

Self-Correction Bench: Uncovering and Addressing the Self-Correction Blind Spot in Large Language Models

Ken Tsui

Independent Researcher

kenhktsui@gmail.com

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, but studying it requires disentangling *activation failure* from *knowledge deficiency*: when a model fails to correct an error, is it because it cannot, or because it does not? We introduce Self-Correction Bench, a controlled evaluation framework that isolates this distinction by injecting the same error as either an external (user-attributed) or internal (model-attributed) error, keeping all other context identical. Testing 14 open-source non-reasoning models reveals a 64.5% **Self-Correction Blind Spot**: models correct external errors but fail on identical internal ones, proving the capability exists but is not activated. On models' own naturally generated errors, a measurable share of what a model fails to catch in its own output is caught when the identical error is presented externally. We trace the cause to post-training data composition: supervised fine-tuning datasets lack error-correction sequences, and fine-tuning with as few as 5,306 such traces already reduces the blind spot by 76.0%. Mechanistically, we identify a transferable *conversational-role direction* in representation space that causally gates self-correction. Appending "Wait" requires no training yet reduces the blind spot by 89.3%, and operates through a nearly independent pathway, indicating that correction activation is not reducible to this single mechanism.

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, 2024; 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 (Nezhurina et al., 2025) 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. Understanding why self-correction fails is critical for deploying LLMs in settings where errors carry real consequences.

Apart from the rarity of errors, a central difficulty in studying self-correction is that failure is ambiguous: when a model does not correct an error, it may lack the knowledge to do so, or it may possess the knowledge but fail to activate it. These two explanations have very different implications. Knowledge deficiency requires stronger base capabilities; activation failure requires better post-training signals or inference-time interventions. Prior work has largely been unable to distinguish between them.

We isolate this distinction by constructing Self-Correction Bench, which systematically injects the same error into both the user prompt (external error) and the model generation (internal error). The error content and surrounding context are identical; only the conversational role differs. If a model corrects the external error but not the internal one, knowledge deficiency is ruled out.

Testing 14 open-source non-reasoning models, we find a 64.5% average *Self-Correction Blind Spot*: models reliably correct external errors but fail on identical internal ones. This gap is consistent across model families, scales, and task complexities ranging from trivial arithmetic to multi-step mathematical reasoning. We further validate the finding in closed-source frontier models, non-mathematical domains (logic, object tracking), and we show that models catch a share of their own naturally generated errors once the same error is presented externally.

We trace the cause to post-training data: supervised fine-tuning datasets contain near-zero correction sequences, while reasoning-model training data contains orders of magnitude more. Fine-tuning with just 5,306 error-correction traces, a minimal intervention requiring only two epochs of LoRA, reduces the blind spot by 76.0%. Going further, we extract the activation difference induced by the internal-external distinction and find it encodes a transferable *conversational-role direction* in representation space. Steering along this direction causally activates self-correction on internal errors, confirming that conversational role gates access to latent correction capabilities. Appending “Wait” requires no training yet reduces the blind spot by 89.3%, narrowing the gap between reasoning and non-reasoning models. It operates through a nearly independent pathway from the conversational-role direction, indicating that the conversational-role direction does not fully account for correction activation.

Our contributions are threefold.

- A controlled methodology that disentangles self-correction activation from knowledge limitations, revealing a 64.5% Self-Correction Blind Spot.
- Causal evidence tracing the blind spot to post-training data composition.
- Mechanistic evidence, demonstrated in two model families at 7–8B scale, that a transferable conversational-role direction in representation space gates self-correction on internal errors. The behavioral blind spot itself is observed across all 14 models and closed-source frontier models.

These results advance our understanding of what suppresses LLM self-correction and provide a practical solution to improve their reliability in real-world use.

2 Related work

Intrinsic self-correction in LLMs. Recent work explores intrinsic self-correction via self-feedback (Shinn et al., 2023; Madaan et al., 2023; Kim et al., 2023; Kamoi et al., 2024b) or critic ensemble (Mousavi et al., 2023), but limitations persist. Feedback quality suffers without oracle labels (Huang et al., 2024): prior studies attribute this to poor error localization (Tyen et al., 2024) and detection (Kamoi et al., 2024a). Most approaches use multi-step prompting, whereas we focus on single-pass self-correction and study limitations from a cognitive perspective. Related work using RL (Kumar et al., 2025) or training signals from ground truth (DeepSeek-AI et al., 2025a) induces self-correction without characterizing what factors drive it.

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

Hallucination snowballing. Zhang et al. (2024) demonstrate that once LLMs hallucinate, subsequent tokens often align with the initial error, suggesting inherent limits to self-correction during generation. We explain this phenomenon by identifying the blind spot.

Test-time interventions. Recent efforts have shifted compute from training to test time (Snell et al., 2025), yielding improved performance (e.g. Muennighoff et al. (2025) appends

“Wait” to force longer reasoning traces on fine-tuned models), but improvement mechanisms remain understudied. We show interventions activate dormant self-correction capabilities in unfine-tuned models, improving performance on error-prone tasks.

Cognitive bias in LLM. LLMs exhibit human-like cognitive biases (Koo et al., 2024; Echterhoff et al., 2024; Jones & Steinhardt, 2022). The Self-Correction Blind Spot bears surface resemblance to the bias blind spot (the tendency to overlook one’s own biases) (Pronin et al., 2002), though we trace its cause to training data composition and argue in Section 5 that the blind spot is inherited from human demonstrations through the training pipeline.

Activation steering and representation engineering. Our mechanistic analysis builds on prior works, which compute steering vectors from contrastive prompt pairs and add them during inference to steer a particular behavior (Turner et al., 2024; Panickssery et al., 2024; Ardit et al., 2024; Zou et al., 2025; Chen et al., 2025). We apply this paradigm to self-correction. Our setting benefits from the controlled design of Self-Correction Bench, which ensures the steering direction reflects conversational role rather than confounding factors.

Our work integrates these threads into a systematic methodology for testing self-correction, and reveals LLMs’ inability to correct internal errors despite possessing the knowledge.

3 Methodology

3.1 Disentangling activation from knowledge

When a model fails to correct an error, two explanations are possible: the model lacks the knowledge to identify the error, or the model possesses the knowledge but fails to activate it. We isolate this distinction by presenting the same error in two matched conditions: attributed to the user (*external error*) or attributed to the model itself (*internal error*).

1. **Internal correction:** the error is injected into the model’s response via the assistant turn of the chat template, and the model is allowed to continue generating.
2. **External correction:** the identical error is placed in the user prompt, and the model responds in a new assistant turn.

Figure 1 illustrates both conditions. The error content, surrounding context, and question are identical; only the chat template role header differs. If a model corrects the external error but not the internal one, knowledge deficiency is ruled out, reflecting activation failure.

We quantify this gap as the *Self-Correction Blind Spot*:

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

where $P_M(r_{\text{correct}}|r, e)$, termed mean accuracy, is the fraction of samples in which model M arrives at the correct final answer despite error e under attribution r , r_m and r_u denote the error attributed to the model and user, respectively. A value of 1 indicates a complete blind spot. By conditioning on the same error e , we **isolate 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. On-policy errors conflate activation failure with knowledge limitations, making it impossible to isolate the mechanism we study.

This minimal contrast, differing only in conversational role, also enables mechanistic analysis: activation differences between conditions reflect the conversational-role variable rather than confounding factors, a property we exploit in Section 4.2.

