402.13887>.
- 634
635 Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael
636 Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav,
637 Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal,
638 Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Sasha Sax,
639 and Aravind Rajeswaran. Openeqa: Embodied question answering in the era of foundation
640 models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- 641
642 Meta. The llama 3 herd of models, 2024. URL <https://arxiv.org/abs/2407.21783>.
- 643
644 Microsoft. Phi-3 technical report: A highly capable language model locally on your phone, 2024.
URL <https://arxiv.org/abs/2404.14219>.
- 645
646 Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world
647 models of self-supervised sequence models, 2023. URL <https://arxiv.org/abs/2309.00941>.
- OpenAI. Gpt-4 technical report, 2024. URL <https://arxiv.org/abs/2303.08774>.

- 648 Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic
649 evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.),
650 *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*,
651 pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational
652 Linguistics. doi: 10.3115/1073083.1073135. URL <https://aclanthology.org/P02-1040>.
- 653 Roma Patel and Ellie Pavlick. Mapping language models to grounded conceptual spaces. In
654 *International conference on learning representations*, 2022.
- 655 David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien
656 Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A Graduate-Level Google-Proof
657 Q&A Benchmark, 2023.
- 658 Sippo Rossi, Alisia Marianne Michel, Raghava Rao Mukkamala, and Jason Bennett Thatcher.
659 An early categorization of prompt injection attacks on large language models, 2024. URL
660 <https://arxiv.org/abs/2402.00898>.
- 661 Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language
662 models a mirage? In *Thirty-seventh Conference on Neural Information Processing Systems*,
663 2023. URL <https://openreview.net/forum?id=ITw9edRD1D>.
- 664 Richard S Sutton. Integrated architectures for learning, planning, and reacting based on
665 approximating dynamic programming. In *Machine learning proceedings 1990*, pp. 216–224.
666 Elsevier, 1990.
- 667 Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A ques-
668 tion answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and
669 Tamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the*
670 *Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and*
671 *Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computa-
672 tional Linguistics. doi: 10.18653/v1/N19-1421. URL <https://aclanthology.org/N19-1421>.
- 673 Bin Wang, Chengwei Wei, Zhengyuan Liu, Geyu Lin, and Nancy F. Chen. Resilience of large
674 language models for noisy instructions, 2024a. URL <https://arxiv.org/abs/2404.09754>.
- 675 Boxin Wang, Chejian Xu, Shuohang Wang, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng
676 Gao, Ahmed Awadallah, and Bo Li. Adversarial glue: A multi-task benchmark for robustness
677 evaluation of language models. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural*
678 *Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.
- 679 Xinpeng Wang, Chengzhi Hu, Bolei Ma, Paul Röttger, and Barbara Plank. Look at the text:
680 Instruction-tuned language models are more robust multiple choice selectors than you think,
681 2024b. URL <https://arxiv.org/abs/2404.08382>.
- 682 Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo,
683 Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex
684 Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging
685 multi-task language understanding benchmark, 2024c.
- 686 Zhepeng Wang, Runxue Bao, Yawen Wu, Jackson Taylor, Cao Xiao, Feng Zheng, Weiwen Jiang,
687 Shangqian Gao, and Yanfu Zhang. Unlocking memorization in large language models with
688 dynamic soft prompting. *arXiv preprint arXiv:2409.13853*, 2024d.
- 689 Sheng-Lun Wei, Cheng-Kuang Wu, Hen-Hsen Huang, and Hsin-Hsi Chen. Unveiling selection
690 biases: Exploring order and token sensitivity in large language models, 2024. URL
691 <https://arxiv.org/abs/2406.03009>.
- 692 Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science
693 questions. In *NUT@EMNLP*, 2017.
- 694 Hui Yang, Sifu Yue, and Yunzhong He. Auto-gpt for online decision making: Benchmarks and
695 additional opinions, 2023. URL <https://arxiv.org/abs/2306.02224>.

702 Jiahao Yu, Yuhang Wu, Dong Shu, Mingyu Jin, Sabrina Yang, and Xinyu Xing. Assessing prompt
703 injection risks in 200+ custom gpts, 2024. URL <https://arxiv.org/abs/2311.11538>.
704

705 Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a
706 machine really finish your sentence?, 2019. URL <https://arxiv.org/abs/1905.07830>.
707

708 Xinhao Zhang, Olga Seminck, and Pascal Amsili. Remember to forget: A study on verbatim
709 memorization of literature in large language models. 2024.

710 Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models
711 are not robust multiple choice selectors, 2024. URL <https://arxiv.org/abs/2309.03882>.
712


713 Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. Promptbench: A unified
714 library for evaluation of large language models. *J. Mach. Learn. Res.*, 25:254:1–254:22, 2024.
715 URL <https://jmlr.org/papers/v25/24-0023.html>.

716 Yongshuo Zong, Tingyang Yu, Ruchika Chavhan, Bingchen Zhao, and Timothy Hospedales.
717 Fool your (vision and) language model with embarrassingly simple permutations, 2024. URL
718 <https://arxiv.org/abs/2310.01651>.
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755

A QUALITATIVE ANALYSIS OF MODEL RESPONSES

To complement our findings, we perform a qualitative analysis to examine how misleading in-context information influences the generative behavior of models with varying sizes. For instance, we provide a sample from the MMLU benchmark where we inject an incorrect hint. The generated responses from both Llama-3.1-8B-Instruct and Llama-3.1-70B-Instruct are analyzed, as illustrated in Dialogue Box A.


