When Small Models Are Right for Wrong Reasons: Process Verification for Trustworthy Agents

Laksh Advani

Independent Researcher
Seattle WA
laad8452@colorado.edu

Abstract

Deploying small language models (7-9B parameters) as autonomous agents requires trust in their reasoning, not just their outputs. We reveal a critical reliability crisis: 50-69% of correct answers from these models contain fundamentally flawed reasoning—a “Right-for-Wrong-Reasons” phenomenon invisible to standard accuracy metrics. Through analysis of 10,734 reasoning traces across three models and diverse tasks, we introduce the Reasoning Integrity Score (RIS), a process-based metric validated with substantial inter-rater agreement ($\kappa = 0.657$). Conventional practices are challenged by our findings: while retrieval-augmented generation (RAG) significantly improves reasoning integrity (Cohen’s $d = 0.23$–0.93), meta-cognitive interventions like self-critique often harm performance ($d = -0.14$ to -0.33) in small models on the evaluated tasks. Mechanistic analysis reveals RAG succeeds by grounding calculations in external evidence, reducing errors by 7.6%, while meta-cognition amplifies confusion without sufficient model capacity. To enable deployment, verification capabilities are distilled into a neural classifier achieving 0.86 F1-score with 100× speedup. These results underscore the necessity of process-based verification for trustworthy agents: accuracy alone is dangerously insufficient when models can be right for entirely wrong reasons.

Introduction

Autonomous agents powered by small language models (7-9B parameters) promise democratized AI deployment: running on consumer hardware, responding in milliseconds, and operating at marginal cost. Yet a fundamental question threatens this vision: *Can we trust their reasoning?* We present evidence of a critical reliability crisis: even when producing correct outputs, these models exhibit fundamentally flawed reasoning 50–69% of the time, a phenomenon we term “Right-for-Wrong-Reasons” (RWR).

Consider an agent tasked with financial calculations. Given “What is 15% of 80?”, it responds: *“Step 1: To find 15% of 80, I multiply 80 by 0.2. Step 2: 80 × 0.2 = 12. Answer: 12.”* While correct, the reasoning is mathematically wrong—using 0.2 instead of 0.15. In autonomous operation, such hidden failures compound catastrophically: an agent might approve transactions, make medical recommendations, or control systems based on coincidentally correct but fundamentally flawed logic.

This problem is particularly acute for the small models that will power most deployed agents. Unlike frontier models confined to centralized servers, 7-9B parameter models enable edge deployment, privacy preservation, and cost-effective scaling. However, their limited capacity makes them vulnerable to reasoning failures that accuracy metrics cannot detect. Current evaluation paradigms judge only final outputs, creating a dangerous blind spot: agents that appear reliable in benchmarks but fail unpredictably in deployment.

This paper presents a large-scale study of reasoning integrity in small language models, extending prior diagnostics to agentic contexts, analyzing 10,734 reasoning traces across three models (Llama-3-8B, Mistral-7B, Qwen-2.5-7B) and three diverse tasks (mathematical reasoning, multi-hop QA, and commonsense reasoning). We address three critical questions: **(RQ1)** How severe is the hidden reasoning failure problem in small models deployed as agents?; **(RQ2)** Which interventions improve reasoning integrity—do popular approaches like self-critique and retrieval-augmentation help?; and **(RQ3)** What mechanisms explain why interventions succeed or fail, and how can we detect failures efficiently?

Our analysis suggests that understanding *why* interventions succeed or fail is more important than *what* they do. We show that while **retrieval-augmentation (RAG)** substantially improves reasoning integrity ($d = 0.23$–0.93), primarily by reducing calculation errors through external evidence, **meta-cognitive prompts** consistently harm performance ($d = -0.14$ to -0.33) by amplifying internal confusion. Our contributions include the **Reasoning Integrity Score (RIS)** ($\kappa = 0.657$), large-scale evidence of 50–69% hidden reasoning flaws across 10,734 traces, and a **deployable verifier** (0.86 F1, 100× speedup) for real-time trust assessment.

