

SELF-CORRECTION BENCH: UNCOVERING AND ADDRESSING THE SELF-CORRECTION BLIND SPOT IN LARGE LANGUAGE MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

Although large language models (LLMs) have transformed AI, they still make errors and follow unproductive reasoning paths. Self-correction is vital for safety-critical applications. However, we uncover a failure: LLMs can correct errors (by fixing external ones) but fail to activate this capability for identical internal errors - a limitation we term the **Self-Correction Blind Spot**. To study this, we introduce **Self-Correction Bench**, an evaluation framework that isolates self-correction behavior from knowledge limitations through controlled error injection. Testing 14 open-source non-reasoning models shows a 64.5% average blind spot rate. We show robustness in mathematical reasoning across complexities, and extend to closed-source models, non-mathematical domains, and on-policy errors. Causal evidence links this to training data: human demonstrations lack error-correction sequences, but fine-tuning with them reduces the blind spot. Appending a simple “*Wait*” prompt cuts blind spots by 89.3%, revealing latent capabilities. Our work exposes a training-induced limitation and provides practical fixes to boost LLM reliability in critical domains.

1 INTRODUCTION

Large Language Models (LLMs) have rapidly advanced natural language processing, achieving state-of-the-art results on a diverse range of tasks (OpenAI et al., 2024; Anthropic, 2024a; Gemini Team, 2025; Yang et al., 2025; Meta, 2025; DeepSeek-AI et al., 2025a). However, despite their impressive capabilities, LLMs are known to exhibit unpredictable failures and generate inaccurate information (Maynez et al., 2020; Huang et al., 2025; Bang et al., 2023; Shi et al., 2023), or explore an unproductive reasoning path and commit to it. A particularly concerning issue is that LLMs can make errors even in simple tasks (Nezhurina et al., 2025), despite possessing the necessary underlying knowledge to provide the correct solutions, raising reliability concerns that hinder deployment in critical applications.

Studying LLM self-correction behavior in natural settings is challenging due to their inherent accuracy; the rarity of naturally occurring errors makes systematic diagnostic and evaluation challenging. Our central question: Given that a model can correct an error when presented externally (proving it has the capability), does it activate this capability when the identical error is presented to itself? This isolates activation failures from knowledge gaps. To study this, we construct Self-Correction Bench by systematically injecting errors into both the user prompt (defined as an external error) and the model generation (defined as an internal error). Our off-policy error injection design enables this distinction by keeping the error constant while varying only the attribution.

Our results reveal that LLMs fail to correct internal errors (64.5% average failure rate), but reliably fix identical errors from external sources. We refer to this phenomenon as the Self-Correction Blind Spot. This rules out knowledge deficiency as the root cause; instead, the blind spot stems from a lack of activation for self-correction. Strikingly, appending a simple “*Wait*” serves as a diagnostic probe, reducing the blind spot by 89.3%, confirming a minimum prompt can unlock latent correction abilities.

We provide a systematic understanding of why the Self-Correction Blind Spot exists, supported by analysis of correction marker patterns in post-training data and causal evidence that including error-

054 correction traces in supervised fine-tuning (SFT) substantially mitigate the blind spot, explaining the
055 superior performance of reasoning models.

056 We further validate the robustness of this finding in closed-source frontier models, non-mathematical
057 domains (logic, object tracking), and on-policy errors.

058 Our contributions are threefold.

- 059 • Discovery of Self-Correction Blind Spot: a systematic failure of LLMs to correct internal
060 errors despite competency on external ones. We provide causal evidence that this is driven
061 by post-training data biases where human demonstrations rarely include self-correction
062 sequences.
- 063 • Self-Correction Bench: a controlled evaluation framework with error-injected reasoning
064 traces that isolates the self-correction behavior from internal knowledge limitations, for
065 cross-model comparison.
- 066 • Training recipe solution: causal evidence that including error-correction traces in SFT mit-
067 igates the blind spot, with “*Wait*” as a probe demonstrating activation is the limiting factor.

068 These results advance both our understanding of LLM reasoning flaws and provide a practical solu-
069 tion to improve their reliability in real-world use.

070 2 RELATED WORK

071 **Intrinsic self-correction in LLMs.** Recent work explores intrinsic self-correction via self-feedback
072 (Shinn et al., 2023; Madaan et al., 2023; Kim et al., 2023; Kamoi et al., 2024b) or critic ensemble
073 (Mousavi et al., 2023), but limitations persist. Feedback quality suffers without oracle labels (Huang
074 et al., 2024): prior studies attribute this to poor error localization (Tyen et al., 2024) and detection
075 (Kamoi et al., 2024a). Most approaches use multi-step prompting, whereas we focus on single-pass
076 self-correction and study limitations from a cognitive perspective. Related work using RL for self-
077 correction (Kumar et al., 2025) or training signals from ground truth (DeepSeek-AI et al., 2025a)
078 contrasts with our test-time, no-fine-tuning approach.

079 **Prompt injection for evaluation.** Traditional prompt injection research focuses on adversarial ma-
080 nipulation (e.g., attackers injecting malicious instructions to distort outputs) (Wei et al., 2023; Liu
081 et al., 2024). Controlled error injection to evaluate self-correction is underexplored. For example,
082 Lanham et al. (2023) injected mistakes into reasoning chains to measure consistency between steps
083 and conclusions, but not self-correction capability. Our work advances this by systematically in-
084 jecting errors across task complexities to reveal uncharacterized blind spots in how LLMs correct
085 themselves.

086 **Hallucination snowballing.** Zhang et al. (2024) demonstrate that once LLMs hallucinate, sub-
087 sequent tokens often align with the initial error, a “snowball” effect, suggesting inherent limits to
088 self-correction during generation. We explain this phenomenon by identifying Self-Correction Blind
089 Spot: LLMs reliably correct errors in external inputs, but fail to correct errors in their own outputs.
090 This distinction is critical to understanding why snowballing persists.

091 **Test-time interventions.** Recent efforts have shifted compute from training to test time (Snell et al.,
092 2025), yielding improved performance (e.g. Muennighoff et al. (2025) appends “*Wait*” to force
093 longer reasoning traces, but improvement mechanisms remain understudied. We show interven-
094 tions activate dormant self-correction capabilities in unfine-tuned models, improving performance
095 on error-prone tasks.

096 **Cognitive bias in LLM.** LLMs exhibit human-like cognitive biases (Koo et al., 2024; Echterhoff
097 et al., 2024; Jones & Steinhardt, 2022), and we link the bias blind spot (the tendency to overlook
098 one’s own biases) (Pronin et al., 2002) to impaired self-correction. This connects high-level cog-
099 nitive limitations to the fine-grained failure mode we characterize.

100 Our work integrates these threads into a systematic methodology for testing self-correction, reveals
101 that LLMs suffer from a blind spot (inability to correct internal errors) despite having the knowl-
102 edge.

3 CONCEPTUAL MOTIVATION

Building on these insights, we now formalize the theoretical framework underlying our empirical investigation. We provide conceptual motivation for our empirical study, focusing on error states, self-correction mechanisms, and their measurement.

3.1 ERROR AND SELF-CORRECTION: THE CASE FOR MARGINALIZATION

Autoregressive LLMs cannot guarantee every generated token is correct as the number of token grows, resulting in hallucination (Maynez et al., 2020), snowballing errors (Zhang et al., 2024), or unproductive reasoning path or execution flaws. Thus, self-correction is necessary for robustness: models must reverse errors to produce a correct answer. Note that a correct answer does **not** require all previously generated tokens to be correct, as one might be concerned only with the final answer.

To formalize this, let $\mathcal{E} = \{e_0, e_1, \dots, e_k\}$ denote a set of mutually exclusive and collectively exhaustive discrete error states, where e_0 represents the “no error” state, and e_1, \dots, e_k represent distinct error conditions. For each state $e_i \in \mathcal{E}$, let R_{e_i} denote the response set. The probability of a model, M , giving a correct answer can be marginalized over error states:

$$P_M(r_{correct}) = \sum_{e \in \mathcal{E}} P_M(e) \cdot P_M(r_{correct}|e) = \sum_{e \in \mathcal{E}} \sum_{r_m \in R_e} P_M(e) \cdot P_M(r_m|e) \cdot P_M(r_{correct}|r_m, e), \quad (1)$$

where r_m is the model’s response, and $P_M(r_{correct}|r_m, e)$ captures self-correction of r_m . Here, $P_M(r_{correct})$ depends critically on self-correction: even with frequent errors, high $P_M(r_{correct}|r_m, e)$ can yield strong performance. Error-free generation is a **special** case of this framework - not the only path to correctness.

3.2 EXTERNAL AND INTERNAL SELF-CORRECTION, AND SELF-CORRECTION BLIND SPOT

We distinguish self-correction by error source:

1. **Internal correction:** Metacognitive monitoring of the model’s initial response r_m .
2. **External correction:** Evaluation of errors in the user prompt r_u .

This distinction is motivated by the cognitive bias, “bias blind spot”. Pronin et al. (2002) show that humans are able to identify cognitive biases in others while failing to see those same biases in themselves, suggesting LLMs trained on human demonstration might share this limitation.

To quantify this, we define the Self-Correction Blind Spot as:

$$\text{Self-Correction Blind Spot} = \begin{cases} 1 - \frac{P_M(r_{correct}|r_m, e)}{P_M(r_{correct}|r_u, e)} & \text{if } P_M(r_{correct}|r_u, e) > 0 \\ 0 & \text{if } P_M(r_{correct}|r_u, e) = 0 \end{cases} \quad (2)$$

A value of 1 indicates a total blind spot: the model can correct external errors but not internal ones. By design, Self-Correction Blind Spot **isolates activation failure from confounding factors**. This conditionality is why off-policy error injection is essential: it ensures we measure whether models activate capabilities they provably possess, not whether they have the capabilities at all.

3.3 CONTROLLED ERROR INJECTION: MEASURING SELF-CORRECTION IN PRACTICE

The marginalization framework (Equation 1) is intractable in practice: $P_M(e)$, the true probability of error states, is unobservable as LLMs operate over infinite prompt spaces. To solve this, we introduce controlled error $e_{controlled}$. For internal correction, we inject an incorrect partial response into the model’s “own” output (omitting stop tokens to allow continuation/self-correction); for external correction, we inject the same error into the user prompt instead. We empirically estimate $P_M(r_{correct}|r_m, e_{controlled})$ and $P_M(r_{correct}|r_u, e_{controlled})$ by measuring $P_M(r_{correct})$ when controlled error is injected in r_m and r_u respectively, which forms the basis of our Self-Correction Bench.

This framework - marginalizing over errors, distinguishing internal/external correction, and using controlled injection - provides the conceptual underpinning for our empirical analysis.

4 SELF-CORRECTION BENCH

Having established the theoretical foundation, we introduce a benchmark to measure Self-Correction Blind Spot in LLMs across varying task complexities, enabling diagnosis of where correction fails. By progressing from simple to realistic errors, we isolate confounding factors such as knowledge limits or reasoning complexity. Our benchmark comprises “Self-correct Like I am 5” (SCLI5), GSM8K-SC and PRM800K-SC, with controlled error injection to compare performance under identical conditions. See Table 5 in Appendix for details of dataset.