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|> (model ends turn - no correction)
```

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 actu-
ally 2, not 3. Basic arithmetic operations like addition follow a set of
rules and patterns 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 concept.<|eot_id|>
```

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

3.2 Datasets

We evaluate self-correction across three datasets of increasing complexity and realism. See Appendix A for full details.

SCLI5. “Self-Correct Like I am 5” introduces simple answer errors (off-by-one, flip) to trivial tasks requiring no reasoning. If models cannot correct obvious errors, the failure reflects activation rather than knowledge. (286 samples)

GSM8K-SC. We inject reasoning errors into multi-step solutions in Cobbe et al. (2021) using GPT-4.1 (OpenAI, 2025) and validate with Gemini 2.5 Flash (Gemini Team, 2025) that errors propagate to incorrect answers. (1,313 samples)

PRM800K-SC. Derived from PRM800K (Lightman et al., 2024), we select samples where LLM-generated solutions contain genuine errors, capturing errors from real-world LLM use. (448 samples)

This progression from simple answer errors to realistic failures lets us map exactly where self-correction breaks down, making our methodology a useful diagnostic framework for improving LLM robustness.

3.3 Experimental Setup

Models. We evaluated a wide range of open-source LLMs, as most 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 and a fixed token budget of 1,024 to isolate self-correction from sampling variance and test-time compute effects. Sensitivity analysis in Appendix C confirms robustness to these choices.

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. To further validate robustness to judge choice, we compare Gemini 2.5 Flash against Claude Sonnet 4.6 and GPT-5.4 on 238 samples: three-way agreement is 95.0% (Appendix C.3). The prompt is

¹<https://deepinfra.com/>

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. We report 95% confidence intervals estimated via $\pm 1.96 \times$ standard error of the mean (SEM). For mean accuracy, $\sigma_M = \frac{\sigma}{\sqrt{N}}$, where N is the sample size and σ is the sample standard deviation; for Self-Correction Blind Spot, a ratio of two means, we propagate both SEMs through Eq. 1 by the delta method.

4 Result and analysis

4.1 Empirical findings

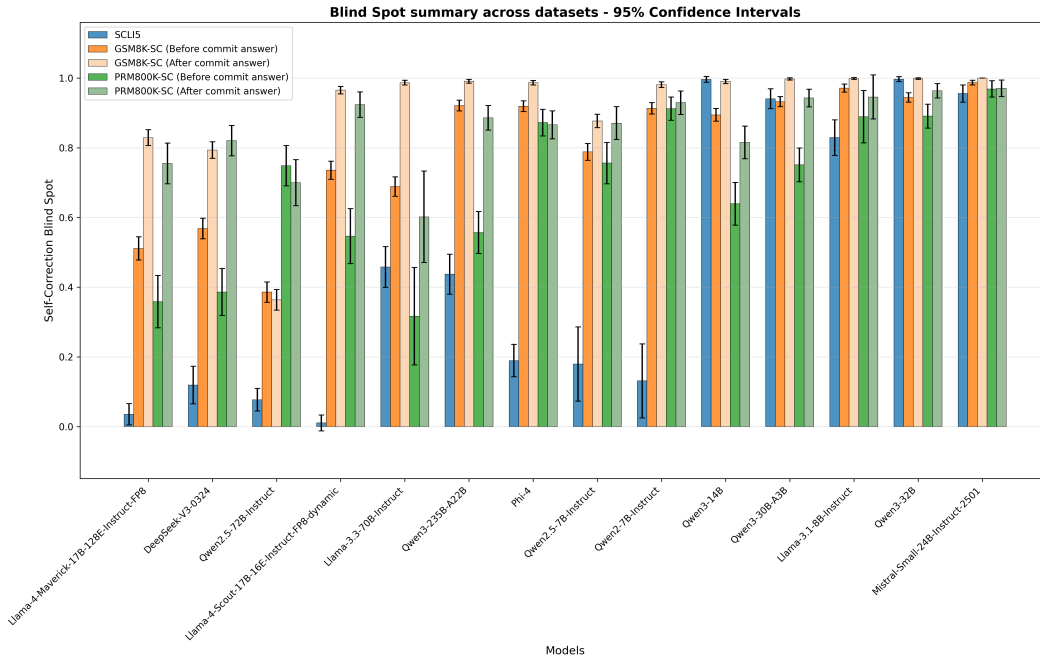


Figure 2: Self-Correction Blind Spot and 95% confidence interval across models

The blind spot is systematic. We identify a statistically significant Self-Correction Blind Spot across most models (Figure 2). The blind spot averages 64.5% and is consistent across model families and scales. We observe moderate positive correlations across datasets (Figure 5), indicating a fundamental rather than task-specific limitation: if models cannot self-correct trivial or complex tasks, the failure reflects an activation problem rather than a knowledge problem. The finding extends to closed-source frontier models² (Claude 3.5 Haiku: 52.5%, Sonnet 4: 41.4% blind spot in Table 7). We also demonstrate the 62.8% average blind spot in non-mathematical domains including logical deduction and object tracking from BIG-Bench Mistake (Tyen et al., 2024) (See Table 8).

On-policy validation. We also validate on on-policy errors from two independent sources: mathematical reasoning and factuality errors from Zheng et al. (2025) and Malik et al. (2026). Because on-policy errors are, by definition, errors the model committed and left uncorrected in its own generation, internal-condition accuracy is 0% by construction and Eq. 1 is uninformative; the quantity of interest is instead how many of its own errors a model catches once they are presented externally. That fraction is 4.3% - 10.8% (Table 9

²Most models do not support prefiling assistant responses, which is required in our setting.

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 (Hugging 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

Table 1: Descriptive statistics of correction markers in post-training datasets

and 10). These are errors the model had the knowledge to catch but did not catch in its own output. The uncorrected remainder is ambiguous: failure in the external condition may reflect knowledge deficiency rather than activation failure. This conflation is what the off-policy design removes.

Training data explains the asymmetry. Analysis of model responses reveals that external errors trigger substantially more correction markers³ than internal errors - 179.5% more in GSM8K-SC and 73.6% more in PRM800K-SC. This asymmetry is predictable from post-training data composition. Table 1 shows that the median correction marker frequency in non-reasoning supervised fine-tuning datasets (OpenAssistant, OpenHermes2.5, UltraFeedback, Tulu3) is zero, with the 95th percentile at just 1. In contrast, reasoning-model generated training data (Mixture-of-Thoughts, OpenThoughts3) has median marker densities of 30-170, with 99% of examples containing at least one marker. Models trained with little exposure to error-correction patterns rarely generate correction markers in response to their own errors, even though they produce them when evaluating external errors.

Correction markers as diagnostic probes. If the blind spot stems from the absence of correction markers in training, does forcing one activate latent capabilities? Inspired by the fact that correction markers are the most frequent first token for several reasoning models given an internal error (see Table 11), we append “Wait” after the model’s erroneous output. Even without fine-tuning, it reduces the blind spot by 89.3% on average (Figure 6) and increases mean accuracy by 156% (Figure 7). Other markers (“But”, “However”) also activate self-correction, though less effectively (Table 12). This confirms that the correction capability is latent rather than absent, and raises the question: what mechanism suppresses self-correction on internal errors, and what can unlock it?

4.2 Mechanistic Analysis

Having established the blind spot across 14 models, we select two distinct model families, Llama3.1 and Qwen2.5, to prioritize causal depth over additional breadth.