These results have immediate implications for deploying trustworthy agents. We provide actionable guidance: prioritize RAG for small models on fact-grounded tasks where retrieval is feasible, avoid meta-cognitive prompting in sub-10B models for knowledge-intensive tasks, and implement process-based verification as a non-negotiable safety layer.

Related Work

Process vs. Outcome Evaluation. Recent work recognizes that outcome-based evaluation masks reasoning failures (Lightman et al. 2023; Yuan et al. 2025). However, these focus on *training* improvements rather than *detecting* hidden failures in deployed models, and none quantify the prevalence of right-for-wrong-reasons behavior we reveal.

Small Model Reliability. The challenges of sub-10B parameter models are well-documented (Magesh et al. 2024; Kalai et al. 2025; Cheng et al. 2024), including hallucinations and factual inaccuracies. Yet prior work lacks systematic measurement of *hidden* failures—when models produce correct outputs through flawed reasoning—which we show affects 50-69% of "successful" cases.

Intervention Strategies. While RAG (Wang et al. 2025; Chen, Myrzakhan, and Lee 2025) and meta-cognitive techniques like self-critique (Gou et al. 2023; Yang et al. 2024b) are common, limited prior work has systematically compared their effects on reasoning integrity rather than accuracy. We provide evidence that meta-cognition can actively *harm* reasoning quality ($d = -0.14$ to -0.33) while RAG consistently helps ($d = 0.23$–0.93).

Gap. Despite growing deployment of small models as agents, few large-scale studies have quantified hidden reasoning failures, systematically compared intervention effects on process quality, or explained the mechanisms. Our 10,734-trace analysis fills this critical gap for trustworthy agent deployment.

Methodology

To systematically investigate the Right-for-Wrong-Reasons (RWR) phenomenon in small language models, we conducted a large-scale empirical study. Our methodology encompasses dataset selection, model choices, trace generation, intervention implementation, reasoning evaluation via RIS, error classification, and statistical validation. All trace generation experiments were conducted via the OpenRouter API, while trace analysis and distilled model training were performed locally.

Datasets and Models

We selected three diverse benchmarks: **GSM8K** (Cobbe et al. 2021) (1,319 mathematical word problems), **HotpotQA** (Yang et al. 2018) (1,000 multi-hop QA samples), and **ARC** (Clark et al. 2018) (1,119 commonsense science questions). These were subsampled to balance computational feasibility while maintaining diversity.

We evaluated three popular open-source small models: **Llama-3-8B** (Dubey et al. 2024), **Mistral-7B** (Jiang et al. 2023), and **Qwen-2.5-7B** (Yang et al. 2024a). Models were used in their base instruction-tuned variants with greedy decoding (temperature=0).

Reasoning Trace Generation

For each model-dataset pair, we generated reasoning traces under four conditions (baseline + three interventions), yielding 10,734 traces total (3 models × 3 datasets, with approximately 298 samples per condition per dataset). Traces were prompted to produce step-by-step reasoning in a structured format: "Step 1: [reasoning] Step 2: [reasoning] ... Final Answer: [output]", adapted from standard Chain-of-Thought templates (Wei et al. 2022).

Interventions

We implemented three lightweight interventions: (1) **Retrieval-Augmented Generation (RAG)**, which provided oracle ground-truth context (e.g., Wikipedia snippets for HotpotQA) with the prompt: "Use the provided context to reason step by step."; (2) **Self-Critique**, which prompted the model to review its reasoning: "Critique your previous reasoning for errors and provide a corrected version if needed."; and (3) **Verification Prompts**, which added to the initial prompt: "Verify each step for accuracy before proceeding to the next."