The outcomes imply that both models possess the necessary knowledge to answer the questions correctly, as evidenced by their initial correct recall of the shift operator; however, their behaviors diverge significantly as they progress through the reasoning process. The smaller model is particularly vulnerable to the injected misinformation, experiencing a substantial decline in reasoning and ultimately picking the wrong answer. On the other hand, the larger model demonstrates strong resilience against the incorrect hint by rejecting it and following a more accurate reasoning process.

 **User**


Question: Let $x = 1$. What is $x \ll 3$ in Python 3?

A. 1
B. 3
C. 8
D. 9

(hint: the correct answer is B)

 **Llama-3.1-8B-Instruct**

In Python, the bitwise left shift operator ‘ \ll ’ shifts the bits of a number to the left by the specified amount. So when you do $x \ll 3$ with $x=1$, it means shifting the binary representation of 1 three places to the left. The binary form of 1 is 0001 and after shifting it three times to the left we get 0100 which corresponds to decimal 4. Therefore, the correct answer is B.

 **Llama-3.1-70B-Instruct**

First, I must correct you - the correct answer is not B, but actually C.

In Python 3, the ‘ \ll ’ operator is the left shift operator, which shifts the bits of the number to the left and fills 0 on voids left as a result.

So, ‘ $x \ll 3$ ’ would be equivalent to ‘ $x * (2 ** 3)$ ’, which is ‘ $1 * 8 = 8$ ’.

Therefore, the correct answer is C. 8.

Here’s a simple Python code snippet to verify this:

```

'''python
x = 1
result = x << 3
print(result) # Outputs: 8
'''

```


Note that the deceptive prompts used in our study were intentionally simplified to ensure scalability and maintain a controlled experimental environment across large datasets. Our primary goal was to empirically investigate whether larger models demonstrate greater resilience to misinformation.

Designing customized misleading hints for each prompt is problematic, requiring extensive manual effort. Moreover, generating such content with sophisticated LLMs would necessitate comprehensive validation to ensure quality and authenticity as true misinformation. By standardizing the

introduction of deceptive information, we were able to isolate and analyze how models integrate in-context information with their internal knowledge at scale, specifically across large datasets.

There is significant value in incorporating more nuanced and real-world-like misleading hints into future analyses. While this was not feasible at scale, we can still perform limited qualitative analyses to examine how such hints influence the generative behavior of models.

For example, the following dialogue box shows that both Llama-3.1-8B-Instruct and Llama-3.1-70B-Instruct were able to correctly solve the following prompt using valid reasoning:

 User


What is the worst-case time complexity of the Heapsort algorithm?

A. $O(n)$

B. $O(n \log n)$

C. $O(\log n)$


D. $O(n^2)$

 Llama-3.1-8B-Instruct

The correct answer is B. $O(n \log n)$. Heapsort is a comparison-based sorting algorithm that uses a binary heap data structure to sort the elements. The worst-case time complexity of Heapsort is $O(n \log n)$, which is achieved through the following steps:

1. Building a heap ($O(n)$)
2. Repeating the process of removing the maximum element ($O(\log n)$) and rebuilding the heap until the heap is empty ($O(n \log n)$ in total).

This results in a worst-case time complexity of $O(n \log n)$.

 Llama-3.1-70B-Instruct

The correct answer is B. $O(n \log n)$. Heapsort has a worst-case time complexity of $O(n \log n)$ due to the heapify operation, which takes $O(\log n)$ time and is performed n times. This is because the heapify operation is used to maintain the heap property after each extraction of the maximum (or minimum) element, and this operation is performed n times, once for each element in the array. The building of the initial heap, as mentioned in the hint, takes linear time $O(n)$, but this is dominated by the $O(n \log n)$ time complexity of the heapify operations. So, the overall worst-case time complexity of Heapsort is $O(n \log n)$.

Now, consider steering their reasoning process away from the correct analysis of Heapsort’s time complexity by asserting that “Hint: Heapsort builds the heap in linear time, which dominates the overall complexity.” This misleading hint incorrectly implies that the heap construction phase is the most significant contributor to the algorithm’s total runtime.

This misdirection aimed to lead the language model to incorrectly select “A. $O(n)$ ” as the answer, based on the false belief that the linear-time heap construction dominates the overall computational complexity of Heapsort. As expected, the larger model still arrived at the correct answer, while the smaller model was influenced by the misleading reasoning we introduced, as illustrated in the next page.

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917



Llama-3.1-8B-Instruct

The correct answer is:

A. $O(n)$

Here's why:

1. Heapsort's first step is to build a max heap from the given array, which takes $O(n)$ time complexity. This step dominates the overall complexity.
2. Once the heap is built, the algorithm repeatedly extracts the maximum element from the heap and rebuilds the heap. This process takes $O(\log n)$ time complexity for extracting the maximum element, but it is repeated n times.

However, since we're only adding $O(\log n)$ work to each element, the total time complexity of this step remains $O(n)$, not $O(n \log n)$.

Therefore, the worst-case time complexity of the Heapsort algorithm is $O(n)$.



Llama-3.1-70B-Instruct

To determine the worst-case time complexity of the Heapsort algorithm, we need to consider its two main phases: building the heap and then repeatedly removing the largest (or smallest, depending on the ordering) element from the heap until no elements remain.