The conversational-role direction. We hypothesize that the model’s internal representation of conversational role gates access to correction capabilities, where the key variable is whether content is attributed to the user or to the model itself, as encoded by the chat template role header. To test this, we apply steering at the last error token during generation. This differs from prior activation steering approaches: ActAdd (Turner et al., 2024) applies the steering vector across the first N token positions of the prompt, while Ardit et al. (2024) ablate across all positions and all layers. Our single-position, single-layer approach is motivated by our minimal difference in context except the difference in role. We extract residual-stream activations at the last error token, and compute the mean difference:

³Correction markers include “Wait”, “But”, “However”, “No”, “Hold on”, “Alternatively”, “Hmm”.

$d = \frac{1}{N} \sum_i (h_{\text{internal}}^{(i)} - h_{\text{external}}^{(i)})$. We call this the *conversational-role direction*. Because the role header is the only cue that differs, role attribution and turn position cannot be varied independently within this design; we therefore scope our claims to conversational role rather than authorship.

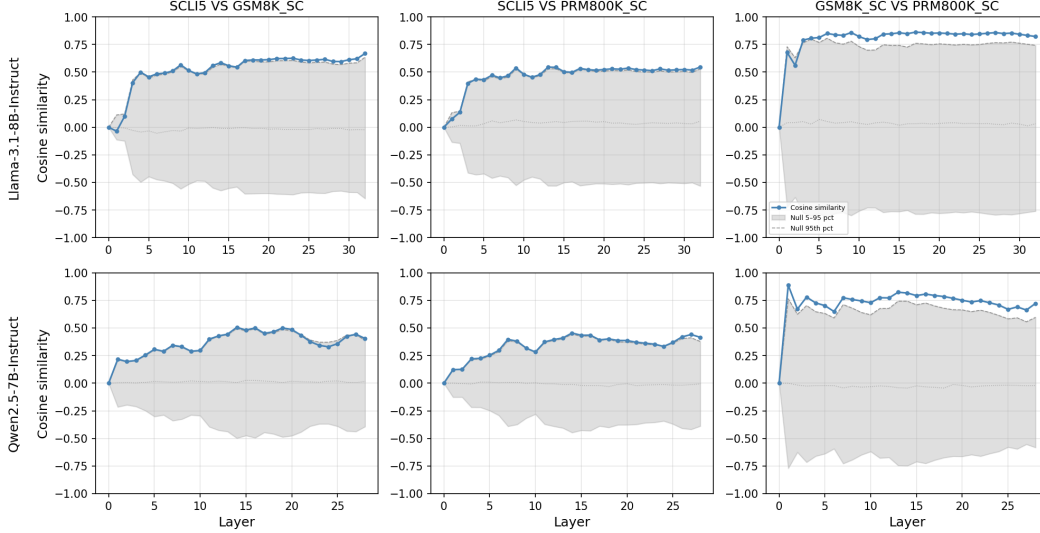


Figure 3: Cross-dataset cosine similarity of conversational-role direction with permutation baseline

This direction is content-independent. Figure 3 shows that the cross-dataset cosine similarity of the direction, computed on one dataset and measured on another, is significantly above a permutation null baseline ($n = 1000$) from mid-layers onward in both models.

Steering causally activates self-correction. To show that the conversational-role direction **causally** gates self-correction, we steer activations along it and measure if the model can self-correct internal error. After hyperparameter tuning, we apply steering at layer 13 by subtracting the conversational-role direction, scaled by $|\alpha|$, from the residual stream during internal-error processing (i.e. shifting the representation toward the external-role state).

Figure 4 shows results across both models and all three datasets. The direction is computed on GSM8K-SC and is cross-applied to SCLI5 and PRM800K-SC. As $|\alpha|$ increases from 1 to 3, steering along the direction increases self-correction monotonically, while the random control generally stays near the internal correction rate and the opposite direction generally holds correction at or below it. In SCLI5, Qwen2.5-7B’s 18% baseline blind spot leaves too little headroom to separate the three conditions. Cross-dataset steering transfers successfully, confirming the direction encodes a general conversational-role feature rather than task-specific information.

Unlike prompt-based interventions, activation steering modifies the model’s internal representation without altering the input, confirming that the conversational-role direction reflects a computational feature of the model rather than an artifact of the evaluation design. At $\alpha = -5.0$, the controls depart from their $\alpha \geq -3.0$ behavior on some pairs: Qwen2.5-7B’s opposite direction rises, so does its random control in SCLI5. Steering can also over-shoot: Llama-3.1-8B on SCLI5 falls from 0.486 to 0.273; at this magnitude Llama-3.1-8B also occasionally emits the assistant role header before correcting, as if its own prior output had become a user message to respond to. $\Delta(\text{real} - \text{random})$ remains significant at $\alpha = -5.0$ for the same five pairs, but $\alpha \geq -3.0$ gives the cleanest read (Table 13).

Relationship to “Wait”. A natural question is whether the conversational-role direction fully accounts for correction activation. To test this, we project the activation shift induced by appending “Wait” onto the conversational-role direction, computing an effective α that

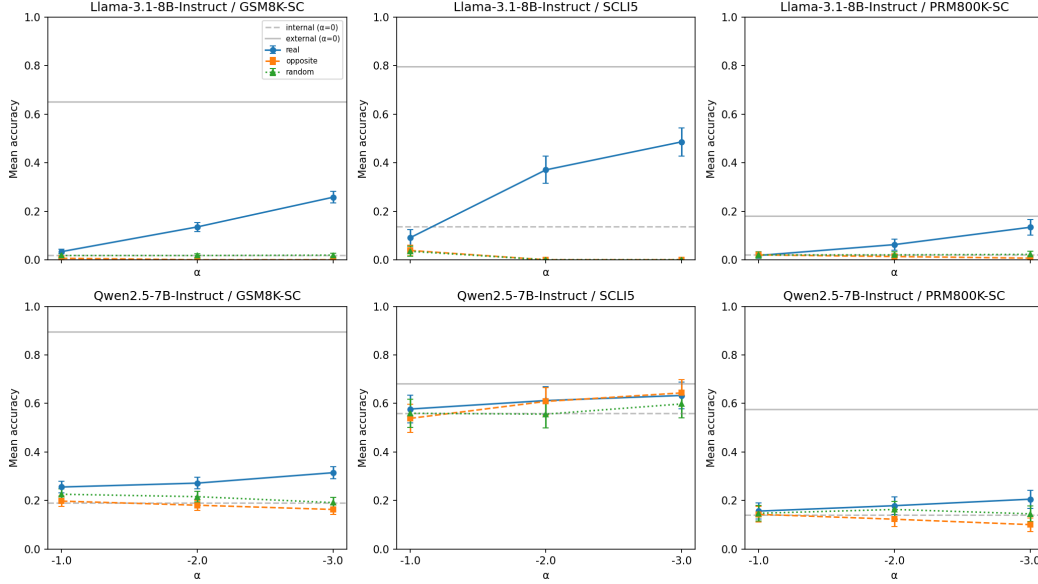


Figure 4: Mean accuracy (95% CI) versus steering coefficient α for the real conversational-role direction (blue), its negation (orange), and a norm-matched random direction (green). Grey lines mark the internal (dashed) and external (solid) baselines.

measures how far “Wait” moves the representation toward the external-role state. In Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct, the effective α s average -0.09 and -0.01 respectively across all datasets at layer 13. Yet both models show substantial correction improvement from “Wait”, confirming that correction activation involves at least two nearly independent mechanisms: conversational-role shift and a separate pathway likely related to pretraining associations between correction markers and re-evaluation sequences. Fully characterizing this second mechanism remains future work.