Reasoning Integrity Score (RIS)

RIS measures process quality by scoring each reasoning step on a 0.0-1.0 scale: 1.0 (fully correct), 0.5 (partial flaw), 0.0 (wrong). Steps were extracted via regex parsing. For each trace, RIS = average step score. Scoring used three independent LLM judges (GPT-4o-mini, Claude-3.5-Sonnet, Gemini-1.5-Flash) with a detailed rubric. Inter-rater reliability was validated on 500 steps (Fleiss' $\kappa = 0.657$, substantial agreement). Final RIS used majority vote. A trace was classified as "flawed" if $RIS < 0.8$, a threshold determined via sensitivity analysis. We tested thresholds from 0.7 to 0.9 and selected 0.8 as it optimized the balance between sensitivity (detecting flawed reasoning) and precision (avoiding false alarms).

Error Analysis

To uncover mechanisms, we manually categorized 1,000 flawed steps into four types: **Calculation Error** (wrong arithmetic, numbers, or fact application), **Hallucination** (fabricated information), **Logical Leap** (invalid inference), or **Other**. Distributions were computed per condition relative to baseline. We also measured error position (normalized 0–1), context misuse (fraction of retrieved facts incorrectly applied), and correlations using Pearson's r.

Distilled Verification System

To enable efficient deployment, we trained a lightweight MLP classifier to predict flawed traces ($RIS < 0.8$) using hybrid features: Sentence-BERT embeddings (384D from all-MiniLM-L6-v2) + 7 structural metrics (e.g., step count, trace length). The model (391 input features, 5 layers, ∼300k params) was trained on 80% of traces (stratified split) using Focal Loss ($\gamma = 2.0$, $\alpha = 0.25$), AdamW ($lr = 5 \times 10^{-4}$), and early stopping. Evaluation on 20% held-out test data yielded 0.86 macro F1, with 0.88 precision on "flawed" class, achieving ∼100× speedup over LLM judging (5–10ms inference on CPU).

Statistical Analysis

Statistical power was computed post-hoc, with substantial variation across intervention types: RAG effects showed excellent power (0.95-1.00), self-critique effects showed good power (0.76-0.99), and verification effects showed mixed power (0.56-0.96). We report only findings that met the conventional 0.75 threshold for adequate power, though some verification effects approached this threshold.

Results

Our analysis of 10,734 reasoning traces reveals pervasive hidden failures in small language models and clear patterns in intervention effectiveness. We present our key empirical findings, validated by statistical analysis.

Prevalence of Hidden Reasoning Failures

As shown in Table 1, the Right-for-Wrong-Reasons problem is severe: 50–69% of correct final answers exhibit flawed reasoning (RIS < 0.8). The issue varies by model and task, with Qwen-2.5-7B showing the highest average rate (69.3%) despite its relative strength, potentially due to more verbose reasoning chains that increase error opportunities. HotpotQA demonstrates the most acute failures (67.9% average), suggesting that knowledge-intensive tasks exacerbate reliance on spurious patterns rather than robust logic.

Table 1: Percentage of correct outputs with flawed reasoning (RIS < 0.8) across models and tasks.

Model	ARC	GSM8K	HotpotQA	Avg
Mistral-7B	45.8%	44.3%	60.5%	50.2%
Llama-3-8B	47.0%	59.2%	59.4%	55.2%
Qwen-2.5-7B	61.4%	62.7%	83.8%	**69.3%**
Average	51.4%	55.4%	67.9%	58.2%

Intervention Effects

Figure 1 summarizes the impact of interventions on RIS. Retrieval-augmented generation (RAG) consistently improves reasoning integrity, with medium-to-large effect sizes on fact-grounded tasks (mean $d = 0.41$, up to 0.93 on HotpotQA for Qwen). In contrast, self-critique and verification prompts harm performance in 78% of conditions (mean $d = -0.14$ and -0.15, respectively), with effects most negative for weaker models like Mistral and Llama.