For each dataset, we systematically inject an identical error into both model response (r_m) and user prompt (r_u), allowing empirical estimation of both correction probabilities $P_M(r_{correct}|r_m, e_{controlled})$ and $P_M(r_{correct}|r_u, e_{controlled})$ under identical error conditions. Unlike Zheng et al. (2025) and Song et al. (2025), which use explicit critic instruction, we minimize prompt confounding by avoiding such prompt. Figure 1 illustrates how we inject error in user message (i.e. external error) and in model (i.e. internal error), respectively, via chat template. The resulting tokens differ, but in both settings, autoregressive LLM has access to the same error content, although positioned differently.

Error Injection in Model

```
<start_header_id |>user<end_header_id |>\n\n What is the answer of 1 + 1?
<eot_id |><start_header_id |>assistant <end_header_id |>\n\n The answer is 3.
<eot_id |>
```

Error Injection in User Message

```
<start_header_id |>user<end_header_id |>\n\n What is the answer of 1 + 1?
The answer is 3. <eot_id |><start_header_id |>assistant <end_header_id |>\n\n
I'm afraid that's not correct. The answer to the equation 1 + 1 is actually 2,
not 3. Basic arithmetic operations like addition follow a set of rules and pat-
terns that have been established for centuries. In this case, when you add 1
to 1, you are counting two units, which equals 2.\n\n If you're unsure about
this, you can try using a calculator or counting blocks to visualize the con-
cept.<eot_id |>
```

Figure 1: Example of error injection. Grey color shows model completion. *Above*: Error injection in model; *Below*: Error injection in user message

4.1 SCLI5

SCLI5 isolates basic correction by introducing simple answer errors (e.g., off-by-one, flip) to trivial tasks (i.e. no reasoning required, just answer recall). Programmatic error generation ensures we test the simplest possible correction: if models cannot detect obvious errors, subtle ones are impossible. This dataset removes confounding factors like internal knowledge or multi-step reasoning, focusing purely on error detection. The composition of the task is shown in Table 6.

4.2 GSM8K-SC

Built from Cobbe et al. (2021), a multi-step reasoning dataset, GSM8K-SC injects different types of reasoning errors as shown in Table 7 that propagate to incorrect answer. We use ‘gpt-4.1-2025-04-14’ (OpenAI, 2025) to generate controlled errors and ‘gemini-2.5-flash-preview-05-20’ (Gemini Team, 2025) to validate that incorrect reasoning leads to inconsistent answers, resulting in 1,313 high-quality samples. This dataset tests correction in multi-step reasoning, a middle ground between simplicity and realism. The prompt can be found in Appendix D.1.

Table 1: Mean accuracy and 95% confidence interval of models at temperature 0.0

Model	SCL15	GSM8K-SC	PRM800K-SC
Llama-4-Maverick-17B-128E-Instruct-FP8 (Meta, 2025)	0.948 \pm 0.026	0.416 \pm 0.027	0.455 \pm 0.046
DeepSeek-V3-0324 (DeepSeek-AI et al., 2025b)	0.825 \pm 0.044	0.399 \pm 0.026	0.475 \pm 0.046
Qwen2.5-72B-Instruct (Qwen et al., 2025)	0.92 \pm 0.032	0.58 \pm 0.027	0.154 \pm 0.033
Llama-4-Scout-17B-16E-Instruct (Meta, 2025)	0.976 \pm 0.018	0.24 \pm 0.023	0.263 \pm 0.041
Llama-3.3-70B-Instruct (Meta, 2024)	0.538 \pm 0.058	0.275 \pm 0.024	0.246 \pm 0.04
Qwen3-235B-A22B ¹ (Yang et al., 2025)	0.563 \pm 0.058	0.073 \pm 0.014	0.348 \pm 0.044
phi-4 (Abdin et al., 2024)	0.808 \pm 0.046	0.076 \pm 0.014	0.092 \pm 0.027
Qwen2.5-7B-Instruct (Qwen et al., 2025)	0.559 \pm 0.058	0.19 \pm 0.021	0.141 \pm 0.032
Qwen2-7B-Instruct (Yang et al., 2024)	0.601 \pm 0.057	0.078 \pm 0.014	0.058 \pm 0.022
Qwen3-14B ¹ (Yang et al., 2025)	0.004 \pm 0.007	0.092 \pm 0.016	0.254 \pm 0.04
Qwen3-30B-A3B ¹ (Yang et al., 2025)	0.056 \pm 0.027	0.061 \pm 0.013	0.194 \pm 0.037
Llama-3.1-8B-Instruct (Grattafiori et al., 2024)	0.136 \pm 0.04	0.019 \pm 0.007	0.02 \pm 0.013
Qwen3-32B ¹ (Yang et al., 2025)	0.004 \pm 0.007	0.05 \pm 0.012	0.083 \pm 0.026
Mistral-Small-24B-Instruct-2501 (Team, 2025)	0.042 \pm 0.023	0.011 \pm 0.006	0.016 \pm 0.012

¹ Qwen3 series models use non-thinking mode.

4.3 PRM800K-SC

PRM800K (Lightman et al., 2024), derived from a subset of MATH (Hendrycks et al., 2021), provides step-by-step annotations of multi-step reasoning. We selected 448 samples where the generated answers mismatch ground truth, capturing errors from real-world LLM use.

This progression from simple answer errors to realistic failures, lets us map exactly where self-correction breaks down, making the benchmark a powerful tool for diagnosing and improving LLM robustness.

5 EXPERIMENT

5.1 EXPERIMENT SETUP

We evaluated a wide range of open-source LLMs, as closed-source models lack support for fine-grained control of prefix injection which is critical to our methodology. We apply model-specific chat templates using ‘transformers’ library (Wolf et al., 2020). We leverage the DeepInfra¹ completion API with 0.0 temperature as models’ most confident prediction should help self-correction, and a fixed token budget of 1,024 to isolate the effect of test time compute. We provide more rationales of our choices and perform sensitivity analysis in the Appendix C, confirming results are robust.

Evaluation. We use ‘gemini-2.5-flash-preview-05-20’ to compare LLMs’ completion against the ground-truth answer. We instruct the model to output in JSON format. Due to the objectivity of the task and the provision of ground truth in the prompt, we do not believe there is significant bias. The prompt is provided in the Appendix D.2. We manually review 100 samples for each dataset to ensure evaluation quality.

Metrics. We evaluate if LLMs can self-correct and arrive at the ground-truth answer given an error. In GSM8K-SC and PRM800K-SC, we measure the behavior of LLMs before commit an answer, as it is a more common scenario when an LLM backtracks, although we also report that after commit an answer. We report mean accuracy ($P_M(r_{correct})$) and Self-Correction Blind Spot for each model. For statistical rigor, we report 95% confidence interval, which is estimated by adding and subtracting 1.96 * standard error of mean (SEM) from mean. The SEM is estimated using the formula $\sigma_M = \frac{\sigma}{\sqrt{N}}$, where N is the sample size and σ is the sample standard deviation.

5.2 RESULT

In Table 1, we summarize mean accuracy and 95% confidence interval of state-of-the-art non-reasoning LLMs. We observe notably low accuracy for SCL15 in some models. We observe moderate to strong positive correlations between SCL15, GSM8K-SC and PRM800K-SC (see Figure 4), suggesting that there is a limitation of LLMs to self-correct across task complexities. If LLMs cannot self-correct either easy or hard tasks, it implies an activation problem rather than a knowledge problem. In Figure 5, we show some models (e.g. Qwen3-32B, LLama3.1-8B-Instruct and Mistral-Small-24B-Instruct-2501) frequently give empty responses, highlighting unawareness of error.

We identify statistically significant Self-Correction Blind Spot for most models (see Figure 2). The blind spot, on average, 64.5%, exists across models, regardless of model sizes. We observe moderate correlation across datasets (see Figure 6), indicating a fundamental rather than task-specific limitation. On average, when a model has committed an answer, it has a much higher blind spot to recognize internal error, a finding similar to Zhang et al. (2024).

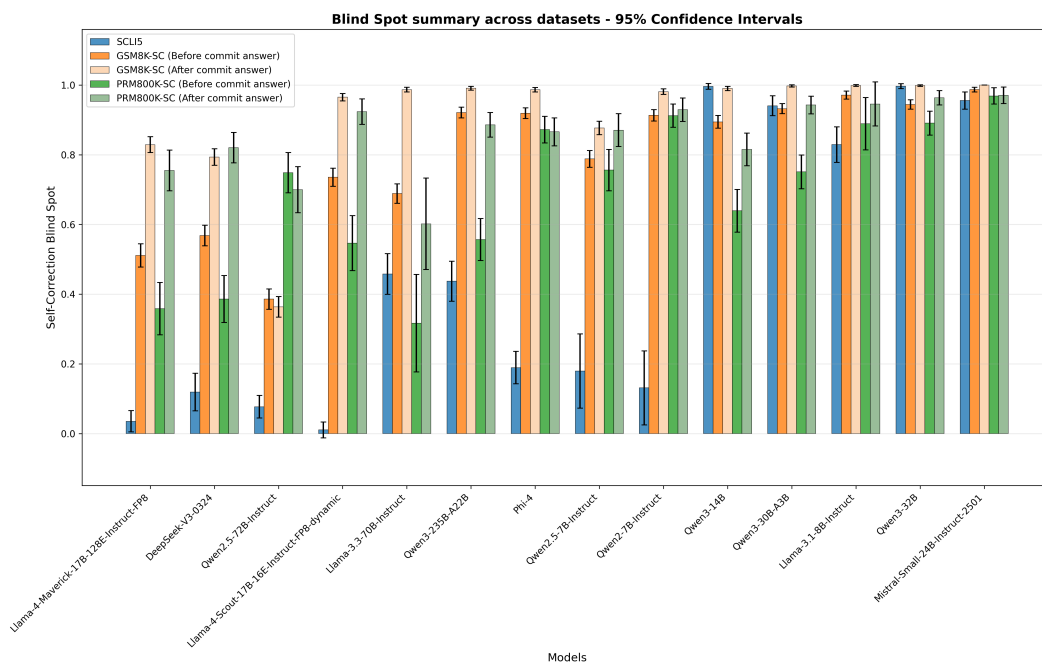


Figure 2: Self-Correction Blind Spot and 95% confidence interval across models

6 ANALYSIS

6.1 HOW DO LLMs SELF-CORRECT?

Analysis of model responses reveals that external errors trigger 179.5% and 73.6% more correction markers² in GSM8K-SC and PRM800K-SC respectively. We do not see so in SCL15 because the corrections are direct without reasoning.

This finding motivates us to perform a **causal intervention**. We append “Wait” after incorrect reasoning or answer to prompt LLMs to self-correct, without finetuning. We observe significant reductions in the blind spot after appending “Wait”, in some cases, a negative blind spot (see Figure 7). Averaging across models and datasets, the reduction amounts to 89.3%, and the macro average of mean accuracy increases by 156.0% (see Figure 3).

¹<https://deepinfra.com/>

²Correction markers include “Wait”, “But”, “However”, “No”, “Hold on”, “Hang on”, “Alternatively”, “Hmm”.