4.3 Causal Evidence via Fine-Tuning

The preceding analysis identifies two mechanisms that gate self-correction: the conversational-role direction and the pathway activated by correction markers. If both are consequences of training data composition, fine-tuning with error-correction traces should address both, as such data necessarily includes sequences where the model encounters its own errors and generates correction markers.

We use the MLX LM library⁴ to fine-tune Llama-3.1-8B-Instruct with LoRA (Hu et al., 2021) on a 10% sample from Mixture-of-Thoughts (Hugging Face, 2025) with a maximum length of 1,024, resulting in 5,306 samples. We fine-tune for 2 epochs and apply cosine decay with initial and terminal learning rates of 5e-5 and 5e-6. After fine-tuning, we observe a 76.0% reduction (Table 2) in the blind spot across datasets, providing causal evidence that the Self-Correction Blind Spot is a consequence of training data composition. Over 98% of the sampled traces are non-mathematical by source subset - the 1,024-token cap removes most mathematical traces, which are substantially longer - yet fine-tuning on them reduces the blind spot on our mathematical datasets. Correction behaviour learned from non-math traces transferring to math indicates that what the data supplies is a domain-general activation pattern rather than domain-specific knowledge. Fine-tuning on the same data with the <think> reasoning traces stripped out reduces the blind spot by 38.1%, roughly half the full-data effect, indicating that the error-correction content carries a large share of reduction; however, the remaining solutions still contain multiple-choice wrong-option elimination, which functionally resembles trial-and-correction.

⁴<https://github.com/ml-explore/mlx-lm/>

We further evaluate DeepSeek-R1-Distill-Llama-8B and DeepSeek-R1-Distill-Llama-70B, fine-tuned from Llama-3.1-8B and Llama-3.3-70B-Instruct respectively with 600K reasoning trajectories containing error-correction sequences (DeepSeek-AI et al., 2025a). Aligning with the result in our fine-tuning experiment, they show a 84.1% average reduction in the blind spot across datasets compared to their base models (Table 2).

Model	Dataset	Mean Accuracy		Blind Spot	Blind Spot (Base Model)
		External	Internal		
Llama-3.1-8B-Instruct fine-tuned with LoRA (ours)	SCLI5	0.678	0.619	0.088	0.829
	GSM8K-SC	0.556	0.363	0.347	0.971
	PRM800K-SC	0.295	0.232	0.212	0.889
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.

Table 2: Self-Correction Blind Spot in model fine-tuned with error and self-correction data

4.4 Reasoning Models

Reasoning models exhibit a small, even negative, Self-Correction Blind Spot (Figure 8), in stark contrast to non-reasoning models. This offers a partial explanation for the superior benchmark performance of reasoning models: by correcting errors during generation rather than committing to them, reasoning models recover from mistakes that would otherwise propagate to incorrect final answers. Our results suggest this advantage stems in part from training on error-correction sequences rather than a fundamental capability difference. Consistent with this, appending “Wait” to a non-reasoning base model nearly matches the performance of its corresponding reasoning model (Table 3), closing much of the gap without any additional training.

Base Model	Reasoning Model	Base Model	Appending “Wait”	Reasoning Model
DeepSeek-V3-0324	DeepSeek-R1-0528	0.567	0.902	0.908
phi-4	phi-4-reasoning-plus	0.325	0.701	0.707
Qwen3-14B ¹	Qwen3-14B ²	0.117	0.868	0.843
Qwen3-32B ¹	Qwen3-32B ²	0.045	0.793	0.894
Qwen3-30B-A3B ¹	Qwen3-30B-A3B ²	0.104	0.860	0.845
Qwen3-235B-A22B ¹	Qwen3-235B-A22B ²	0.328	0.856	0.876

¹ Non-thinking mode

² Thinking mode

Table 3: Macro average mean accuracy of base model vs appending “Wait” vs reasoning model

Although Qwen3 models undergo continued fine-tuning after GRPO (Shao et al., 2024) to unify thinking and non-thinking modes under a single chat template, non-thinking mode still exhibits the blind spot. The chat template conditions the model into distinct distributions, so the unified training does not eliminate the asymmetry.

5 Discussion

Decomposing self-correction. Our results suggest self-correction depends on at least three separable factors: *knowledge* (the ability to identify and fix the error, measured by

external error performance), *attribution* (whether the model activates correction given the conversational role, captured by the conversational-role direction), and *metacognitive triggering* (whether correction markers like “Wait” independently activate re-evaluation). Self-Correction Bench and our mechanistic analysis resolve the first two components; the effective alpha analysis provides initial evidence that the third is separable, motivating investigation in future work.

Benefit of error and self-correction data. LLMs are known to exhibit cognitive biases (Koo et al., 2024; Echterhoff et al., 2024; Jones & Steinhardt, 2022). Self-Correction Blind Spot bears resemblance to bias blind spot in humans. 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 (Teknium, 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-generated data lacks 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 Table 1. 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 fine-tuning (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 provide an equally important training signal as error-free demonstration.

Implications for AI Safety. The Self-Correction Blind Spot has direct safety implications. Our work identifies a concrete cause: a single direction in representation space that gates access to correction capabilities based on conversational role. Multi-agent architectures may partially address the blind spot by routing model outputs as external inputs to other agents, effectively converting internal errors into external ones.

6 Conclusion and limitations

We disentangle self-correction activation from knowledge deficiency by introducing a controlled evaluation where the same error is attributed to either the user or the model. Across 14 non-reasoning models, we find a 64.5% Self-Correction Blind Spot: models correct external errors but fail on identical internal ones. We trace this to a conversational-role direction in representation space that gates correction capabilities. This direction transfers across tasks, steering along it causally activates self-correction, and it is a training-induced artifact that is reducible by including error-correction sequences in fine-tuning. A nearly independent pathway activated by correction markers like “Wait” warrants further investigation. Our results decompose self-correction into knowledge, role attribution, and metacognitive triggering, informing both training recipes and inference-time interventions for improving LLM reliability.

Our headline 64.5% blind spot is measured on injected errors. This isolates activation from knowledge, but it also means the magnitude is a property of the controlled design rather than an estimate of how often models fail to correct themselves in deployment. On models’ own errors (Section 4.1), the design yields only a lower bound: at least 4.3–10.8% of the errors a model commits are ones it demonstrably had the knowledge to catch. Our mechanistic analysis deliberately targets two maximally distinct families (Llama and Qwen) at the 7-8B scale to establish existence and causality of the conversational-role direction. Confirming universality beyond these two families is a natural extension that does not affect the causal conclusions within them. We encourage future work to extend the mechanistic analysis to non-mathematical domains and to further characterize the metacognitive pathway through which correction markers operate independently of the conversational-role direction.

Ethics Statement

This work identifies a systematic failure mode in which LLMs fail to self-correct despite possessing the requisite knowledge and proposes methods to address it, including activation steering and fine-tuning with error-correction data. We believe these contributions are net positive for AI safety: models that reliably correct their own errors are more trustworthy in deployment. The safety implications discussed in Section 5 suggest that reducing the Self-Correction Blind Spot could mitigate error propagation in safety-critical applications.