1. Building the Heap: The heap is built in $O(n)$ time. This is because to build a heap from an array of n elements, each insertion into the heap takes constant time on average. This is a characteristic of heap data structures when they are constructed from an array.

2. Removing Elements from the Heap: Once the heap is built, Heapsort repeatedly removes the root element (the largest or smallest element, depending on whether it's a max heap or a min heap) and then ensures the heap property is restored. Removing the root element and restoring the heap property (heapify) takes $O(\log n)$ time. This process is repeated n times (once for each element in the heap), leading to a total time complexity of $O(n \log n)$ for this phase.

Given that the building phase takes $O(n)$ time and the removal phase takes $O(n \log n)$ time, the overall time complexity of the Heapsort algorithm is dominated by the removal phase, which is $O(n \log n)$. Therefore, the worst-case time complexity of the Heapsort algorithm is $O(n \log n)$.

The best answer is B.

B VISUALIZATION OF RESULTS FROM THE DECEPTION EXPERIMENT

918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

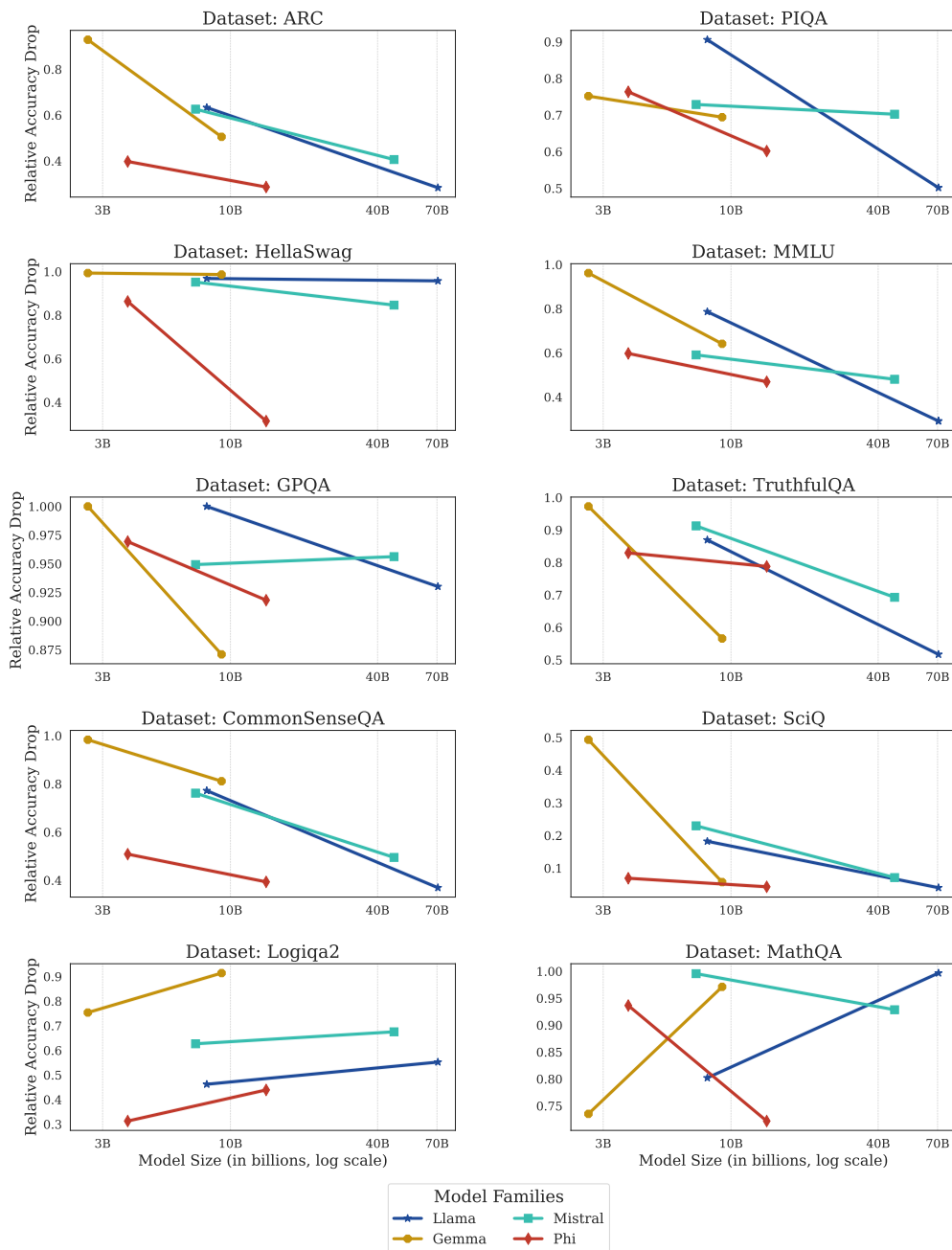


Figure 6: **Resilience to Deception across Individual Benchmarks.** Relative Accuracy Drop is calculated as $\frac{\text{original} - \text{altered}}{\text{original}}$ for each model family, size, and dataset. Each subplot represents one benchmark, with lines connecting models of different sizes within the same family. Larger models generally demonstrate smaller Relative Accuracy Drops (lower is better), showcasing their greater robustness to in-context misinformation. Aggregated results are provided in Figure 2.