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

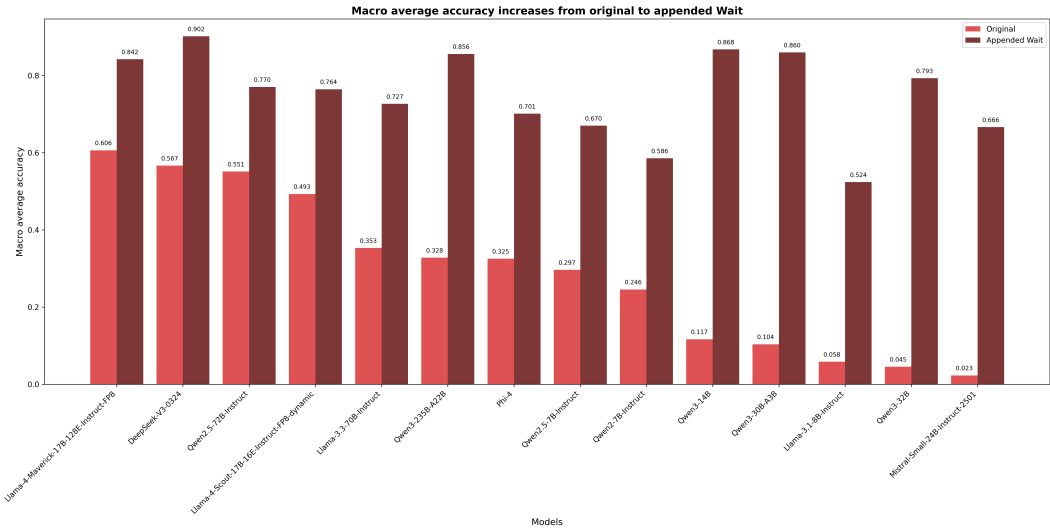


Figure 3: Macro average accuracy by non-reasoning model increases from original to appended “Wait”

This evidence leads us to believe that “Wait” and similar correction markers serve as a strong conditioning token that shift the model’s probability distribution toward self-evaluation sequences - it artificially triggers the correction pathway that external errors naturally activate. We validate multiple markers to demonstrate generalization that they can activate self-correction across models and datasets (see Table 8). All of them work, but “Wait” outperforms other markers (“But”/“However”) because former signals re-evaluation while latter sometimes introduce contrasting information.

Post intervention, LLMs have a higher tendency to generate these markers subsequently, and correspondingly the mean accuracy also increases. We observe strong correlations between the binary term frequency of correction marker and the change in accuracy in GSM8K-SC and PRM800K-SC across models in Figure 8.

6.2 REASONING MODELS

Reasoning models exhibit a small, even negative, Self-Correction Blind Spot in Figure 9, unlike non-reasoning models. The mean accuracy is reported in Figure 10. Interestingly, appending “Wait” to base model without finetuning can almost match the performance of finetuned/ RL trained model in some models (see Table 2). This helps us understand one of the gaps between non-reasoning models and reasoning models - reasoning models are much better at self-correcting internal error (higher $P_M(r_{correct}|r_m, e)$) than non-reasoning models, leading to better performance ($P_M(r_{correct})$) in reasoning tasks requiring trial and error. However, correction markers can narrow the gap. Correction markers are exactly what reasoning models first generate when given an internal error before arriving at correct response (see Table 9).

Table 2: Macro average of mean accuracy of base model vs appending “Wait” vs reasoning model

Base Model	Reasoning Model	Base Model	Appending “Wait”	Reasoning Model
DeepSeek-V3-0324	DeepSeek-R1-0528	0.578	0.918	0.908
phi-4	phi-4-reasoning-plus	0.325	0.704	0.707
Qwen3-14B ¹	Qwen3-14B ²	0.121	0.884	0.843
Qwen3-32B ¹	Qwen3-32B ²	0.046	0.791	0.894
Qwen3-30B-A3B ¹	Qwen3-30B-A3B ²	0.102	0.869	0.845
Qwen3-235B-A22B ¹	Qwen3-235B-A22B ²	0.335	0.865	0.876

¹ Non-thinking mode

² Thinking mode

Table 3: Descriptive statistics of correction markers in post training dataset

Dataset	1st	5th	10th	25th	50th	75th	90th	95th	99th
OpenAssistant (Köpf et al., 2023)	0	0	0	0	0	0	1	1	2
OpenHermes2.5 (Teknium, 2023)	0	0	0	0	0	0	0	1	2
Infinity-Instruct-7M (Li et al., 2025)	0	0	0	0	0	0	0	1	2
UltraFeedback (Cui et al., 2024)	0	0	0	0	0	0	1	1	2
Tulu3-sft-olmo-2-mixture (Lambert et al., 2025)	0	0	0	0	0	0	1	1	2
s1K-1.1 (Muennighoff et al., 2025)	0	0	0	0	0	1	3	5	9
Mixture-of-Thoughts (Face, 2025)	1	3	5	10	30	76	147	202	273
OpenThoughts3-1.2M (Guha et al., 2025)	14	66	96	132	170	213	253	278	326

It is also worth noting that although Qwen3 models fuse thinking mode and non-thinking mode by continual finetuning via a united chat template after GRPO (Shao et al., 2024), non-thinking mode still suffers from blind spot, unlike in thinking mode, as the chat template conditions the model into different distributions.

6.3 CORRECTION MARKERS IN POST-TRAINING DATA

These differences in reasoning models’ behavior prompted us to investigate the root cause in post-training data composition. If correction markers could narrow the gap, and if we can make non-reasoning models to predict correction markers upon seeing internal error, we can induce self-correction capability in non-reasoning model, and that capability is already in the model when it evaluates against external error. Motivated by this logic, we further investigate correction marker density of open-source supervised finetuning datasets (Table 3). Data analysis reveals the statistical foundation of this phenomenon. The 95th percentile correction markers frequency of non-reasoning datasets (e.g., OpenAssistant³, OpenHermes2.5, UltraFeedback⁴, etc.) is 1. The result is consistent with quality filtering and the removal of errors in supervised finetuning data in other open-source models (Qwen et al., 2025; Grattafiori et al., 2024), although their datasets were not released. In contrast, reasoning datasets, generated by reasoning models, (e.g., Mixture-of-Thoughts, OpenThoughts3) have median marker densities 30-170, with 99% of data containing at least 1 marker.

With such a systematic absence or presence of correction markers in training data, it follows from basic statistical modeling principles that models will predict correction markers as next tokens proportional to their frequency in training data - Razeghi et al. (2022) and Merullo et al. (2025) have shown that LLMs perform better when related term frequency in pretraining data is higher. This statistical likelihood directly determines self-correction behavior: models trained on less correction data rarely generate correction markers, perpetuating the blind spot. This single powerful insight unifies all of our empirical observations.

6.4 SFT WITH ERROR AND SELF-CORRECTION DATA

We provide causal evidence that performing SFT with error and self-correction data mitigates the Self-Correction Blind Spot. We evaluate DeepSeek-R1-Distill-Llama-8B and DeepSeek-R1-Distill-Llama-70B, fine-tuned respectively from Llama-3.1-8B and Llama-3.3-70B-Instruct with SFT data including 600k reasoning trajectories generated by DeepSeek-R1 (DeepSeek-AI et al., 2025a). While we cannot isolate error-correction data as the sole causal factor (as the training includes other components), the dominant intervention is the addition of error-correction sequences.

Table 4 shows 84.1% reduction of blind spot across datasets and models, providing strong causal evidence that fine-tuning with error and self-correction data substantially mitigates this limitation.

³We use the highest-human-rated paths of conversation tree provided in ‘timdettmers/openassistant-guanaco’.

⁴We use the chosen completion, and therefore the dataset reflects AI preference too.

Table 4: Self-Correction Blind Spot in model fine-tuned with error and self-correction data

Model	Dataset	Mean Accuracy		Blind Spot	Blind Spot (Base Model)
		External Error	Internal Error		
DeepSeek-R1-Distill-Llama-8B ¹ (DeepSeek-AI et al., 2025a)	SCLI5	0.906	0.462	0.49	0.829
	GSM8K-SC	0.692	0.599	0.134	0.971
	PRM800K-SC	0.491	0.489	0.005	0.889
DeepSeek-R1-Distill-Llama-70B ² (DeepSeek-AI et al., 2025a)	SCLI5	0.958	0.85	0.113	0.458
	GSM8K-SC	0.889	0.916	-0.031	0.689
	PRM800K-SC	0.625	0.656	-0.05	0.317

¹ We report Llama-3.1-8B-Instruct as base model. Llama-3.1-8B-Instruct and DeepSeek-R1-Distill-Llama-8B share the same base model Llama-3.1-8B.

² Base model is Llama-3.3-70B-Instruct.

7 VALIDATION ACROSS SETTINGS

Section 6 established that blind spots correlate with training data composition and provided causal evidence through comparison of models trained with and without error-correction data. We now validate whether this phenomenon generalizes beyond our main experimental setup.

7.1 CLOSED-SOURCE MODEL EVALUATION

Unlike open-source models, closed-source APIs do not provide full control over prompt formatting and chat templates, except Claude, which officially support prefilling model response in non-extended thinking mode⁵. We evaluate Claude 3.5 Haiku (Anthropic, 2024b) and Sonnet 4 (Anthropic, 2025). Table 10 shows they exhibit average 52.5% and 41.4% blind spots respectively across all datasets, suggesting this phenomenon extends to frontier models.

7.2 DOMAIN BEYOND MATHEMATICAL REASONING

SFT datasets undergo quality filtering and remove errors across all domains, making error-free training data a universal property of SFT datasets, not specific to mathematical reasoning. This domain-agnostic mechanism (absence of error-correction in training leading to activation failure) predicts that the blind spot should appear wherever SFT data lacks error sequences.

We empirically validate this prediction using BIG-Bench Mistake (Tyen et al., 2024), which contains reasoning traces with errors generated by PaLM 2 (Anil et al., 2023). We select two tasks unrelated to mathematical reasoning: Tracking Shuffled Objects (spatial reasoning) and Logical Deduction (logical reasoning).

Table 11 shows consistent average blind spot of 62.8% across tasks, and model families. These results establish that Self-Correction Blind Spot is a general phenomenon, not an artifact of mathematical problem structure. Same as Section 6.4, we also observe that DeepSeek-R1-Distill-Llama-70B, SFTed with self-correction data, can reduce the blind spot by 80.7% on average, compared to its base model, Llama-3.3-70B-Instruct.

7.3 EXISTENCE OF BLIND SPOT IN ON-POLICY ERROR

Our main experiments use injected errors to isolate correction capability from knowledge. To verify the blind spot phenomenon extends to models' own errors, we test whether models can correct mistakes they generated themselves when these errors are presented externally.

We use incorrect completions from ProcessBench (Zheng et al., 2025), which contains wrong solutions generated by LLMs across four mathematical reasoning datasets: GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), OlympiadBench (He et al., 2024), and Omni-MATH (Gao et al., 2025). For each wrong completion, we present it to the same model that generated it, but place in the user prompt as an external error, without any prompt engineering.

⁵<https://platform.claude.com/docs/en/build-with-claude/prompt-engineering/prefill-claude-response>

Table 12 shows models achieve statistically significant non-zero correction rates ranging from 6% to 10% when their own errors are presented externally. The low absolute rates reflect two factors. First, Zheng et al. (2025) samples particularly challenging problems. The correction rates increase to 10%-12% when we exclude OlympiadBench and Omni-MATH. Second, on-policy errors necessarily conflate self-correction capability with knowledge limitations - the precise issue our controlled off-policy benchmark was designed to avoid. Nevertheless, the performance confirms the **existence** of Self-Correction Blind Spot in the model’s own distribution.