The conversational-role direction we identify could in principle be used adversarially, for instance, by steering a model away from the external-role state to suppress self-correction. However, this requires white-box access to model weights. Furthermore, our steering experiments are designed to establish causality, not to optimize attack effectiveness. Many stronger attack approaches already exist (Turner et al., 2024; Panickssery et al., 2024; Ardit et al., 2024; Zou et al., 2025). Lastly, our error injection methodology involves only benign mathematical and logical errors, not harmful content.

Reproducibility statement

Our experiments utilize various open-source models, closed-source models, and datasets. The dataset is available under Hugging Face <https://huggingface.co/collections/kenhktsui/self-correction-bench>. Our codes for constructing datasets, running the experiment, and building tables and graphs are released in Github <https://github.com/kenhktsui/self-correction-bench>. We use LLMs to refine writing, including clarity, conciseness and word choice. We also use LLMs to write selected Python functions, to generate the GSM8K-SC dataset, and for automated evaluation.

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>.
- Anthropic. The claude 3 model family: Opus, sonnet, haiku, Mar 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.
- Andy Ardit, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction, 2024. URL <https://arxiv.org/abs/2406.11717>.
- Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi (eds.), *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 675–718, Nusa Dua, Bali, November 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.ijcnlp-main.45. URL <https://aclanthology.org/2023.ijcnlp-main.45/>.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models, 2025. URL <https://arxiv.org/abs/2507.21509>.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback, 2024. URL <https://openreview.net/forum?id=pNk0x3IVWI>.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanbiao Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yudian Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025a. URL <https://arxiv.org/abs/2501.12948>.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanbiao Zhao, Wei

An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025b. URL <https://arxiv.org/abs/2412.19437>.

Jessica Maria Echterhoff, Yao Liu, Abeer Alessa, Julian McAuley, and Zexue He. Cognitive bias in decision-making with LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 12640–12653, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.739. URL <https://aclanthology.org/2024.findings-emnlp.739/>.

Google Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities., June 2025. URL https://storage.googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid 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, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean

Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Rapparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenxin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenber, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippas Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabza, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshchev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, 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 Rodriguez, 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, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara

- Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL <https://arxiv.org/abs/2407.21783>.
- Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanxia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models, 2025. URL <https://arxiv.org/abs/2506.04178>.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL <https://arxiv.org/abs/2106.09685>.
- Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet, 2024. URL <https://arxiv.org/abs/2310.01798>.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, January 2025. ISSN 1558-2868. doi: 10.1145/3703155. URL <http://dx.doi.org/10.1145/3703155>.
- Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL <https://github.com/huggingface/open-r1>.
- Erik Jones and Jacob Steinhardt. Capturing failures of large language models via human cognitive biases. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL <https://openreview.net/forum?id=fc09Cgn-X-R>.
- Ryo Kamoi, Sarkar Snigdha Sarathi Das, Renze Lou, Jihyun Janice Ahn, Yilun Zhao, Xiaoxin Lu, Nan Zhang, Yusen Zhang, Haoran Ranran Zhang, Sujeeth Reddy Vummanthala, Salika Dave, Shaobo Qin, Arman Cohan, Wenpeng Yin, and Rui Zhang. Evaluating LLMs at detecting errors in LLM responses. In *First Conference on Language Modeling*, 2024a. URL <https://openreview.net/forum?id=dnwRScljXr>.
- Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024b. doi: 10.1162/tacl_a_00713. URL <https://aclanthology.org/2024.tacl-1.78/>.
- Geunwoo Kim, Pierre Baldi, and Stephen Marcus McAleer. Language models can solve computer tasks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=M60mjAZ4CX>.

- Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. Benchmarking cognitive biases in large language models as evaluators. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 517–545, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.29. URL <https://aclanthology.org/2024.findings-acl.29/>.
- Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Minh Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Alexandrovich Glushkov, Arnav Varma Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Julian Mattick. Openassistant conversations - democratizing large language model alignment. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL <https://openreview.net/forum?id=VSJotgbPHF>.
- Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=CjwERcAU7w>.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxu Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL <https://openreview.net/forum?id=i1uGbFHHpH>.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL <https://arxiv.org/abs/2307.13702>.
- Jijie Li, Li Du, Hanyu Zhao, Bo wen Zhang, Liangdong Wang, Boyan Gao, Guang Liu, and Yonghua Lin. Infinity instruct: Scaling instruction selection and synthesis to enhance language models, 2025. URL <https://arxiv.org/abs/2506.11116>.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=v8L0pN6EOi>.
- Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Formalizing and benchmarking prompt injection attacks and defenses. In *Proceedings of the 33rd USENIX Conference on Security Symposium*, SEC ’24, USA, 2024. USENIX Association. ISBN 978-1-939133-44-1.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=S37h0erQLB>.
- Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. Rewardbench 2: Advancing reward model evaluation.

- In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=fb0G86Dewb>.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1906–1919, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.173. URL <https://aclanthology.org/2020.acl-main.173/>.
- Meta. Llama 3.3, Dec 2024. URL https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/.
- Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, Apr 2025. URL <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>.
- Mistral AI Team. Mistral small 3, Jan 2025. URL <https://mistral.ai/news/mistral-small-3>.
- Sajad Mousavi, Ricardo Luna Gutiérrez, Desik Rengarajan, Vineet Gundecha, Ashwin Ramesh Babu, Avisek Naug, Antonio Guillen, and Soumyendu Sarkar. N-critics: Self-refinement of large language models with ensemble of critics, 2023. URL <https://arxiv.org/abs/2310.18679>.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL <https://arxiv.org/abs/2501.19393>.
- Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. Alice in wonderland: Simple tasks showing complete reasoning breakdown in state-of-the-art large language models, 2025. URL <https://arxiv.org/abs/2406.02061>.
- OpenAI. Introducing gpt-4.1 in the api, Apr 2025. URL <https://openai.com/index/gpt-4-1/>.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kopic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, 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 Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL <https://arxiv.org/abs/2303.08774>.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL <https://openreview.net/forum?id=TG8KACxE0N>.
- Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition, 2024. URL <https://arxiv.org/abs/2312.06681>.
- Emily Pronin, Daniel Y. Lin, and Lee Ross. The bias blind spot: Perceptions of bias in self versus others. *Personality and Social Psychology Bulletin*, 28(3):369–381, 2002. doi: 10.1177/0146167202286008. URL <https://doi.org/10.1177/0146167202286008>.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL <https://arxiv.org/abs/2412.15115>.
- Matthew Renze. The effect of sampling temperature on problem solving in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7346–7356, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.432. URL <https://aclanthology.org/2024.findings-emnlp.432/>.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.

- Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning, ICML'23*. JMLR.org, 2023.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=vAE1hFckW6>.
- Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=4FWAwZtd2n>.
- Teknium. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants, 2023. URL <https://huggingface.co/datasets/teknium/OpenHermes-2.5>.
- Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024. URL <https://arxiv.org/abs/2308.10248>.
- Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. LLMs cannot find reasoning errors, but can correct them given the error location. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13894–13908, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.826. URL <https://aclanthology.org/2024.findings-acl.826/>.
- Yubo Wang, Xiang Yue, and Wenhui Chen. Critique fine-tuning: Learning to critique is more effective than learning to imitate, 2025. URL <https://arxiv.org/abs/2501.17703>.
- Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: how does llm safety training fail? In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA, 2023. Curran Associates Inc.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL <https://aclanthology.org/2020.emnlp-demos.6/>.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL <https://arxiv.org/abs/2407.10671>.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, 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, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei

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

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

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

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2025. URL <https://arxiv.org/abs/2310.01405>.

Appendix

A Dataset construction

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

Table 4: Dataset comparison

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.