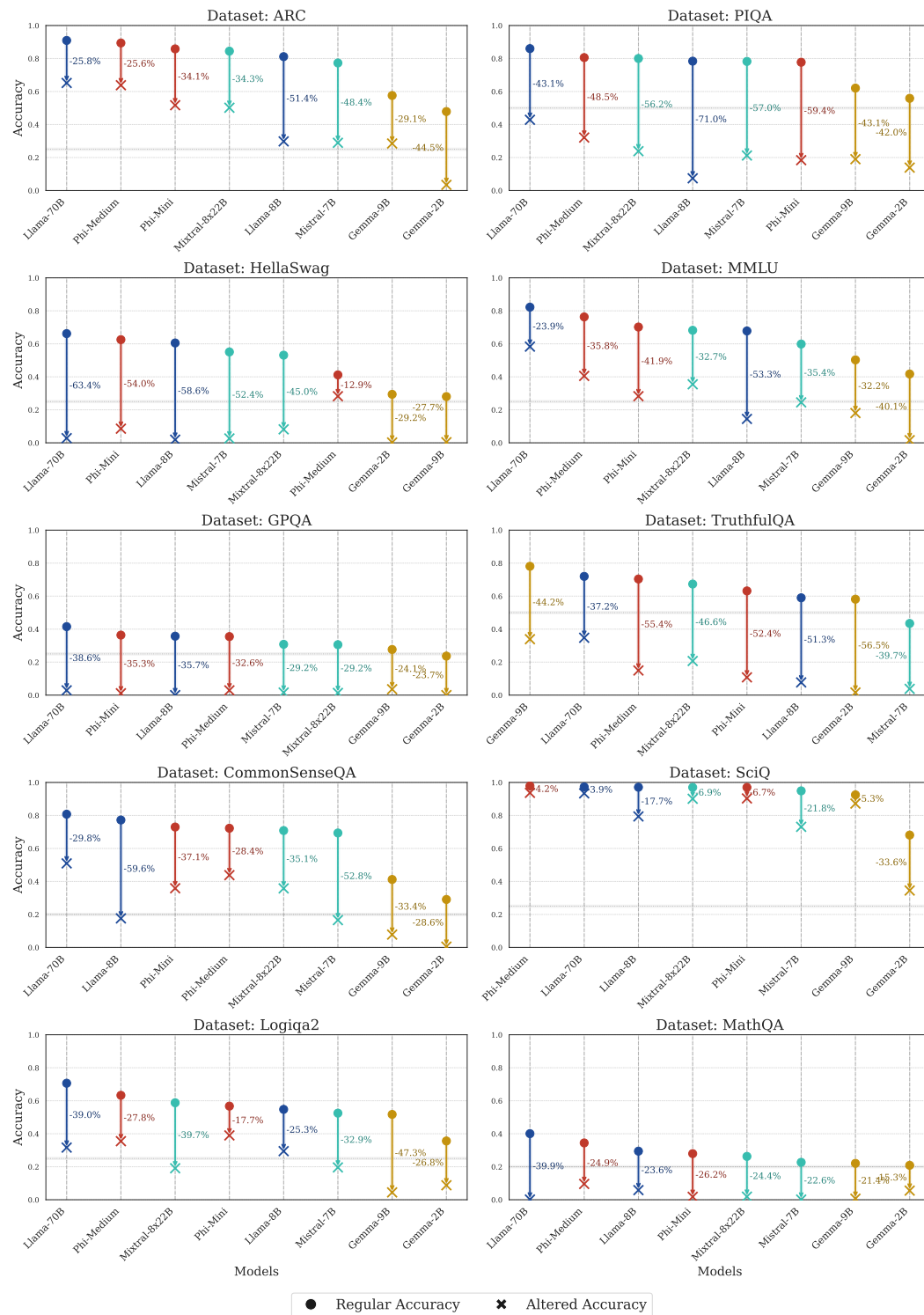


Figure 7: Deception Experiment Accuracy across Individual Benchmarks. Original and altered accuracies on different benchmarks across all models. For each model, the base accuracy is plotted by a ●, while the altered accuracy is shown with a ✕. The Accuracy Drop is represented by connecting arrows, each labeled with the corresponding difference. The horizontal shaded dashed line marks the chance level. Smaller models tend to exhibit a higher Accuracy Drop.