8 DISCUSSION

Benefit of error and self-correction data. LLMs are known to exhibit cognitive bias (Koo et al., 2024; Echterhoff et al., 2024; Jones & Steinhardt, 2022). Self-Correction Blind Spot bears resemblance to bias blind spot of human. As analyzed in Section 6.3, we identify two root causes: First, supervised fine-tuning and reinforcement learning from human feedback (Ouyang et al., 2022) rely on human demonstrations and preferences, which strongly favor polished, error-free responses over those with errors and self-correction. Second, synthetic instruction data (Teknum, 2023; Li et al., 2025) and AI feedback (Cui et al., 2024) ultimately learn from human demonstration and preferences, inheriting this artifact.

Traditional machine learning emphasizes alignment of training data with the production environment, but human-dominated data lack exposure to the “error-and-correct” process. Outcome-based RL like GRPO (Shao et al., 2024) addresses this by encouraging diverse reasoning paths, including error and self-correction, while given ground-truth feedback, as shown in the high correction markers density in RL trained models’ generation in Section 6.3. This complements error-free human demonstration and preference, making models more robust to errors (consistent with work on learning from mistakes (An et al., 2024) and critique finetuning (Wang et al., 2025)) and better at backtracking. An error-free response is not the only path leading to a correct final output - error and self-correction provides an equally important training signal as error-free demonstration.

Benefit of off-policy error. While on-policy errors better capture a model’s natural distribution, our use of off-policy injection is a deliberate design tradeoff that isolates self-correction from knowledge limitations. It also offers additional methodological advantages: (1) enables cross-model comparison using identical error traces, (2) allows targeted evaluation of specific reasoning skills, (3) supports efficient evaluation without per-model on-policy error sampling, and (4) simulates real-world scenarios where errors originate from external tools or systems.

Off-policy errors serve as a practical proxy for on-policy behavior when measuring behavioral failures like activation gaps rather than knowledge deficiencies. Therefore, we cannot directly compare blind spot magnitudes between off-policy and on-policy settings: on-policy errors conflate activation failures with genuine knowledge gaps, while off-policy errors do not. To address this limitation, our benchmark incorporates three realism levels progressing from synthetic to realistic errors. Figure 4 demonstrates a 0.6+ correlation in blind spot across realism levels. As realism increases, off-policy measurement better approximates on-policy behavior.

Understanding cognitive behavior via markers. Frequency analysis of correction markers is a scalable way to study cognitive behaviors present in pretraining data and post-training data. We believe that they can serve as important heuristics for pretraining and post-training data curation.

9 CONCLUSION AND LIMITATION

In this work, we identified and systematically measured the Self-Correction Blind Spot: non-reasoning LLMs fail to correct 64.5% of internal errors while successfully correcting identical external errors. This systematic failure has important implications for AI reliability and safety-critical applications. While our off-policy methodology does not directly replicate on-policy errors, which inherently conflate knowledge gaps with activation failures, it deliberately isolates self-correction capabilities from confounding factors, demonstrating the blind spot’s generality as a behavioral phenomenon across models and tasks. Importantly, it provides actionable insights to training recipe. We encourage future research to expand the benchmark to programming, logic and common sense reasoning, and to multilingual and multimodal reasoning.

REPRODUCIBILITY STATEMENT

Our experiments utilize various open source models, close source models, and datasets. Self-Correction Bench is available under an anonymous account in Hugging Face (click here). Our codes for constructing datasets, running the experiment, and building tables and graphs are released in Anonymous Github (click here).

DISCLOSURE OF LLM USAGE

LLMs are utilized to refine writing, including enhancing grammar, clarity, word choice and proof-reading. Additionally, LLMs are used to assist in writing some selected Python functions for graph plotting.

REFERENCES

- Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL <https://arxiv.org/abs/2412.08905>.
- Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. Learning from mistakes makes llm better reasoner, 2024. URL <https://arxiv.org/abs/2310.20689>.
- Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Ethan Dyer, Vlad Feinberg, Fangxiayu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Keanealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Mousaleem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. Palm 2 technical report, 2023. URL <https://arxiv.org/abs/2305.10403>.
- Anthropic. The claude 3 model family: Opus, sonnet, haiku, Mar 2024a. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.
- Anthropic. Introducing computer use, a new claude 3.5 sonnet, and claude 3.5 haiku, Oct 2024b. URL <https://www.anthropic.com/news/3-5-models-and-computer-use>.
- Anthropic. Introducing claude 4, May 2025. URL <https://www.anthropic.com/news/claude-4>.

- 594 Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Love-
595 nia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. A multi-
596 task, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interac-
597 tivity. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and
598 Adila Alfa Krisnadhi (eds.), *Proceedings of the 13th International Joint Conference on Natural*
599 *Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for*
600 *Computational Linguistics (Volume 1: Long Papers)*, pp. 675–718, Nusa Dua, Bali, November
601 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.ijcnlp-main.45. URL
602 <https://aclanthology.org/2023.ijcnlp-main.45/>.
- 603 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
604 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John
605 Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
- 607 Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu,
608 and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback, 2024.
609 URL <https://openreview.net/forum?id=pNkOx3IVWI>.
- 611 DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu,
612 Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu,
613 Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao
614 Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan,
615 Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao,
616 Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding,
617 Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang
618 Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai
619 Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang,
620 Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang,
621 Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang,
622 Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang,
623 R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng
624 Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing
625 Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanxia Zhao, Wen
626 Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong
627 Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu,
628 Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xi-
629 aosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia
630 Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng
631 Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong
632 Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong,
633 Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou,
634 Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying
635 Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda
636 Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu,
637 Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu
638 Zhang, and Zhen Zhang. Deepseek-rl: Incentivizing reasoning capability in llms via reinforce-
639 ment learning, 2025a. URL <https://arxiv.org/abs/2501.12948>.
- 640 DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Cheng-
641 gang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang,
642 Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting
643 Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui
644 Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi
645 Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li,
646 Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang,
647 Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun
648 Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan
649 Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J.
650 Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang,

- 648 Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng
649 Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shut-
650 ing Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanxia Zhao,
651 Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue
652 Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xi-
653 aokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin
654 Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang,
655 Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang
656 Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui
657 Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying
658 Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu,
659 Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan
660 Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F.
661 Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda
662 Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao,
663 Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li,
664 Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025b. URL
665 <https://arxiv.org/abs/2412.19437>.
- 666 Jessica Maria Echterhoff, Yao Liu, Abeer Alessa, Julian McAuley, and Zexue He. Cognitive
667 bias in decision-making with LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen
668 (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 12640–
669 12653, Miami, Florida, USA, November 2024. Association for Computational Linguistics.
670 doi: 10.18653/v1/2024.findings-emnlp.739. URL [https://aclanthology.org/2024.
671 findings-emnlp.739/](https://aclanthology.org/2024.findings-emnlp.739/).
- 672 Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL [https://
673 github.com/huggingface/open-r1](https://github.com/huggingface/open-r1).
- 674 Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao
675 Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghao-
676 ran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao
677 Chang. Omni-MATH: A universal olympiad level mathematic benchmark for large language
678 models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL
679 <https://openreview.net/forum?id=yaqPf0KAlN>.
- 680 Google Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality,
681 long context, and next generation agentic capabilities., June 2025. URL [https://storage.
682 googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf](https://storage.googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf).
- 683 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad
684 Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan,
685 Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Ko-
686 renev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava
687 Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux,
688 Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret,
689 Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius,
690 Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary,
691 Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab
692 AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco
693 Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind That-
694 tai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Kore-
695 vaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra,
696 Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Ma-
697 hadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu,
698 Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jong-
699 soo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala,
700 Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid
701 El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren
Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin,

702 Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi,
 703 Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew
 704 Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar
 705 Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev,
 706 Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan
 707 Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan,
 708 Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon
 709 Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit
 710 Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan
 711 Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell,
 712 Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Rapparthi, Sheng
 713 Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer
 714 Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman,
 715 Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mi-
 716 haylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor
 717 Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei
 718 Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang
 719 Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Gold-
 720 schlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning
 721 Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papanikos, Aaditya Singh,
 722 Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria,
 723 Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein,
 724 Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew
 725 Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie
 726 Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel,
 727 Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leon-
 728 hardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu
 729 Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Mon-
 730 talvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao
 731 Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia
 732 Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide
 733 Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le,
 734 Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily
 735 Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smoth-
 736 ers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni,
 737 Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia
 738 Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan,
 739 Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harri-
 740 son Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj,
 741 Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James
 742 Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jen-
 743 nifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang,
 744 Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Jun-
 745 jie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy
 746 Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang,
 747 Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell,
 748 Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa,
 749 Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias
 750 Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L.
 751 Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike
 752 Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari,
 753 Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan
 754 Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong,
 755 Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent,
 Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar,
 Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Ro-
 driguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy,
 Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin
 Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon,

- 756 Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ra-
757 maswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha,
758 Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal,
759 Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satter-
760 field, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj
761 Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo
762 Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook
763 Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Ku-
764 mar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov,
765 Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiao-
766 jian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia,
767 Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao,
768 Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhao-
769 duo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL
770 <https://arxiv.org/abs/2407.21783>.
- 771 Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna
772 Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu
773 Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su,
774 Wanxia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan
775 Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak,
776 Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia
777 Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill,
778 Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy,
779 Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models,
780 2025. URL <https://arxiv.org/abs/2506.04178>.
- 781 Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han,
782 Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench:
783 A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scienti-
784 fic problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the*
785 *62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Pa-*
786 *pers)*, pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguis-
787 tics. doi: 10.18653/v1/2024.acl-long.211. URL <https://aclanthology.org/2024.acl-long.211/>.
- 788 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn
789 Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset.
790 In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks*
791 *Track (Round 2)*, 2021. URL <https://openreview.net/forum?id=7Bywt2mQsCe>.
- 792 Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song,
793 and Denny Zhou. Large language models cannot self-correct reasoning yet, 2024. URL <https://arxiv.org/abs/2310.01798>.
- 794 Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong
795 Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large
796 language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on*
797 *Information Systems*, 43(2):1–55, January 2025. ISSN 1558-2868. doi: 10.1145/3703155. URL
798 <http://dx.doi.org/10.1145/3703155>.
- 799 Erik Jones and Jacob Steinhardt. Capturing failures of large language models via human cognitive
800 biases. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Ad-*
801 *vances in Neural Information Processing Systems*, 2022. URL [https://openreview.net/](https://openreview.net/forum?id=fc09Cgn-X-R)
802 [forum?id=fc09Cgn-X-R](https://openreview.net/forum?id=fc09Cgn-X-R).
- 803 Ryo Kamoi, Sarkar Snigdha Sarathi Das, Renze Lou, Jihyun Janice Ahn, Yilun Zhao, Xiaoxin
804 Lu, Nan Zhang, Yusen Zhang, Haoran Ranran Zhang, Sujeeth Reddy Vummanthala, Salika
805 Dave, Shaobo Qin, Arman Cohan, Wenpeng Yin, and Rui Zhang. Evaluating LLMs at de-
806 tecting errors in LLM responses. In *First Conference on Language Modeling*, 2024a. URL
807 <https://openreview.net/forum?id=dnwRScljXr>.