Table 5: Task composition of SCLI5

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.

Table 6: Error composition of GSM8K-SC

B Figures

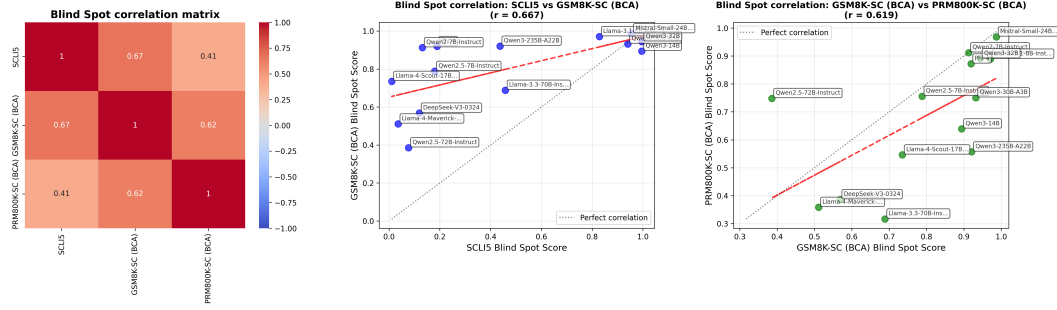


Figure 5: *left*: Blind spot correlation matrix *middle*: Scatter plot between SCLI5 vs GSM8K-SC *right*: Scatter plot between GSM8K-SC vs PRM800K-SC
BCA: Before commit an answer

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 7: Self-Correction Blind Spot in closed-source model

Model	Dataset	Mean Accuracy		Blind Spot	Size
		External	Internal		
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 8: Self-Correction Blind Spot in other domains in BIG-Bench Mistake

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

Table 9: Mean accuracy when on-policy errors in ProcessBench are presented externally

model	Factuality		Math	
	Accuracy (95% CI)	Size	Accuracy (95% CI)	Size
Qwen2.5-7B-Instruct	0.053 $\pm$ 0.051	76	Unavailable	0
Llama-3.1-8B-Instruct	0.108 $\pm$ 0.053	130	0.043 $\pm$ 0.034	138

Table 10: Mean accuracy when on-policy errors in RewardBench 2 are presented externally

Model	SCL15	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)

Table 11: Most common first word and relative frequency generated by reasoning models

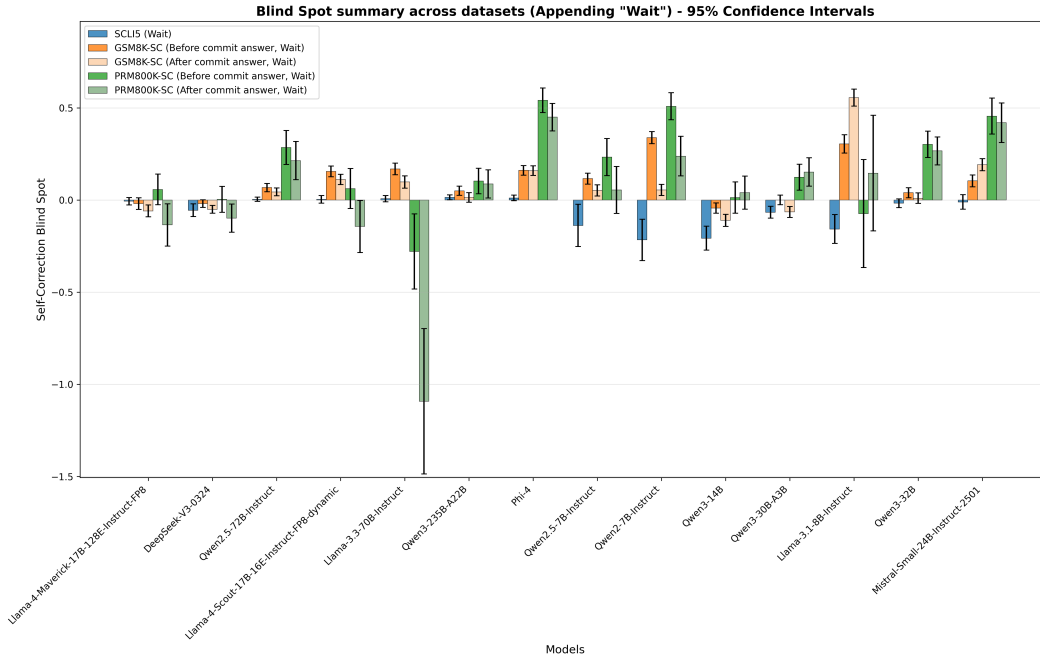


Figure 6: Self-Correction Blind Spot and 95% confidence interval across non-reasoning models after appending “Wait”

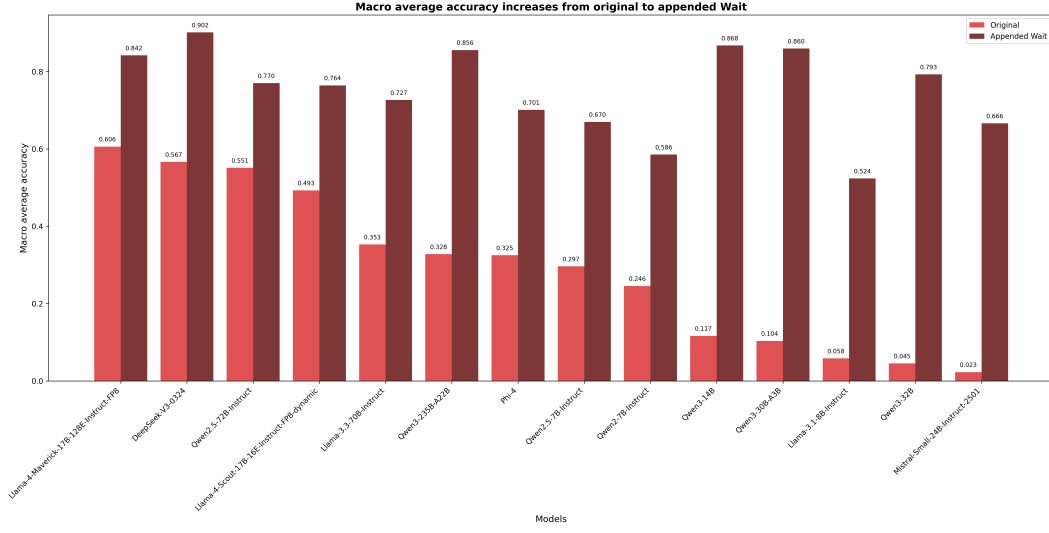


Figure 7: Macro average accuracy by non-reasoning model increases from original to appended “Wait”

Correction Markers	SCLi5	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%)

Table 12: Mean accuracy and relative change after appending various correction markers