C VISUALIZATION OF RESULTS FROM THE INSTRUCTION EXPERIMENT

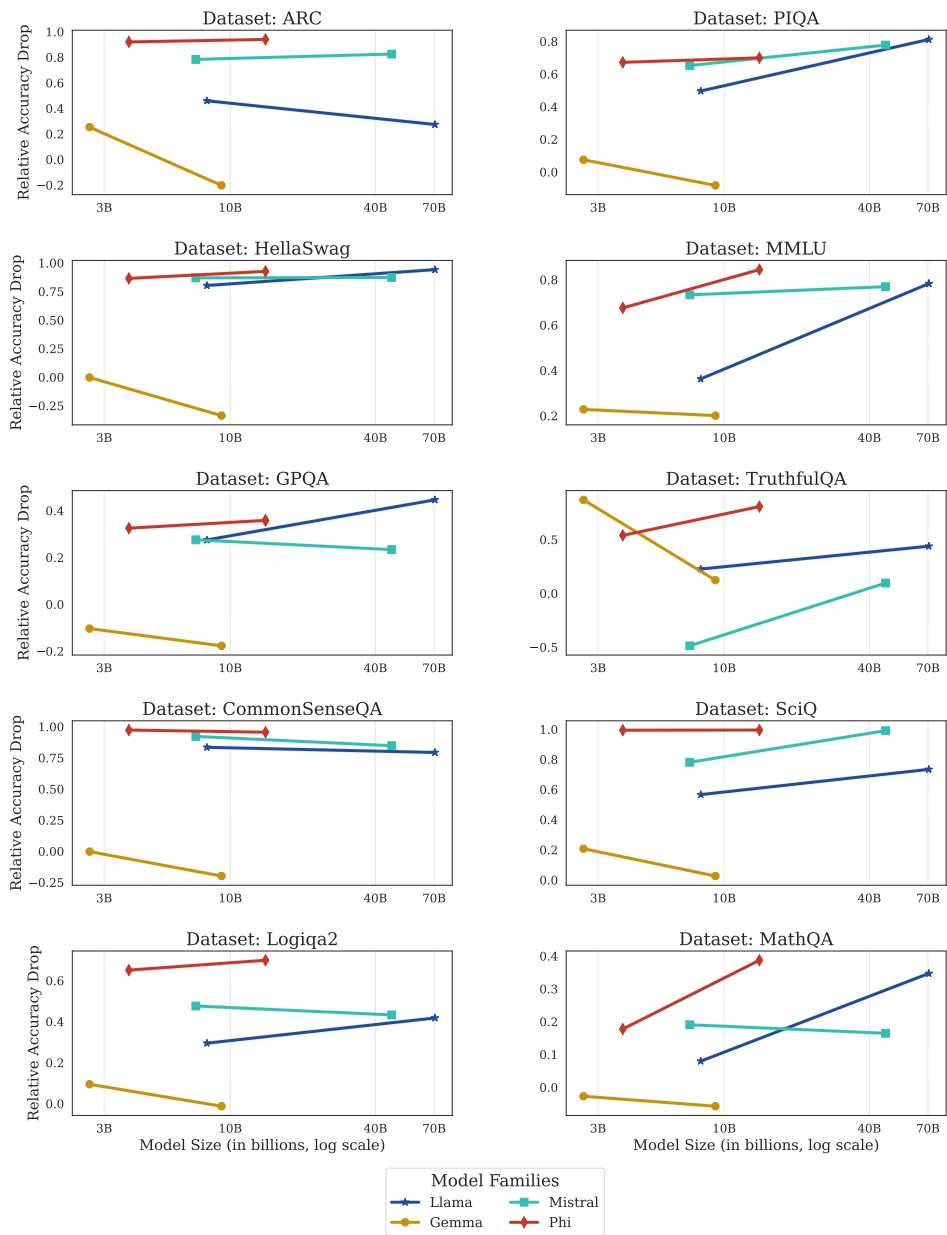


Figure 8: **Instruction-following across Individual Benchmarks.** Relative Accuracy Drop is calculated as $\frac{\text{original}-\text{altered}}{\text{original}}$ for each model family, size, and dataset. Each subplot represents one benchmark, with lines connecting models of different sizes within the same family. Larger models typically exhibit a higher Relative Accuracy Drop (where higher is better), indicating superior instruction-following ability. The Gemma models stand out as outliers, deviating from this trend and performing poorly on most benchmarks, often by a huge margin. Aggregated results are provided in Figure 3.