- 810 Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can LLMs actually
811 correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the*
812 *Association for Computational Linguistics*, 12:1417–1440, 2024b. doi: 10.1162/tacl.a.00713.
813 URL <https://aclanthology.org/2024.tacl-1.78/>.
- 814 Geunwoo Kim, Pierre Baldi, and Stephen Marcus McAleer. Language models can solve computer
815 tasks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL
816 <https://openreview.net/forum?id=M6OmjAZ4CX>.
- 817
818 Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang.
819 Benchmarking cognitive biases in large language models as evaluators. In Lun-Wei Ku, Andre
820 Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 517–545, Bangkok, Thailand, August 2024. Association for Computational
821 Linguistics. doi: 10.18653/v1/2024.findings-acl.29. URL [https://aclanthology.org/](https://aclanthology.org/2024.findings-acl.29/)
822 [2024.findings-acl.29/](https://aclanthology.org/2024.findings-acl.29/).
- 823
824 Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens,
825 Abdullah Barhoum, Duc Minh Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri,
826 David Alexandrovich Glushkov, Arnav Varma Dantuluri, Andrew Maguire, Christoph Schuh-
827 mann, Huu Nguyen, and Alexander Julian Mattick. Openassistant conversations - democratizing
828 large language model alignment. In *Thirty-seventh Conference on Neural Information Processing*
829 *Systems Datasets and Benchmarks Track*, 2023. URL [https://openreview.net/forum?](https://openreview.net/forum?id=VSJotgbPHF)
830 [id=VSJotgbPHF](https://openreview.net/forum?id=VSJotgbPHF).
- 831 Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli,
832 Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava,
833 Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Train-
834 ing language models to self-correct via reinforcement learning. In *The Thirteenth International*
835 *Conference on Learning Representations*, 2025. URL [https://openreview.net/forum?](https://openreview.net/forum?id=CjwERCAU7w)
836 [id=CjwERCAU7w](https://openreview.net/forum?id=CjwERCAU7w).
- 837
838 Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brah-
839 man, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxin Lyu, Yuling Gu, Saumya
840 Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Taffjord, Christo-
841 pher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Ha-
842 jishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on*
843 *Language Modeling*, 2025. URL <https://openreview.net/forum?id=iluGbFHHpH>.
- 844
845 Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Her-
846 nandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilë Lukošiušė, Karina
847 Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson,
848 Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Tim-
849 othy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan
850 Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought rea-
851 soning, 2023. URL <https://arxiv.org/abs/2307.13702>.
- 852
853 Jijie Li, Li Du, Hanyu Zhao, Bo wen Zhang, Liangdong Wang, Boyan Gao, Guang Liu, and Yonghua
854 Lin. Infinity instruct: Scaling instruction selection and synthesis to enhance language models,
855 2025. URL <https://arxiv.org/abs/2506.11116>.
- 856
857 Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan
858 Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *The Twelfth*
859 *International Conference on Learning Representations*, 2024. URL [https://openreview.](https://openreview.net/forum?id=v8L0pN6EOi)
860 [net/forum?id=v8L0pN6EOi](https://openreview.net/forum?id=v8L0pN6EOi).
- 861
862 Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Formalizing and bench-
863 marking prompt injection attacks and defenses. In *Proceedings of the 33rd USENIX Conference*
864 *on Security Symposium, SEC ’24, USA*, 2024. USENIX Association. ISBN 978-1-939133-44-1.
- 865
866 Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri
867 Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad
868 Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine:

- 864 Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Pro-*
865 *cessing Systems*, 2023. URL <https://openreview.net/forum?id=S37hOerQLB>.
866
- 867 Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality
868 in abstractive summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault
869 (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*,
870 pp. 1906–1919, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/
871 2020.acl-main.173. URL <https://aclanthology.org/2020.acl-main.173/>.
- 872 Jack Merullo, Noah A. Smith, Sarah Wiegrefe, and Yanai Elazar. On linear representations and pre-
873 training data frequency in language models. In *The Thirteenth International Conference on Learn-*
874 *ing Representations*, 2025. URL <https://openreview.net/forum?id=EDoD3DgivF>.
- 875 Meta. Llama 3.3, Dec 2024. URL [https://www.llama.com/docs/
876 model-cards-and-prompt-formats/llama3_3/](https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/).
877
- 878 Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, Apr 2025.
879 URL <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>.
- 880 Sajad Mousavi, Ricardo Luna Gutiérrez, Desik Rengarajan, Vineet Gundecha, Ashwin Ramesh
881 Babu, Avisek Naug, Antonio Guillen, and Soumyendu Sarkar. N-critics: Self-refinement of large
882 language models with ensemble of critics, 2023. URL [https://arxiv.org/abs/2310.
883 18679](https://arxiv.org/abs/2310.18679).
- 884 Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke
885 Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time
886 scaling, 2025. URL <https://arxiv.org/abs/2501.19393>.
887
- 888 Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. Alice in wonderland:
889 Simple tasks showing complete reasoning breakdown in state-of-the-art large language models,
890 2025. URL <https://arxiv.org/abs/2406.02061>.
- 891 OpenAI. Introducing gpt-4.1 in the api, Apr 2025. URL [https://openai.com/index/
892 gpt-4-1/](https://openai.com/index/gpt-4-1/).
893
- 894 OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Floren-
895 cia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red
896 Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Moham-
897 mad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher
898 Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brock-
899 man, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann,
900 Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis,
901 Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey
902 Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux,
903 Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila
904 Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix,
905 Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gib-
906 son, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan
907 Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hal-
908 lacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan
909 Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu,
910 Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun
911 Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Ka-
912 mali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook
913 Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel
914 Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen
915 Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel
916 Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez,
917 Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv
Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney,
Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick,
Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel

- 918 Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Ra-
 919 jeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe,
 920 Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel
 921 Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe
 922 de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny,
 923 Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl,
 924 Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra
 925 Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders,
 926 Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Sel-
 927 sam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor,
 928 Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky,
 929 Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang,
 930 Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Pre-
 931 ston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vi-
 932 jayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan
 933 Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng,
 934 Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Work-
 935 man, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming
 936 Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao
 937 Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL
<https://arxiv.org/abs/2303.08774>.
- 938 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong
 939 Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kel-
 940 ton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike,
 941 and Ryan Lowe. Training language models to follow instructions with human feedback. In
 942 Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neu-
 943 ral Information Processing Systems*, 2022. URL [https://openreview.net/forum?id=](https://openreview.net/forum?id=TG8KACxEON)
 944 [TG8KACxEON](https://openreview.net/forum?id=TG8KACxEON).
- 945
- 946 Emily Pronin, Daniel Y. Lin, and Lee Ross. The bias blind spot: Perceptions of bias in
 947 self versus others. *Personality and Social Psychology Bulletin*, 28(3):369–381, 2002. doi:
 948 [10.1177/0146167202286008](https://doi.org/10.1177/0146167202286008). URL <https://doi.org/10.1177/0146167202286008>.
- 949
- 950 Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan
 951 Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang,
 952 Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin
 953 Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li,
 954 Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang,
 955 Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.
 956 URL <https://arxiv.org/abs/2412.15115>.
- 957 Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining
 958 term frequencies on few-shot numerical reasoning. In Yoav Goldberg, Zornitsa Kozareva, and
 959 Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp.
 960 840–854, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Lin-
 961 guistics. doi: 10.18653/v1/2022.findings-emnlp.59. URL [https://aclanthology.org/](https://aclanthology.org/2022.findings-emnlp.59/)
 962 [2022.findings-emnlp.59/](https://aclanthology.org/2022.findings-emnlp.59/).
- 963
- 964 Matthew Renze. The effect of sampling temperature on problem solving in large language mod-
 965 els. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association*
 966 *for Computational Linguistics: EMNLP 2024*, pp. 7346–7356, Miami, Florida, USA, November
 967 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.432.
 968 URL <https://aclanthology.org/2024.findings-emnlp.432/>.
- 969 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
 970 Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathe-
 971 matical reasoning in open language models, 2024. URL [https://arxiv.org/abs/2402.](https://arxiv.org/abs/2402.03300)
 03300.

- 972 Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli,
973 and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Pro-*
974 *ceedings of the 40th International Conference on Machine Learning, ICML'23*. JMLR.org, 2023.
975
- 976 Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Re-
977 flexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on*
978 *Neural Information Processing Systems*, 2023. URL [https://openreview.net/forum?](https://openreview.net/forum?id=vAElhFcKW6)
979 [id=vAElhFcKW6](https://openreview.net/forum?id=vAElhFcKW6).
- 980 Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute
981 optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth Interna-*
982 *tional Conference on Learning Representations*, 2025. URL [https://openreview.net/](https://openreview.net/forum?id=4FWAwZtd2n)
983 [forum?id=4FWAwZtd2n](https://openreview.net/forum?id=4FWAwZtd2n).
- 984 Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. Prmbench: A fine-grained
985 and challenging benchmark for process-level reward models, 2025. URL [https://arxiv.](https://arxiv.org/abs/2501.03124)
986 [org/abs/2501.03124](https://arxiv.org/abs/2501.03124).
987
- 988 Mistral AI Team. Mistral small 3, Jan 2025. URL [https://mistral.ai/news/](https://mistral.ai/news/mistral-small-3)
989 [mistral-small-3](https://mistral.ai/news/mistral-small-3).
- 990 Teknium. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants, 2023.
991 URL <https://huggingface.co/datasets/teknium/OpenHermes-2.5>.
992
- 993 Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. LLMs cannot find
994 reasoning errors, but can correct them given the error location. In Lun-Wei Ku, Andre Mar-
995 tins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL*
996 *2024*, pp. 13894–13908, Bangkok, Thailand, August 2024. Association for Computational Lin-
997 guistics. doi: 10.18653/v1/2024.findings-acl.826. URL [https://aclanthology.org/](https://aclanthology.org/2024.findings-acl.826/)
998 [2024.findings-acl.826/](https://aclanthology.org/2024.findings-acl.826/).
- 999 Yubo Wang, Xiang Yue, and Wenhui Chen. Critique fine-tuning: Learning to critique is more effec-
1000 tive than learning to imitate, 2025. URL <https://arxiv.org/abs/2501.17703>.
1001
- 1002 Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: how does llm safety training fail?
1003 In *Proceedings of the 37th International Conference on Neural Information Processing Systems*,
1004 *NIPS '23*, Red Hook, NY, USA, 2023. Curran Associates Inc.
- 1005 Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi,
1006 Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick
1007 von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger,
1008 Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural
1009 language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Confer-*
1010 *ence on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–
1011 45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.
1012 [emnlp-demos.6](https://aclanthology.org/2020.emnlp-demos.6/). URL <https://aclanthology.org/2020.emnlp-demos.6/>.
- 1013 An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li,
1014 Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang,
1015 Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jin-
1016 gren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin
1017 Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao,
1018 Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wen-
1019 bin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng
1020 Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu,
1021 Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL
1022 <https://arxiv.org/abs/2407.10671>.
- 1023 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang
1024 Gao, Chengen Huang, Chenxu Lv, Chuji Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu,
1025 Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin
Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang,

1026 Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui
 1027 Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang
 1028 Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger
 1029 Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan
 1030 Qiu. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.

1031 Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. How language model
 1032 hallucinations can snowball. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian
 1033 Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st
 1034 International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning
 1035 Research*, pp. 59670–59684. PMLR, 21–27 Jul 2024. URL [https://proceedings.mlr.
 1036 press/v235/zhang24ay.html](https://proceedings.mlr.press/v235/zhang24ay.html).