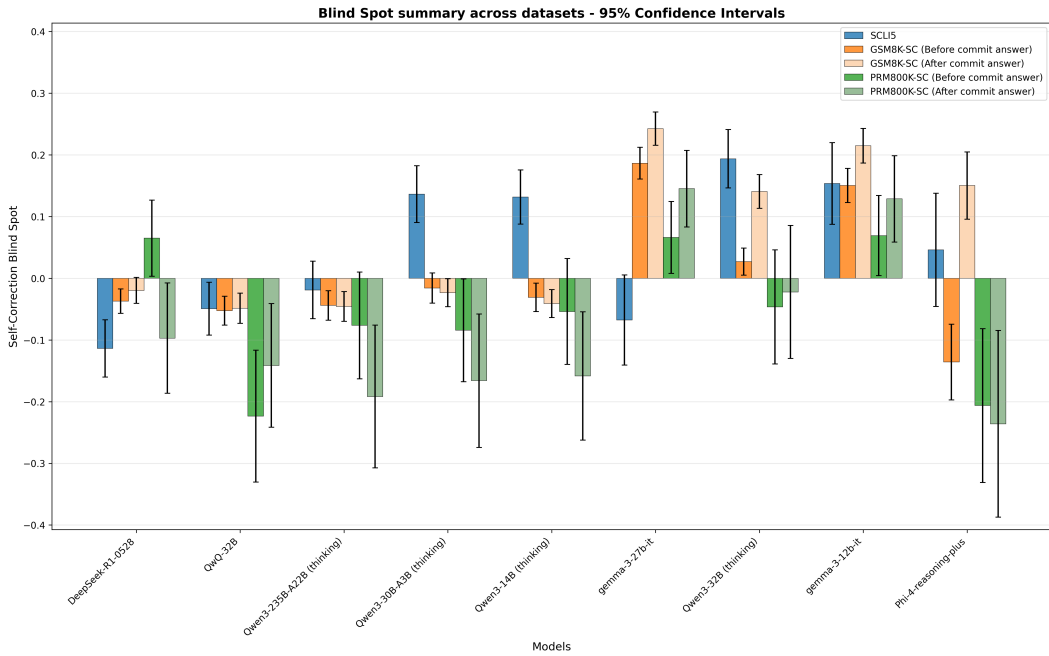


Figure 8: Self-Correction Blind Spot and 95% confidence interval across reasoning models

Condition	$\alpha = -1$	$\alpha = -2$	$\alpha = -3$	$\alpha = -5$
<i>Llama-3.1-8B-Instruct</i> / <i>GSM8K-SC</i> Internal ($\alpha=0$): 0.019, External: 0.652				
Real	0.034	0.135	0.258	0.314
Opposite	0.006	0.000	0.000	0.001
Random	0.018	0.018	0.019	0.028
$\Delta(\text{real}-\text{random})$	+0.016 $\pm$ 0.012	+0.118 $\pm$ 0.020	+0.239 $\pm$ 0.025	+0.286 $\pm$ 0.027
<i>Llama-3.1-8B-Instruct</i> / <i>SCLI5</i> Internal ($\alpha=0$): 0.136, External: 0.797				
Real	0.091	0.371	0.486	0.273
Opposite	0.038	0.000	0.000	0.000
Random	0.035	0.000	0.000	0.122
$\Delta(\text{real}-\text{random})$	+0.056 $\pm$ 0.040	+0.371 $\pm$ 0.056	+0.486 $\pm$ 0.058	+0.150 $\pm$ 0.064
<i>Llama-3.1-8B-Instruct</i> / <i>PRM800K-SC</i> Internal ($\alpha=0$): 0.020, External: 0.181				
Real	0.018	0.062	0.134	0.123
Opposite	0.020	0.013	0.007	0.000
Random	0.020	0.020	0.022	0.018
$\Delta(\text{real}-\text{random})$	-0.002 $\pm$ 0.018	+0.042 $\pm$ 0.026	+0.112 $\pm$ 0.034	+0.105 $\pm$ 0.033
<i>Qwen2.5-7B-Instruct</i> / <i>GSM8K-SC</i> Internal ($\alpha=0$): 0.190, External: 0.895				
Real	0.256	0.272	0.314	0.608
Opposite	0.197	0.181	0.163	0.276
Random	0.226	0.216	0.191	0.192
$\Delta(\text{real}-\text{random})$	+0.030 $\pm$ 0.033	+0.056 $\pm$ 0.033	+0.123 $\pm$ 0.033	+0.416 $\pm$ 0.034
<i>Qwen2.5-7B-Instruct</i> / <i>SCLI5</i> Internal ($\alpha=0$): 0.559, External: 0.682				
Real	0.577	0.612	0.633	0.664
Opposite	0.538	0.608	0.643	0.724
Random	0.559	0.556	0.598	0.643
$\Delta(\text{real}-\text{random})$	+0.017 $\pm$ 0.081	+0.056 $\pm$ 0.081	+0.035 $\pm$ 0.080	+0.021 $\pm$ 0.078
<i>Qwen2.5-7B-Instruct</i> / <i>PRM800K-SC</i> Internal ($\alpha=0$): 0.141, External: 0.576				
Real	0.156	0.179	0.205	0.266
Opposite	0.143	0.123	0.100	0.107
Random	0.147	0.163	0.145	0.143
$\Delta(\text{real}-\text{random})$	+0.009 $\pm$ 0.047	+0.016 $\pm$ 0.049	+0.060 $\pm$ 0.050	+0.123 $\pm$ 0.052

Table 13: **Activation steering along the conversational-role direction causally increases self-correction.** For each model and dataset, we report mean accuracy under steering at strength α along the direction (*Real*), its negation (*Opposite*), and a norm-matched random direction control (*Random*). Direction computed on GSM8K-SC, cross-applied to SCLI5 and PRM800K-SC. $\Delta(\text{real} - \text{random})$ separates the conversational-role effect from generic perturbation: it is significant at $\alpha = -3$ for all model-dataset pairs except Qwen2.5-7B-Instruct / SCLI5, where the small baseline blind spot (18%) leaves little headroom to steer. At $\alpha = -5$ the controls no longer sit at the internal baseline on some pairs (Section 4.2). Bold indicates Δ significantly above zero (95% CI excludes zero)

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 deterministic⁵ 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.

Model	SCLI5	GSM8K-SC	PRM800K-SC
Llama-4-Maverick-17B-128E-Instruct-FP8 (Meta, 2025)	0.948 ± 0.026	0.416 ± 0.027	0.455 ± 0.046
DeepSeek-V3-0324 (DeepSeek-AI et al., 2025b)	0.825 ± 0.044	0.399 ± 0.026	0.475 ± 0.046
Qwen2.5-72B-Instruct (Qwen et al., 2025)	0.92 ± 0.032	0.58 ± 0.027	0.154 ± 0.033
Llama-4-Scout-17B-16E-Instruct (Meta, 2025)	0.976 ± 0.018	0.24 ± 0.023	0.263 ± 0.041
Llama-3.3-70B-Instruct (Meta, 2024)	0.538 ± 0.058	0.275 ± 0.024	0.246 ± 0.04
Qwen3-235B-A22B ¹ (Yang et al., 2025)	0.563 ± 0.058	0.073 ± 0.014	0.348 ± 0.044
phi-4 (Abdin et al., 2024)	0.808 ± 0.046	0.076 ± 0.014	0.092 ± 0.027
Qwen2.5-7B-Instruct (Qwen et al., 2025)	0.559 ± 0.058	0.19 ± 0.021	0.141 ± 0.032
Qwen2-7B-Instruct (Yang et al., 2024)	0.601 ± 0.057	0.078 ± 0.014	0.058 ± 0.022
Qwen3-14B ¹ (Yang et al., 2025)	0.004 ± 0.007	0.092 ± 0.016	0.254 ± 0.04
Qwen3-30B-A3B ¹ (Yang et al., 2025)	0.056 ± 0.027	0.061 ± 0.013	0.194 ± 0.037
Llama-3.1-8B-Instruct (Grattafiori et al., 2024)	0.136 ± 0.04	0.019 ± 0.007	0.02 ± 0.013
Qwen3-32B ¹ (Yang et al., 2025)	0.004 ± 0.007	0.05 ± 0.012	0.083 ± 0.026
Mistral-Small-24B-Instruct-2501 (Mistral AI Team, 2025)	0.042 ± 0.023	0.011 ± 0.006	0.016 ± 0.012

¹ Qwen3 series models use non-thinking mode.