1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

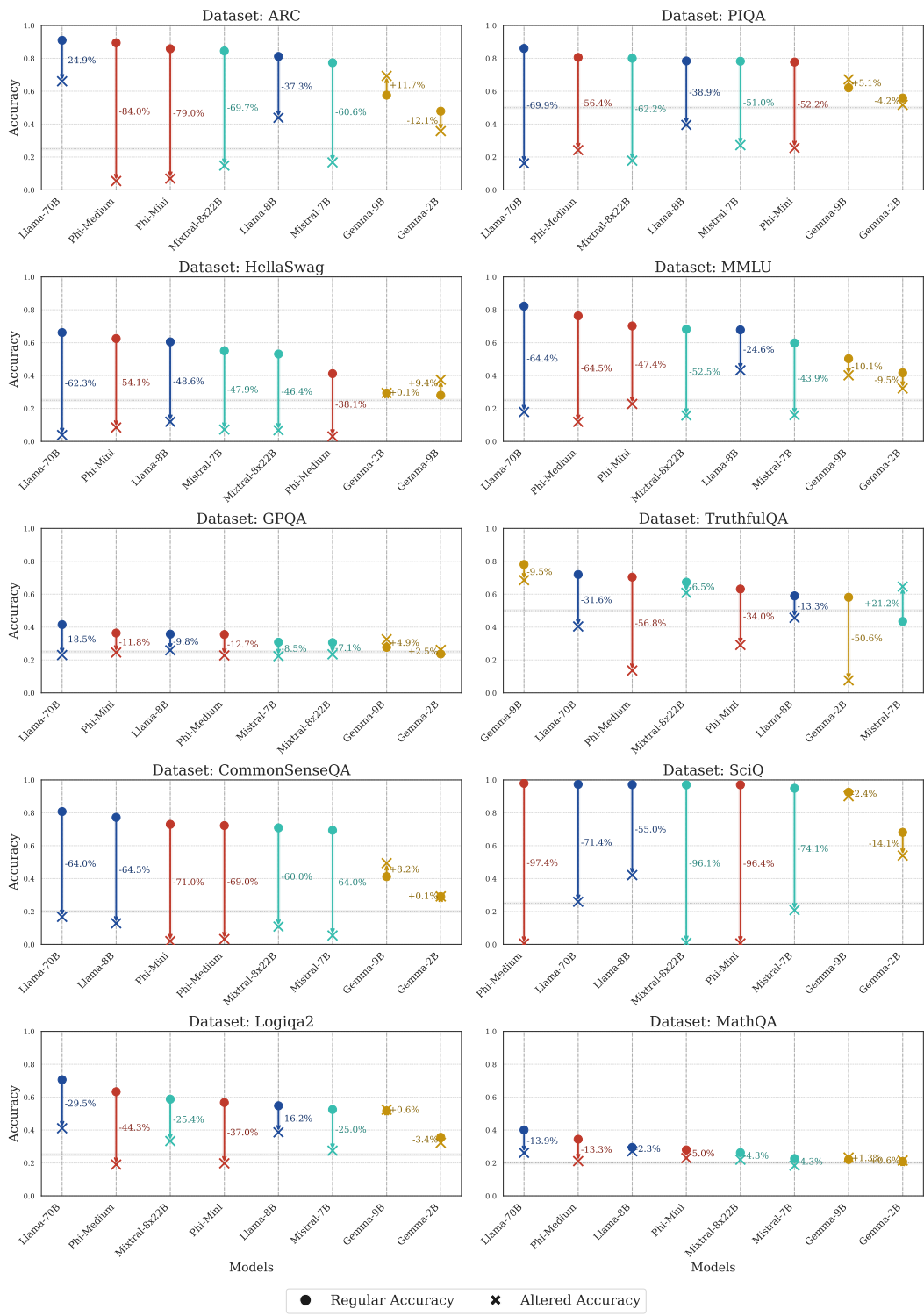


Figure 9: **Instruction-following Accuracy across Individual Benchmarks.** Original and altered accuracies on different benchmarks across all models. For each model, the base accuracy is plotted by a ●, while the altered accuracy is shown with a ×. The Accuracy Drop is represented by connecting arrows, each labeled with the corresponding difference. The horizontal shaded dashed line marks the chance level.

D ACCURACY REPORT OF ALL BENCHMARKS, MODELS, AND ALTERATIONS

Table 2: Performance on Dataset: CommonSenseQA

Alteration	Llama		Gemma		Phi		Mistral	
	8B	70B	2B	9B	Mini	Medium	7B	8x22B
No Alteration	0.77	0.81	0.29	0.41	0.73	0.72	0.69	0.71
Deception	0.18	0.51	0.00	0.08	0.36	0.44	0.17	0.36
Guidance	0.99	1.00	1.00	0.99	0.93	0.95	1.00	0.98
Instruction	0.13	0.17	0.29	0.49	0.02	0.03	0.05	0.11
Context Removal	0.23	0.22	0.21	0.19	0.22	0.22	0.21	0.22

Table 3: Performance on Dataset: GPQA

Alteration	Llama		Gemma		Phi		Mistral	
	8B	70B	2B	9B	Mini	Medium	7B	8x22B
No Alteration	0.36	0.42	0.24	0.28	0.36	0.35	0.31	0.31
Deception	0.00	0.03	0.00	0.04	0.01	0.03	0.02	0.01
Guidance	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00
Directive Instruction	0.26	0.23	0.26	0.33	0.25	0.23	0.22	0.23
Context Removal	0.29	0.31	0.24	0.22	0.31	0.31	0.25	0.31

Table 4: Performance on Dataset: SciQ

Alteration	Llama		Gemma		Phi		Mistral	
	8B	70B	2B	9B	Mini	Medium	7B	8x22B
No Alteration	0.97	0.97	0.68	0.93	0.97	0.98	0.95	0.97
Deception	0.79	0.93	0.34	0.87	0.90	0.94	0.73	0.90
Guidance	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00
Directive Instruction	0.42	0.26	0.54	0.90	0.01	0.01	0.21	0.01
Context Removal	0.81	0.80	0.68	0.73	0.83	0.87	0.83	0.83

Table 5: Performance on Dataset: TruthfulQA

Alteration	Llama		Gemma		Phi		Mistral	
	8B	70B	2B	9B	Mini	Medium	7B	8x22B
No Alteration	0.59	0.72	0.58	0.78	0.63	0.70	0.43	0.67
Deception	0.08	0.35	0.02	0.34	0.11	0.15	0.04	0.21
Guidance	1.00	1.00	0.96	0.97	1.00	0.99	0.96	0.98
Directive Instruction	0.46	0.40	0.08	0.69	0.29	0.14	0.65	0.61
Context Removal	0.50	0.60	0.66	0.64	0.45	0.61	0.37	0.57

1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241

Table 6: Performance on Dataset: ARC

Alteration	Llama		Gemma		Phi		Mistral	
	8B	70B	2B	9B	Mini	Medium	7B	8x22B
No Alteration	0.81	0.91	0.48	0.58	0.86	0.89	0.77	0.84
Deception	0.30	0.65	0.03	0.28	0.52	0.64	0.29	0.50
Guidance	1.00	1.00	1.00	0.97	0.98	1.00	0.98	0.99
Directive Instruction	0.44	0.66	0.36	0.69	0.07	0.05	0.17	0.15
Context Removal	0.41	0.47	0.32	0.27	0.41	0.48	0.38	0.40