Task dependency is evident: RAG shows negligible effects on pure reasoning (ARC, $d \approx 0$) but strong benefits on math and QA (GSM8K $d = 0.23$–0.43; HotpotQA $d = 0.51$–0.93). Meta-cognitive harms are consistent, suggesting a capacity threshold where self-reflection fails.

Error Mechanisms

Table 2 illustrates error type shifts. Calculation errors dominate baselines (60.3%), and RAG reduces them most effectively (-7.6%), while increasing hallucinations ($+4.5\%$) and logical leaps ($+3.3\%$). This trade-off results in positive RIS, as “reasoning attempts” (scored 0.5) represent partial credit that is substantially higher than outright failures (scored 0.0). The reduction in fundamental calculation errors, which are completely incorrect, outweighs the increase in reasoning attempts that contain partial flaws. Metacognitive interventions yield smaller reductions (-4.2%) with comparable increases, resulting in net harm.

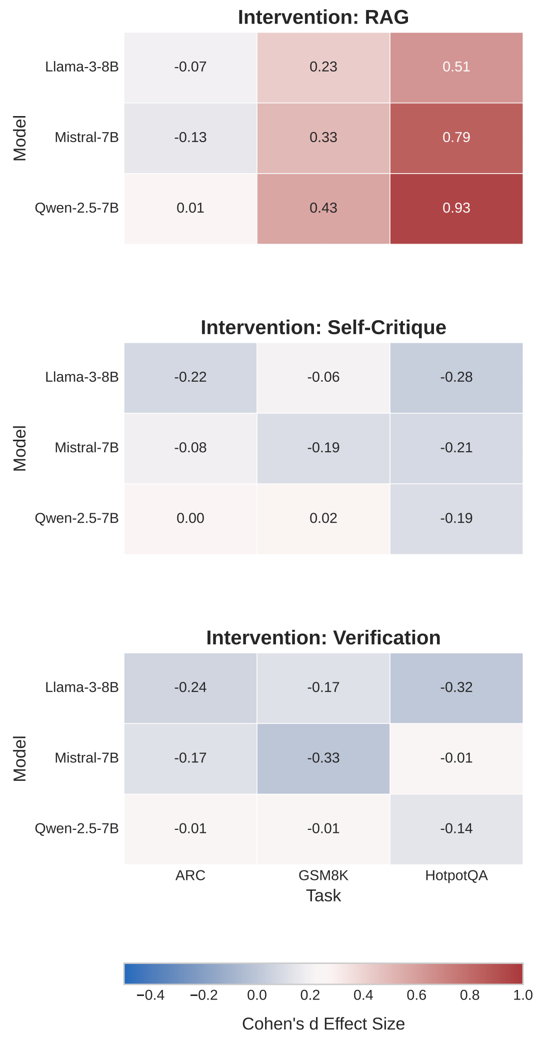

Figure 1: Cohen’s d effect sizes for interventions. Red indicates improved reasoning integrity (positive d), while Blue indicates harm (negative d).

Supporting analyses reveal that context misuse strongly predicts RAG failure ($r = -0.951$), weaker baseline models benefit more from RAG ($r = 0.671$), and errors accumulate late in the reasoning process (mean position=0.56–0.71). All correlations were statistically significant ($p < 0.001$).

Distilled Verifier Performance

The MLP classifier achieves 0.86 macro F1 on held-out data, with 0.88 precision and 0.87 recall on the “flawed” class. This performance, combined with low latency (∼5–10ms), makes it suitable for real-time production alerting in autonomous agents.

Table 2: Error distribution changes relative to baseline (percentage points).

Error Type	Baseline	+RAG	+Self-Crit	+Verify
Calculation Error	60.3%	52.7%	56.1%	56.1%
Change	–	↓7.6	↓4.2	↓4.2
Hallucination	25.2%	29.7%	27.2%	27.9%
Change	–	↑4.5	↑2.0	↑2.7
Logical Leap	14.3%	17.6%	16.7%	16.0%
Change	–	↑3.3	↑2.4	↑1.7

Discussion

Our empirical results demonstrate a critical issue in small language models, but also illuminate a clear path forward. The findings challenge core assumptions about agent reliability, suggesting that *what* interventions do is less important than *why* they work.