Table 14: Mean accuracy and 95% confidence interval of models at temperature 0.0

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

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
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

Table 15: Mean accuracy and 95% confidence interval of models at temperature 0.6

⁵Temperature of 0.0 will not generate fully deterministic results due to finite precision.

C.2 PRM800K-SC result in 4,096 token budget

To ensure a fair comparison between internal and external error correction, and across models, we maintain a fixed token budget of 1,024 across all conditions. This design choice partly isolates self-correction capabilities from the effect of test time compute, providing a more rigorous test of the blind spot phenomenon. We also report our results of PRM800K-SC with a fixed token budget of 4,096 below, which does not change our conclusion. We do not report the result of SCLI5 and GSM8K-SC as the ratio of model responses exceeding 1,024 tokens is immaterial.

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

Table 16: Mean accuracy of models in PRM800K-SC at different compute budget

C.3 Choice of LLM judge

We compare Gemini 2.5 Flash with Claude Sonnet 4.6 and GPT-5.4 on 238 samples. Three-way agreement is 95.0%, demonstrating robustness of LLM judge.

Model Comparison	Agreement	Cohen’s kappa
Gemini 2.5 Flash vs Claude Sonnet 4.6	0.979	0.953
Gemini 2.5 Flash vs GPT-5.4	0.954	0.899
Claude Sonnet 4.6 vs GPT-5.4	0.967	0.928

Table 17: Inter-judge agreement

D Prompt

D.1 Generating GSM8K-SC

```
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 9: Output schema, system prompt and prompt for generating GSM8K-SC dataset

```
from pydantic import BaseModel
```

```
class Calculation(BaseModel):  
    incorrect_answer: float
```

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

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

```
<description_of_mistake>  
{description_of_mistake}  
</description_of_mistake>
```

```
<incorrect_reasoning>  
{incorrect_reasoning}  
</incorrect_reasoning>
```

You are given a question, a description of the mistake,
and the resulting incorrect reasoning.
Your task is to follow the incorrect reasoning to arrive
at the incorrect answer.

Figure 10: Output schema, system prompt and prompt for validating GSM8K-SC dataset

D.2 Automatic Evaluation

```
from pydantic import BaseModel
```

```
class Evaluation(BaseModel):  
    is_correct_answer: bool  
    is_correct_reasoning: bool  
    is_backtracking: bool  
    is_admitting_mistake: bool
```

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

Figure 11: Output schema and system prompt for automatic evaluation of error in model

```
# FOR SCLI5
<question>
{question}
</question>

<golden_answer>
{golden_answer}
</golden_answer>

<given_wrong_answer>
{given_wrong_answer}
</given_wrong_answer>

<completion_from_model>
{completion_from_model}
</completion_from_model>
```

The model was provided a wrong answer <given_wrong_answer>, and is required to self correct and arrive at the golden answer.

You have to assess if <completion_from_model> :

- has backtracked
- has finally provided the answer that matches the <golden_answer>
- has provided correct reasoning
- has admitted the mistake.

```
# FOR GSM8K-SC AND PRM800K-SC
<question>
{question}
</question>

<golden_answer>
{golden_answer}
</golden_answer>

<given_wrong_reasoning>
{given_wrong_reasoning}
</given_wrong_reasoning>

<completion_from_model>
{completion_from_model}
</completion_from_model>
```

The model was provided a wrong step in the reasoning process <given_wrong_reasoning>, and is required to self correct and arrive at the golden answer.

You have to assess if <completion_from_model> :

- has backtracked
- has finally provided the answer that matches the <golden_answer>
- has provided correct reasoning
- has admitted the mistake.

Figure 12: Prompt for automatic evaluation of error in model

```
from pydantic import BaseModel
```

```
class EvaluationErrorInUser(BaseModel):  
    is_correct_answer: bool  
    is_correct_critique: bool
```

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

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

```
# FOR SCLI5
<question_and_user_answer>
{question_and_user_answer}
</question_and_user_answer>

<golden_answer>
{golden_answer}
</golden_answer>

<response_from_model>
{response_from_model}
</response_from_model>

The model was provided with <question_and_user_answer>
from user.
You have to assess if <response_from_model> :
- contains correct answer that matches the <golden_answer>
>
- contains correct critique of <question_and_user_answer>
```

```
# FOR GSM8K-SC AND PRM800K-SC
<question_and_user_reasoning>
{question_and_user_reasoning}
</question_and_user_reasoning>

<golden_answer>
{golden_answer}
</golden_answer>

<response_from_model>
{response_from_model}
</response_from_model>

The model was provided with <question_and_user_reasoning>
from user.
You have to assess if <response_from_model> :
- contains correct answer that matches the <golden_answer>
>
- contains correct critique of <
question_and_user_reasoning>
```

Figure 14: Prompt for automatic evaluation of error in user

E Example of Self-Correction Bench

E.1 SCLI5

```
# 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"
  }
]
```

E.2 GSM8K-SC

```
# 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

```

# error in user prompt
[
  {
    "content": "A Senate committee has 5 Democrats, 5
      Republicans, and 1 Independent. In how many ways can
      they sit around a circular table if all the members of
      each party all sit next to each other? (Two seatings are
      considered equivalent if one is a rotation of the other
      .) I notice that there are three groups of people:
      Democrats, Republicans, and Independent. I wonder how
      many ways I can arrange these groups around the circle,
      ignoring the order within each group for now. I recall
      that the number of ways to arrange n distinct objects
      around a circle is (n-1)!, since we can fix one object
      and then permute the rest. (Truncated due to size.) So,
      there are  $5!/2 = 60$  ways to order the Democrats around a
      circle. Similarly, there are 60 ways to order the
      Republicans around a circle. For the Independent, there
      is only one way to order him or her around a circle. So,
      to get the total number of ways to seat the committee,
      I need to multiply the number of ways to arrange the
      groups by the number of ways to order each group. That
      gives me  $2 \times 60 \times 60 \times 1 = 7200$  ways.",
    "role": "user"
  }
]
# error in model response
[
  {
    "content": "A Senate committee has 5 Democrats, 5
      Republicans, and 1 Independent. In how many ways can
      they sit around a circular table if all the members of
      each party all sit next to each other? (Two seatings are
      considered equivalent if one is a rotation of the other
      .)",
    "role": "user"
  },
  {
    "content": "I notice that there are three groups of people:
      Democrats, Republicans, and Independent. I wonder how
      many ways I can arrange these groups around the circle,
      ignoring the order within each group for now. I recall
      that the number of ways to arrange n distinct objects
      around a circle is (n-1)!, since we can fix one object
      and then permute the rest. (Truncated due to size.) So,
      there are  $5!/2 = 60$  ways to order the Democrats around a
      circle. Similarly, there are 60 ways to order the
      Republicans around a circle. For the Independent, there
      is only one way to order him or her around a circle. So,
      to get the total number of ways to seat the committee,
      I need to multiply the number of ways to arrange the
      groups by the number of ways to order each group. That
      gives me  $2 \times 60 \times 60 \times 1 = 7200$  ways.",
    "role": "assistant"
  }
]

```