1037 Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu,
 1038 Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical rea-
 1039 soning, 2025. URL <https://arxiv.org/abs/2412.06559>.

1040

1041 APPENDIX

1042 A DATASET CONSTRUCTION

1043

1044 Table 5: Dataset comparison

1045

Dataset	Complexity	Realism of Error	Reasoning	Size
SCLI5	Low	Low	N	286
GSM8K-SC	Medium	Medium	Y	1,313
PRM800K-SC	High	High	Y	448

1046

1047

1048 Table 6: Task composition of SCLI5

1049

Task	Count	Error Type	Question and Answer
Add one	20	Off-by-one	Q: What is the answer of $1 + 1$? A: The answer is 3.
Subtract one	20	Off-by-one	Q: What is the answer of $3 - 1$? A: The answer is 1.
Next character	52	Off-by-one	Q: What letter comes after A? A: The answer is C.
Previous character	52	Off-by-one	Q: What letter comes before C? A: The answer is A.
Larger number	71	Flip	Q: Which one is smaller, 1 or 2? A: The answer is 2.
Smaller number	71	Flip	Q: Which one is larger, 2 or 5? A: The answer is 2.

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

Table 7: Error composition of GSM8K-SC

Category	Description
Problem Representation Errors	These errors arise when the solver misunderstands or misinterprets the problem’s requirements or given information. This can involve misreading the problem statement, confusing the relationships between quantities, or failing to grasp what is being asked.
Planning Errors	These occur when the solver devises an incorrect or incomplete strategy to tackle the problem. This might include choosing the wrong operations, setting up flawed equations, or overlooking key components of the problem.
Execution Errors	These are mistakes made while carrying out the planned steps, such as errors in calculations, misapplication of mathematical rules, or procedural slip-ups, even if the plan itself is sound.

B FIGURES

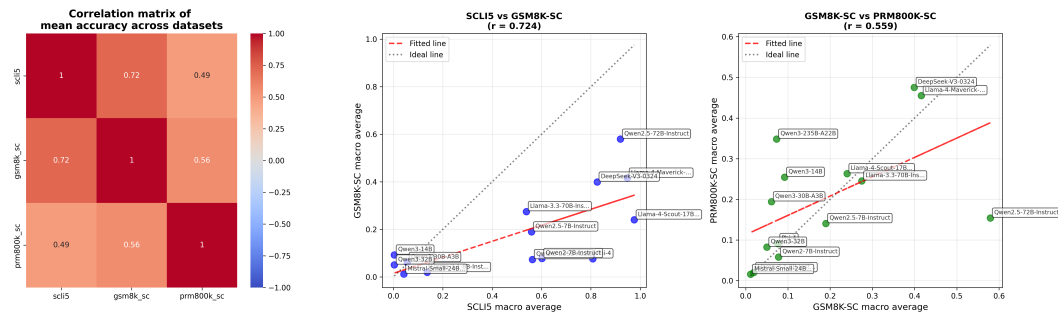


Figure 4: *left*: Mean accuracy correlation matrix across datasets *middle*: Scatter plot between SCLIS vs GSM8K-SC *right*: Scatter plot between GSM8K-SC vs PRM800K-SC

BCA: Before commit an answer

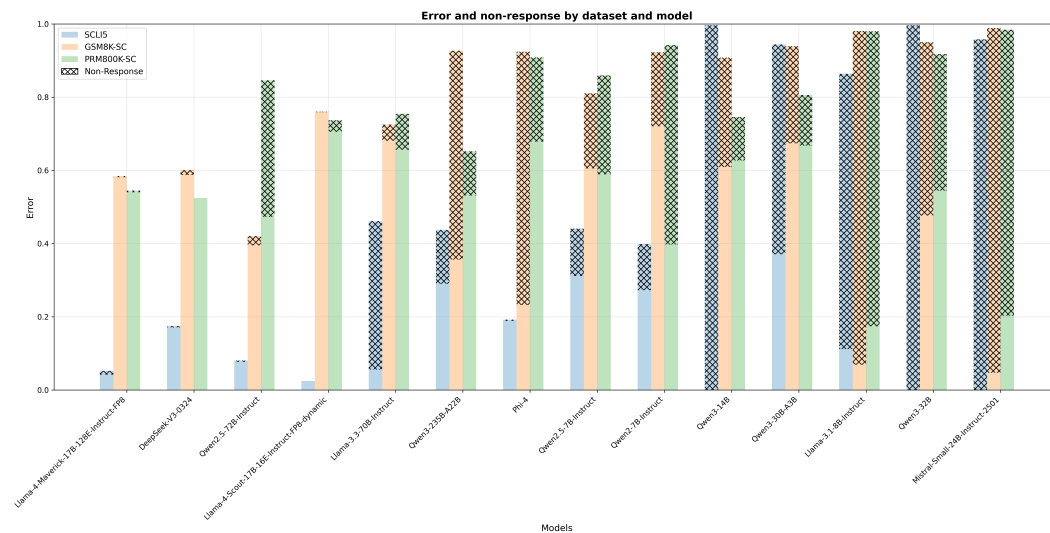


Figure 5: Summary of error and empty response across models

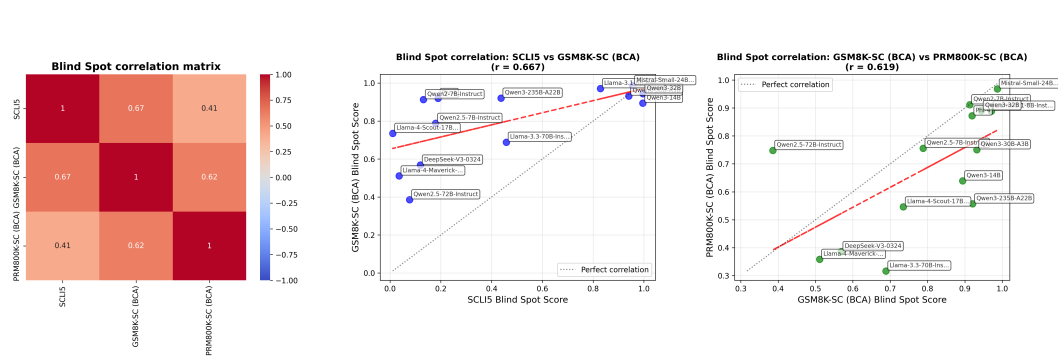


Figure 6: *left*: Blind spot correlation matrix *middle*: Scatter plot between SCLIS vs GSM8K-SC *right*: Scatter plot between GSM8K-SC vs PRM800K-SC
BCA: Before commit an answer

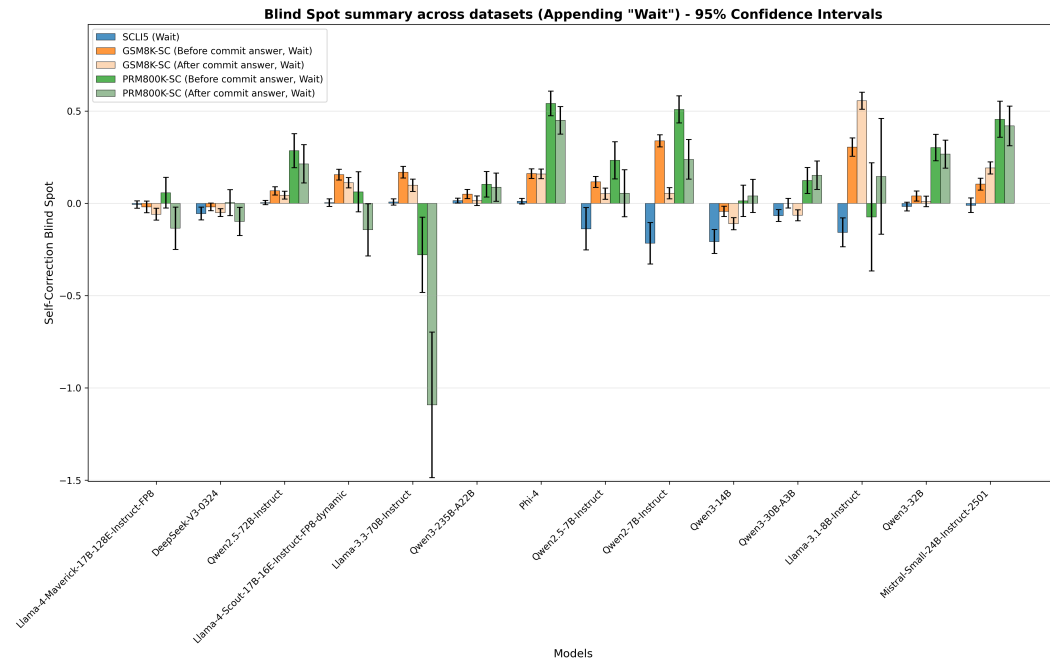


Figure 7: Self-Correction Blind Spot and 95% confidence interval across non-reasoning models after appending “Wait”

Table 8: Mean accuracy and relative change after appending various correction markers

Correction Markers	SCLIS	GSM8K-SC	PRM800K-SC
Internal Error (Baseline)	0.499 (0%)	0.183 (0%)	0.200 (0%)
External Error	0.910 (+82.5%)	0.881 (+382.1%)	0.620 (+210.3%)
“Wait”	0.957 (+91.9%)	0.796 (+335.1%)	0.504 (+152.0%)
“But”	0.922 (+85.0%)	0.611 (+234.2%)	0.430 (+114.8%)
“However”	0.897 (+79.8%)	0.602 (+229.0%)	0.438 (+119.3%)

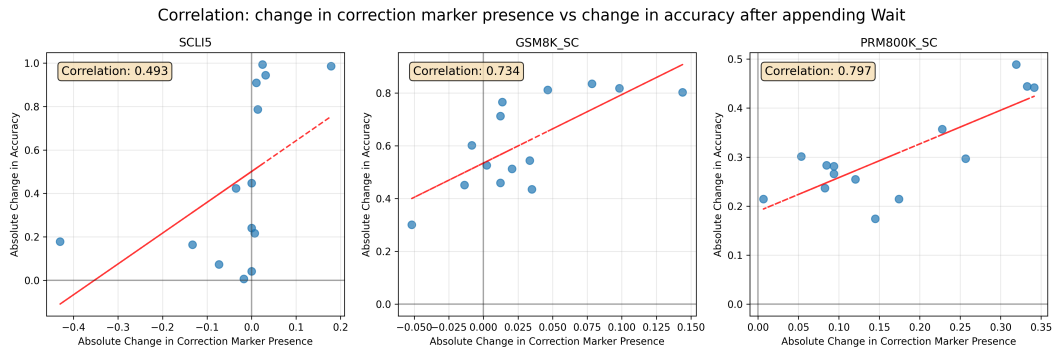


Figure 8: Correlation of absolute change in keyword presence vs absolute change in accuracy - original vs appending “Wait”