Why Meta-Cognition Fails: Pseudo-Reflection

The most striking finding is the consistent, harmful effect of meta-cognitive prompts. Our analysis suggests this is not merely a neutral failure but an active introduction of new errors. We posit this is due to ”pseudo-reflection”: small models lack the genuine, high-level meta-cognitive capacity to introspect. When prompted to ”critique” or ”verify,” they do not *perform* reflection; they *generate text that looks like reflection.*

This pseudo-reflection amplifies errors by inventing incorrect justifications. The model, lacking an internal ”ground truth” to check against, invents plausible-sounding (but incorrect) justifications, as seen in the trade-off analysis (Table 2). While meta-cognitive interventions occasionally reduce calculation errors (-4.2%), they simultaneously increase hallucinations and logical leaps, resulting in a net-negative impact on reasoning integrity (mean $d \approx -0.15$). This suggests that small models can identify and correct some errors but lack the capacity to avoid introducing new ones during the critique process, supporting the existence of a ”capacity threshold” for effective self-reflection that 7-9B models fall below.

Why RAG Succeeds: External Scaffolding

RAG's success (mean $d = 0.41$) may be understood as providing ”external scaffolding.” The strong correlation ($r = 0.671$) between weak baseline performance and high RAG benefit suggests that RAG may function as a cognitive orthotic, potentially compensating for the model's weak internal knowledge and reasoning.

This analysis is supported by the error position data. Errors accumulate late in reasoning traces (mean position=0.56–0.71), where the model's internal state ”drifts” from the original facts. RAG provides a constant external anchor, re-grounding the model at each step and preventing this drift. The near-perfect negative correlation ($r = -0.951$) between context misuse and RAG effectiveness confirms this: RAG's benefit is almost entirely dependent on the model's ability to correctly integrate this external scaffolding. It is important to note that our study used oracle retrieval, which provides an upper bound on RAG's effectiveness. Real-world RAG systems with noisy retrievers may show diminished benefits.

Implications for Agentic Trust

The 50-69% RWR rate (Table 1) demonstrates that output-based accuracy is a dangerously insufficient proxy for reliability. This mandates a shift to ”continuously audit,” for which our distilled verifier (0.86 F1, 5-10ms inference) provides a practical ”trust alarm,” flagging high-risk, flawed reasoning chains for human review in real-time, something impossible with slow LLM-as-a-judge evaluations.

Limitations

We acknowledge several limitations: our study used **oracle RAG**, representing a best-case upper bound on RAG's benefit; our conclusions about meta-cognition failing on 7-9B **models** may not apply to larger ones (e.g., 70B+); our **RIS metric** averages step scores and can miss holistic failures; all experiments were in **English**, using **LLM judges** that may have biases; and our findings are based on three specific models and three task domains, and may not generalize to all small models or tasks.

Future Work

Future work will validate these findings with noisy, ”real-world” RAG retrievers, identify the ”capacity threshold” where meta-cognitive interventions may become effective (e.g., in 40B-70B+ models), and enhance the distilled verifier, potentially using graph-based networks to model the reasoning trace as a dependency graph.

Conclusion

This work quantifies a critical, hidden failure mode in small language model agents: 50-69% of their correct answers are ”Right-for-Wrong-Reasons,” produced by fundamentally flawed reasoning. We show this trust gap is invisible to accuracy metrics. Our 10,734-trace analysis provides a clear, actionable path to mitigating this risk: retrieval-augmented generation (RAG) acts as essential cognitive scaffolding, robustly improving reasoning integrity (d=0.23-0.93). Conversely, we provide strong evidence that common ”best practices” like self-critique are actively harmful (d=-0.14 to -0.33) **when applied to small models in the evaluated domains**, causing ”pseudo-reflection” that amplifies errors.