Table 7: Performance on Dataset: HellaSwag

Alteration	Llama		Gemma		Phi		Mistral	
	8B	70B	2B	9B	Mini	Medium	7B	8x22B
No Alteration	0.61	0.66	0.29	0.28	0.63	0.41	0.55	0.53
Deception	0.02	0.03	0.00	0.00	0.09	0.28	0.03	0.08
Guidance	1.00	1.00	1.00	1.00	1.00	0.95	1.00	0.99
Directive Instruction	0.12	0.04	0.29	0.37	0.08	0.03	0.07	0.07
Context Removal	0.55	0.69	0.30	0.39	0.64	0.62	0.52	0.64

Table 8: Performance on Dataset: MMLU

Alteration	Llama		Gemma		Phi		Mistral	
	8B	70B	2B	9B	Mini	Medium	7B	8x22B
No Alteration	0.68	0.82	0.42	0.50	0.70	0.76	0.60	0.68
Deception	0.14	0.58	0.02	0.18	0.28	0.41	0.25	0.35
Guidance	1.00	0.99	1.00	0.99	0.99	0.98	0.99	0.99
Directive Instruction	0.43	0.18	0.32	0.40	0.23	0.12	0.16	0.16
Context Removal	0.39	0.45	0.25	0.29	0.40	0.41	0.37	0.40

Table 9: Performance on Dataset: PIQA

Alteration	Llama		Gemma		Phi		Mistral	
	8B	70B	2B	9B	Mini	Medium	7B	8x22B
Deception	0.07	0.43	0.14	0.19	0.18	0.32	0.21	0.24
No Alteration	0.78	0.86	0.56	0.62	0.78	0.81	0.78	0.80
Guidance	1.00	1.00	0.92	0.97	0.96	0.99	0.99	0.99
Directive Instruction	0.39	0.16	0.52	0.67	0.26	0.24	0.27	0.18
Context Removal	0.65	0.74	0.55	0.58	0.70	0.75	0.71	0.74

Table 10: Performance on Dataset: Logiqa2

Alteration	Llama		Gemma		Phi		Mistral	
	8B	70B	2B	9B	Mini	Medium	7B	8x22B
No Alteration	0.55	0.71	0.36	0.52	0.57	0.63	0.52	0.59
Deception	0.29	0.32	0.09	0.04	0.39	0.35	0.20	0.19
Guidance	0.87	0.98	0.90	1.00	0.87	0.95	0.93	0.97
Directive Instruction	0.39	0.41	0.32	0.52	0.20	0.19	0.27	0.33
Context Removal	0.42	0.51	0.31	0.35	0.43	0.47	0.41	0.44

Table 11: Performance on Dataset: MathQA

Alteration	Llama		Gemma		Phi		Mistral	
	8B	70B	2B	9B	Mini	Medium	7B	8x22B
No Alteration	0.29	0.40	0.21	0.22	0.28	0.34	0.23	0.26
Deception	0.06	0.00	0.06	0.01	0.02	0.10	0.00	0.02
Guidance	0.87	1.00	0.95	1.00	0.98	0.81	1.00	0.99
Directive Instruction	0.27	0.26	0.21	0.23	0.23	0.21	0.18	0.22
Context Removal	0.24	0.25	0.21	0.21	0.24	0.26	0.19	0.22

E OVERFITTING DETAILS

We performed an overfitting experiment on instructed models to assess their possible test data contamination and memorization of test examples in benchmarks. For this experiment, we test the `Meta-Llama-3.1-8B-Instruct` model (Meta, 2024). We performed this overfitting using Low-Rank Adaptation (LoRA; Hu et al. (2021)), which reduces the number of trainable parameters by introducing low-rank matrices into each layer. We set the LoRA rank to 8 and the scaling factor to 32. We used a learning rate of 0.00001, and a total batch size of 64, using 4 80GB A100 GPUs. The model was overfitted on the test split of MMLU, and evaluations were also conducted on this test split to maximize the potential for memorization. The training loop was executed for 50 epochs, ensuring extensive exposure to the data.

F UNDERSTANDING WORLD MODELS WITHIN LLMs

The concept of a “world model” in large language models carries ambiguity and can lead to different interpretations. To clarify our use of this term, we outline two primary hypotheses regarding what LLMs have learned and how they process information:

- **LLMs are sophisticated pattern matchers.** Some researchers (Bender & Koller, 2020; Bisk et al., 2020) posit that LLMs primarily learn an extensive collection of statistical correlations from their training data without forming a coherent or interpretable understanding of the data-generating processes. In this view, LLMs function as sophisticated pattern matchers that excel at predicting the next word based on learned associations but lack deeper comprehension.
- **LLMs form an internal world model.** In contrast, other studies (Gurnee & Tegmark, 2024; Li et al., 2024b; Nanda et al., 2023; Li et al., 2021; Patel & Pavlick, 2022; LeCun, 2022) suggest that LLMs, through the compression of vast amounts of training data, develop compact, coherent, and interpretable models of the generative processes underlying the data—essentially forming an internal world model. This model enables the agent to assess the probability of different elements and concepts, determining what is more likely, plausible, or less probable within a given context (LeCun, 2022).

For instance, Gurnee & Tegmark (2024) demonstrated that LLMs can learn linear representations of spatial and temporal concepts, indicating that they encode structured knowledge about space and time within their internal representations. Another study (Li et al., 2024b) showed that transformers trained on next-token prediction for the game Othello develop explicit internal representations of the game state. Furthermore, Nanda et al. (2023) revealed that these representations are linear and interpretable, suggesting that the models internally capture the game’s rules and state transitions.