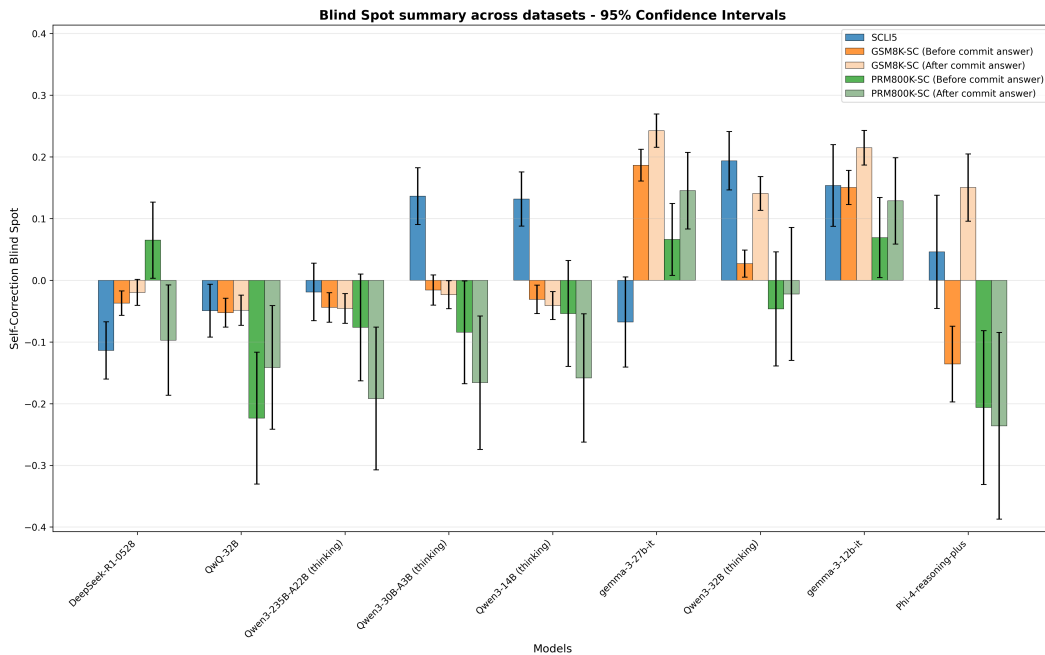


Figure 9: Self-Correction Blind Spot and 95% confidence interval across reasoning models

Table 9: Most common first word and relative frequency generated by reasoning models

Model	SCLi5	GSM8K-SC	PRM800K-SC
QwQ-32B	(‘Wait’, 0.377)	(‘Wait’, 0.725)	(‘Wait’, 0.768)
Qwen3-14B (thinking)	(‘In’, 1.0)	(‘Wait’, 0.38)	(‘Therefore’, 0.219)
Qwen3-32B (thinking)	(‘After’, 1.0)	(‘The’, 0.288)	(‘I’, 0.189)
Qwen3-30B-A3B (thinking)	(‘Wait’, 0.312)	(‘Therefore’, 0.25)	(‘So’, 0.195)
Qwen3-235B-A22B (thinking)	(‘**Step-by-step’, 0.292)	(‘Wait’, 0.198)	(‘Therefore’, 0.256)
DeepSeek-R1-0528	(‘No’, 0.324)	(‘But’, 0.267)	(‘But’, 0.486)
gemma-3-12b-it	(‘The’, 0.284)	(‘The’, 0.239)	(‘Alternatively’, 0.205)
gemma-3-27b-it	(‘Here’s’, 0.31)	(‘Let’, 0.256)	(‘However’, 0.292)
phi-4-reasoning-plus	(‘Wait’, 0.861)	(‘Wait’, 0.677)	(‘However’, 0.217)

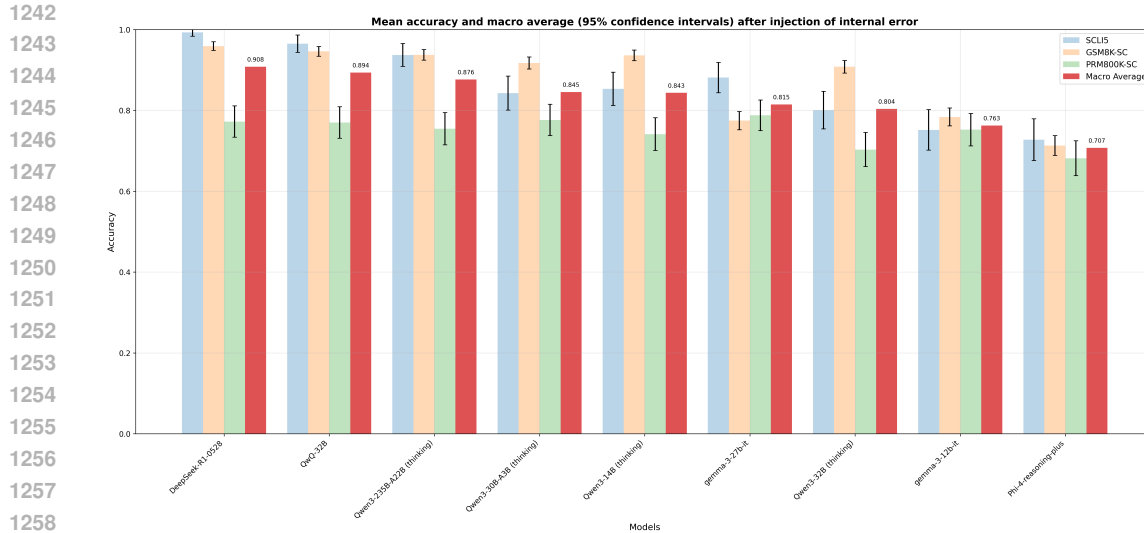


Figure 10: Summary of mean accuracy across reasoning models

Table 10: Self-Correction Blind Spot in close-source model

Model	Dataset	Mean Accuracy		Blind Spot
		External Error	Internal Error	
claude-3-5-haiku-20241022	SCLI5	0.822	0.692	0.157
	GSM8k-SC	0.884	0.328	0.629
	PRM800K-SC	0.382	0.08	0.789
claude-sonnet-4-20250514	SCLI5	0.997	0.427	0.572
	GSM8k-SC	0.974	0.655	0.328
	PRM800K-SC	0.877	0.576	0.344

Table 11: Self-Correction Blind Spot in other domains in BIG-Bench Mistake

Model	Dataset	Mean Accuracy		Blind Spot	Size
		External Error	Internal Error		
Meta-Llama-3.1-8B-Instruct	Tracking shuffled objects	0.246	0.073	0.703	260
	Logical deduction	0.208	0.14	0.328	294
Qwen3-14B	Tracking shuffled objects	0.996	0.096	0.904	260
	Logical deduction	0.854	0.262	0.693	294
Mistral-Small-24B-Instruct-2501	Tracking shuffled objects	0.554	0.2	0.639	260
	Logical deduction	0.449	0.255	0.432	294
Llama-3.3-70B-Instruct	Tracking shuffled objects	0.931	0.127	0.864	260
	Logical deduction	0.694	0.371	0.466	294
Model SFTed with error and self-correction data					
DeepSeek-R1-Distill-Llama-70B	Tracking shuffled objects	0.973	0.95	0.024	260
	Logical deduction	0.905	0.694	0.233	294

Table 12: Mean accuracy when on-policy errors in ProcessBench are presented externally

model	ProcessBench		Without OlympiadBench and Omni-Math	
	Accuracy (95% CI)	Size	Accuracy (95% CI)	Size
Qwen2-7B-Instruct	0.061 ± 0.033	198	0.105 ± 0.059	105
Qwen2.5-7B-Instruct	0.064 ± 0.037	172	0.125 ± 0.082	64
Llama-3.1-8B-Instruct	0.099 ± 0.035	274	0.124 ± 0.063	105

C SENSITIVITY ANALYSIS

C.1 RESULT OF DIFFERENT TEMPERATURE

Apart from using models' most confident prediction, we use temperature of 0.0 for 3 reasons:

- More deterministic⁶ output eliminates sampling variance as a confounding factor.
- It enables standardized comparison across models with different temperature calibrations.
- Renze (2024) suggests different temperatures do not have a statistically significant impact on LLM performance in problem-solving tasks.

We also report results using a temperature of 0.6 below and the result does not change our conclusion.

Table 13: Mean accuracy and 95% confidence interval of models at temperature 0.6

Model	SCLI5	GSM8K-SC	PRM800K-SC
Llama-4-Maverick-17B-128E-Instruct-FP8	0.954 ± 0.024	0.424 ± 0.027	0.469 ± 0.046
DeepSeek-V3-0324	0.874 ± 0.039	0.42 ± 0.027	0.504 ± 0.046
Qwen2.5-72B-Instruct	0.902 ± 0.035	0.574 ± 0.027	0.165 ± 0.034
Llama-4-Scout-17B-16E-Instruct	0.976 ± 0.018	0.248 ± 0.023	0.272 ± 0.041
Llama-3.3-70B-Instruct	0.496 ± 0.058	0.273 ± 0.024	0.243 ± 0.04
Qwen3-235B-A22B	0.57 ± 0.057	0.091 ± 0.016	0.4 ± 0.045
phi-4	0.794 ± 0.047	0.093 ± 0.016	0.116 ± 0.03
Qwen2.5-7B-Instruct	0.563 ± 0.058	0.183 ± 0.021	0.127 ± 0.031
Qwen2-7B-Instruct	0.601 ± 0.057	0.071 ± 0.014	0.065 ± 0.023
Qwen3-14B	0.007 ± 0.01	0.101 ± 0.016	0.27 ± 0.041
Qwen3-30B-A3B	0.108 ± 0.036	0.07 ± 0.014	0.232 ± 0.039
Qwen3-32B	0.038 ± 0.022	0.068 ± 0.014	0.105 ± 0.028
Meta-Llama-3.1-8B-Instruct	0.182 ± 0.045	0.025 ± 0.008	0.022 ± 0.014
Mistral-Small-24B-Instruct-2501	0.122 ± 0.038	0.02 ± 0.008	0.038 ± 0.018

⁶Temperature of 0.0 will not generate fully deterministic result due to finite precision.

1350 C.2 PRM800K-SC RESULT IN 4,096 TOKEN BUDGET
1351

1352 To ensure a fair comparison between internal and external error correction, and across models, we
 1353 maintain a fixed token budget of 1,024 across all conditions. This design choice partly isolates
 1354 self-correction capabilities from the effect of test time compute, providing a more rigorous test of
 1355 the blind spot phenomenon. We also report our results of PRM800-SC with a fixed tokens budget
 1356 of 4,096 below, which does not change our conclusion. We do not report the result of SCL15 and
 1357 GSM8K-SC as the ratio of model responses exceeding 1,024 tokens is immaterial.