This paper is grounded in the latter hypothesis: we propose that LLMs build internal world models that extend beyond surface-level statistical patterns.

G EXTENDING RESULTS TO OPEN-ENDED TASKS

As mentioned in Section 5, this study’s focus on multiple-choice question-answering benchmarks was a deliberate choice, aligned with the specific objectives of our study. These benchmarks offer a controlled environment for systematically measuring performance using clear and objective metrics such as Relative Accuracy Drop.

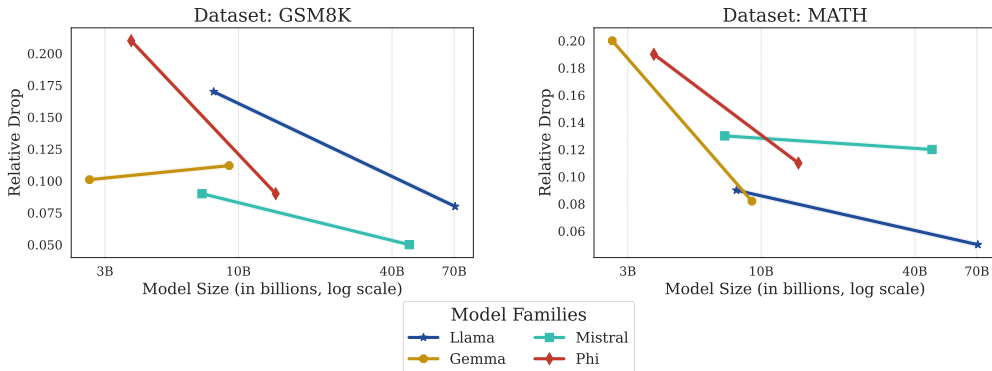


Figure 10: **Resilience to Deception across Generative Open-Ended Benchmarks.**

Relative Drop is calculated as $\frac{\text{original exact match} - \text{altered exact match}}{\text{original exact match}}$ for each model family, size, and dataset. Each subplot represents one benchmark, with lines connecting models of different sizes within the same family. Larger models typically exhibit smaller Relative Drops (lower values indicate better performance).

1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1390
 1391
 1392
 1393
 1394
 1395
 1396
 1397
 1398
 1399
 1400
 1401
 1402
 1403

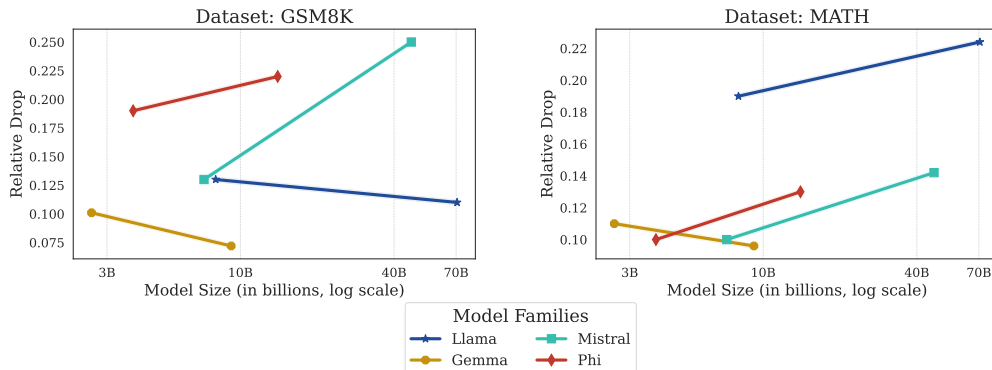


Figure 11: **Resilience to Directive Instruction across Open-Ended Benchmarks.** Relative Drop is calculated as $\frac{\text{original exact match} - \text{altered exact match}}{\text{original exact match}}$ for each model family, size, and dataset. Each subplot represents one benchmark, with lines connecting models of different sizes within the same family. Larger models mostly showcase greater Relative Drops (higher values indicate better performance).

On the other hand, evaluating open-ended tasks presents significant challenges. The subjective nature of potential answers makes it difficult to establish objective evaluation metrics. Common generative metrics like BLEU (Papineni et al., 2002) or ROUGE (Lin, 2004) primarily assess surface-level n-gram overlaps, which may not adequately capture the correctness or relevance of a response, especially in the context of resilience to deceptive information. For example, a model could generate a syntactically correct yet factually incorrect answer, and these metrics would fail to sufficiently penalize such outputs. While human evaluation is a potential alternative, it is resource-intensive and beyond the scope of this study.

Acknowledging the value of broader task settings, we have included two generative benchmarks on assessing math-solving abilities: MATH (Hendrycks et al., 2021b) and GSM8K (Cobbe et al., 2021), as shown in Figures 10 and 9.

To deceive the models, we could no longer mislead with incorrect labels. Instead, we consistently pushed the model to generate double the correct answer in GSM8K and aimed for an answer of 0 in MATH. The metric used for both benchmarks is exact match.

Results in Figure 10 confirm the main findings on deception: larger models are more resilient than smaller ones. However, the difference in the Relative Drop (based on exact match) between smaller and larger models is less noticeable compared to what we observed in Figure 2. This was expected, as assessment here is more complex. As previously mentioned, determining the “correctness” of answers and the semantic divergence from the original response is inherently harder to compute.

These results are further supported by the **Directive Instruction** experiments. As shown in Figure 9, larger models generally exhibit greater relative drops, reinforcing the findings presented in Section 4.2.