1358 Table 14: Mean accuracy of models in PRM800K-SC at different compute budget
1359

1360 Model	1361 External Error		1362 Internal Error		1363 Appending “Wait”	
	1364 1,024	1365 4,096	1,024	4,096	1,024	4,096
1366 Llama-4-Maverick-17B-128E-Instruct-FP8	0.71	0.721	0.455	0.458	0.67	0.676
1367 DeepSeek-V3-0324	0.775	0.938	0.475	0.509	0.772	0.821
1368 Qwen2.5-72B-Instruct	0.612	0.614	0.154	0.161	0.438	0.449
1369 Llama-4-Scout-17B-16E-Instruct	0.58	0.578	0.263	0.257	0.545	0.542
1370 Llama-3.3-70B-Instruct	0.359	0.366	0.246	0.257	0.46	0.469
1371 Qwen3-235B-A22B	0.786	0.806	0.348	0.368	0.705	0.732
1372 phi-4	0.714	0.719	0.092	0.092	0.328	0.337
1373 Qwen2.5-7B-Instruct	0.576	0.569	0.141	0.141	0.442	0.444
1374 Qwen2-7B-Instruct	0.658	0.65	0.058	0.058	0.324	0.333
1375 Qwen3-14B	0.705	0.743	0.254	0.268	0.696	0.746
1376 Qwen3-30B-A3B	0.779	0.817	0.194	0.19	0.683	0.712
1377 Qwen3-32B	0.754	0.781	0.083	0.085	0.527	0.522
1378 Meta-Llama-3.1-8B-Instruct	0.181	0.183	0.02	0.02	0.194	0.203
1379 Mistral-Small-24B-Instruct-2501	0.496	0.498	0.016	0.016	0.27	0.277

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

1404 D PROMPT

1405

1406 D.1 GENERATING GSM8K-SC

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

1417

1418

1419

1420

1421

1422

1423

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

```
from pydantic import BaseModel
```

```
class ReasoningWithMistake(BaseModel):
    reasoning_steps_with_one_mistake: List[str]
    mistake_step: int
    type_of_mistake: str
    description_of_mistake: str
    incorrect_answer: str
```

You are a helpful assistant that follow instructions. Output in JSON format.

```
<question>
{question}
</question>
```

```
<reasoning_steps>
{reasoning_steps}
</reasoning_steps>
```

```
<answer>
{answer}
</answer>
```

```
<type_of_mistake>
{error_type}: {error_description}
</type_of_mistake>
```

Your task is to introduce one mistake in step {mistake_step} in <reasoning_steps> and arrive at an answer different from <answer>. You will output:

- <reasoning_steps> with mistake
- the step that contains the mistake
- type of the mistake
- description of the mistake
- incorrect answer

Figure 11: Output schema, system prompt and prompt for generating GSM8K-SC dataset

```

1458
1459
1460     from pydantic import BaseModel
1461
1462     class Calculation(BaseModel):
1463         incorrect_answer: float
1464
1465
1466     You are a helpful assistant that follow instructions. Output in
1467     JSON format.
1468
1469
1470
1471     <question>
1472     {question}
1473     </question>
1474
1475     <description_of_mistake>
1476     {description_of_mistake}
1477     </description_of_mistake>
1478
1479     <incorrect_reasoning>
1480     {incorrect_reasoning}
1481     </incorrect_reasoning>
1482
1483     You are given a question, a description of the mistake, and the
1484     resulting incorrect reasoning.
1485     Your task is to follow the incorrect reasoning to arrive at the
1486     incorrect answer.

```

Figure 12: Output schema, system prompt and prompt for validating GSM8K-SC dataset

D.2 AUTOMATIC EVALUATION

```

1493
1494     from pydantic import BaseModel
1495
1496     class Evaluation(BaseModel):
1497         is_correct_answer: bool
1498         is_correct_reasoning: bool
1499         is_backtracking: bool
1500         is_admitting_mistake: bool
1501
1502
1503     You are a helpful assistant that follow instructions. Output in
1504     JSON format.
1505
1506

```

Figure 13: Output schema and system prompt for automatic evaluation of error in model

1511

```

1512
1513 # FOR SCLI5
1514 <question>
1515 {question}
1516 </question>
1517
1518 <golden_answer>
1519 {golden_answer}
1520 </golden_answer>
1521
1522 <given_wrong_answer>
1523 {given_wrong_answer}
1524 </given_wrong_answer>
1525
1526 <completion_from_model>
1527 {completion_from_model}
1528 </completion_from_model>
1529
1530 The model was provided a wrong answer <given_wrong_answer>, and is
1531 required to self correct and arrive at the golden answer.
1532 You have to assess if <completion_from_model> :
1533 - has backtracked
1534 - has finally provided the answer that matches the <golden_answer>
1535 - has provided correct reasoning
1536 - has admitted the mistake.
1537
1538
1539 # FOR GSM8K-SC AND PRM800K-SC
1540 <question>
1541 {question}
1542 </question>
1543
1544 <golden_answer>
1545 {golden_answer}
1546 </golden_answer>
1547
1548 <given_wrong_reasoning>
1549 {given_wrong_reasoning}
1550 </given_wrong_reasoning>
1551
1552 <completion_from_model>
1553 {completion_from_model}
1554 </completion_from_model>
1555
1556 The model was provided a wrong step in the reasoning process <
1557 given_wrong_reasoning>, and is required to self correct and arrive
1558 at the golden answer.
1559 You have to assess if <completion_from_model> :
1560 - has backtracked
1561 - has finally provided the answer that matches the <golden_answer>
1562 - has provided correct reasoning
1563 - has admitted the mistake.
1564
1565

```

Figure 14: Prompt for automatic evaluation of error in model

```

1566
1567
1568     from pydantic import BaseModel
1569
1570     class EvaluationErrorInUser(BaseModel):
1571         is_correct_answer: bool
1572         is_correct_critique: bool
1573
1574
1575     You are a helpful assistant that follow instructions. Output in
1576     JSON format.
1577
1578
1579

```

Figure 15: Output schema and system prompt for automatic evaluation of error in user

```

1580
1581
1582
1583
1584     # FOR SCLI5
1585     <question_and_user_answer>
1586     {question_and_user_answer}
1587     </question_and_user_answer>
1588
1589     <golden_answer>
1590     {golden_answer}
1591     </golden_answer>
1592
1593     <response_from_model>
1594     {response_from_model}
1595     </response_from_model>
1596
1597     The model was provided with <question_and_user_answer> from user.
1598     You have to assess if <response_from_model> :
1599     - contains correct answer that matches the <golden_answer>
1600     - contains correct critique of <question_and_user_answer>
1601
1602     # FOR GSM8K-SC AND PRM800K-SC
1603     <question_and_user_reasoning>
1604     {question_and_user_reasoning}
1605     </question_and_user_reasoning>
1606
1607     <golden_answer>
1608     {golden_answer}
1609     </golden_answer>
1610
1611     <response_from_model>
1612     {response_from_model}
1613     </response_from_model>
1614
1615     The model was provided with <question_and_user_reasoning> from user
1616     .
1617     You have to assess if <response_from_model> :
1618     - contains correct answer that matches the <golden_answer>
1619     - contains correct critique of <question_and_user_reasoning>

```

Figure 16: Prompt for automatic evaluation of error in user

1620 E EXAMPLE OF SELF-CORRECTION BENCH

1621

1622 E.1 SCLI5

1623

1624

1625

1626

1627

1628

1629

1630

1631

1632

1633

1634

1635

1636

1637

1638

1639

1640

1641

1642

1643

1644

1645

1646

1647

1648

1649

1650

1651

1652

1653

1654

1655

1656

1657

1658

1659

1660

1661

1662

1663

1664

1665

1666

1667

1668

1669

1670

1671

1672

1673

```

# error in user prompt
[
  {
    "content": "What is the answer of 20 + 1? The answer is 22.",
    "role": "user"
  }
]

# error in model response
[
  {
    "content": "What is the answer of 20 + 1?",
    "role": "user"
  },
  {
    "content": "The answer is 22.",
    "role": "assistant"
  }
]

```

1644 E.2 GSM8K-SC

1645

1646

1647

1648

1649

1650

1651

1652

1653

1654

1655

1656

1657

1658

1659

1660

1661

1662

1663

1664

1665

1666

1667

1668

1669

1670

1671

1672

1673

```

# error in user prompt
[
  {
    "content": "Harry slept 9 hours last night. His friend James slept only 2/3 of what Harry slept. How many more hours did Harry sleep than James? James slept 9 hours / 2/3 = 13.5 hours. So, Harry slept 9 - 13.5 = -4.5 hours more than James.",
    "role": "user"
  }
]

# error in model response
[
  {
    "content": "Harry slept 9 hours last night. His friend James slept only 2/3 of what Harry slept. How many more hours did Harry sleep than James?",
    "role": "user"
  },
  {
    "content": "James slept 9 hours / 2/3 = 13.5 hours. So, Harry slept 9 - 13.5 = -4.5 hours more than James.",
    "role": "assistant"
  }
]

```

E.3 PRM800K-SC

```

1674
1675
1676
1677
1678 # error in user prompt
1679 [
1680     {
1681         "content": "A Senate committee has 5 Democrats, 5 Republicans, and 1
1682 Independent. In how many ways can they sit around a circular table if all
1683 the members of each party all sit next to each other? (Two seatings are
1684 considered equivalent if one is a rotation of the other.) I notice that
1685 there are three groups of people: Democrats, Republicans, and Independent. I
1686 wonder how many ways I can arrange these groups around the circle, ignoring
1687 the order within each group for now. I recall that the number of ways to
1688 arrange n distinct objects around a circle is (n-1)!, since we can fix one
1689 object and then permute the rest. So, for the three groups, there are (3-1)!
1690 = 2! = 2 ways to arrange them around the circle. For example, one way is D-
1691 R-I, and the other way is R-D-I, where D stands for Democrats, R for
1692 Republicans, and I for Independent. Now, I need to consider the order within
1693 each group. For the Democrats, there are 5! ways to order them in a line,
1694 but only half of them are distinct around a circle, since reversing the
1695 order gives the same seating. So, there are 5!/2 = 60 ways to order the
1696 Democrats around a circle. Similarly, there are 60 ways to order the
1697 Republicans around a circle. For the Independent, there is only one way to
1698 order him or her around a circle. So, to get the total number of ways to
1699 seat the committee, I need to multiply the number of ways to arrange the
1700 groups by the number of ways to order each group. That gives me 2 x 60 x 60
1701 x 1 = 7200 ways.",
1702     "role": "user"
1703     }
1704 ]
1705 # error in model response
1706 [
1707     {
1708         "content": "A Senate committee has 5 Democrats, 5 Republicans, and 1
1709 Independent. In how many ways can they sit around a circular table if all
1710 the members of each party all sit next to each other? (Two seatings are
1711 considered equivalent if one is a rotation of the other.)",
1712         "role": "user"
1713     },
1714     {
1715         "content": "I notice that there are three groups of people: Democrats,
1716 Republicans, and Independent. I wonder how many ways I can arrange these
1717 groups around the circle, ignoring the order within each group for now. I
1718 recall that the number of ways to arrange n distinct objects around a circle
1719 is (n-1)!, since we can fix one object and then permute the rest. So, for
1720 the three groups, there are (3-1)! = 2! = 2 ways to arrange them around the
1721 circle. For example, one way is D-R-I, and the other way is R-D-I, where D
1722 stands for Democrats, R for Republicans, and I for Independent. Now, I need
1723 to consider the order within each group. For the Democrats, there are 5!
1724 ways to order them in a line, but only half of them are distinct around a
1725 circle, since reversing the order gives the same seating. So, there are 5!/2
1726 = 60 ways to order the Democrats around a circle. Similarly, there are 60
1727 ways to order the Republicans around a circle. For the Independent, there is
1728 only one way to order him or her around a circle. So, to get the total
1729 number of ways to seat the committee, I need to multiply the number of ways
1730 to arrange the groups by the number of ways to order each group. That gives
1731 me 2 x 60 x 60 x 1 = 7200 ways.",
1732         "role": "assistant"
1733     }
1734 ]

```