We contribute both the Reasoning Integrity Score (RIS) as a validated, process-based metric, and a fast, high-precision distilled classifier (0.86 F1) to deploy this verification at scale. For the field to move toward trustworthy autonomous agents, we should consider shifting our evaluation paradigm. Our findings suggest that accuracy alone is insufficient; process-based verification may need to become an essential safety layer **for small language models deployed on tasks requiring factual knowledge or multi-step reasoning**.

References

Chen, S. S.; Myrzakhan, A.; and Lee, K. 2025. DRAG: Distilling RAG for SLMs from LLMs to Transfer Knowledge across Domains. *arXiv preprint arXiv:2506.01954*.

Cheng, X.; Li, S.; Wen, K.; Wang, H.; Zhou, Y.; Tang, L.; Li, C.; and Wang, J. 2024. Small Agent Can Also Rock! Empowering Small Language Models as Hallucination Detector. *arXiv preprint arXiv:2406.11277*.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv preprint arXiv:1803.05457*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, H.; Bedoui, A.; Zhang, A.; et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Gou, Z.; Shao, Z.; Gong, Y.; Yang, Y.; Huang, M.; Duan, N.; and Chen, W. 2023. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. *arXiv preprint arXiv:2305.11738*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Kalai, A. T.; Nachum, O.; Vempala, S. S.; and Zhang, E. 2025. Why Language Models Hallucinate. *arXiv preprint arXiv:2509.04664*.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Branwen, G.; Biles, J.; Sutskever, I.; Schulman, J.; and Liang, P. 2023. Let's Verify Step by Step. *arXiv preprint arXiv:2305.20050*.

Magesh, V.; Surani, F.; Dahl, M.; Suzgun, M.; Manning, C. D.; and Ho, D. E. 2024. Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools. *arXiv preprint arXiv:2405.20362*.

Wang, L.; Chu, Z.; Chen, X.; Li, Y.-F.; Tran, Q. H.; Knopman, D.; Zhang, W.; He, Q.; Qiu, X.; Guo, Q.; and Wang, Y. 2025. MiniRAG: Towards Extremely Simple Retrieval-Augmented Generation. *arXiv preprint arXiv:2501.06713*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; et al. 2024a. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2369–2380.

Yang, Z.; Zhang, Y.; Wang, Y.; Xu, Z.; Lin, J.; Sui, Z.; and Liu, T. 2024b. Confidence v.s. Critique: A Decomposition of Self-Correction Capability for LLMs. *arXiv preprint arXiv:2412.19513*.

Yuan, X.; Zhang, X.; Xu, K.; Xu, Y.; Yu, L.; Wang, J.; Dong, Y.; and Wang, H. 2025. Tracing LLM Reasoning Processes with Strategic Games: A Framework for Planning, Revision, and Resource-Constrained Decision Making. arXiv:2506.12012.

Supplementary Material

A. Generation Prompts Interventions

To ensure reproducibility, we provide the exact system and user prompts used for all generation tasks. The base system prompt was modified dynamically based on the intervention type.

- **Base System Prompt:**

  > ”Solve the user's request step by step. For math problems, put the final answer in brackets [like this]. For multiple-choice questions, put the final answer (e.g., [A] or [1]) in brackets.”

- **Intervention: Prompt-Based Verification**
  Prepend to System Prompt:

  > ”Verify each step before proceeding. [Base System Prompt]”

- **Intervention: Retrieval-Augmented Generation (RAG)**
  Modification to User Prompt:

  > ”Context: {retrieved_context} {user_question}”

  Note: Context was sourced from ground_truth_decomposition for GSM8K/ARC and supporting sentences for HotpotQA.

- **Intervention: Self-Critique**
  Append to System Prompt:

  > ”[Base System Prompt] After solving, review your reasoning for any flaws.”

B. Judge Verifier Prompts

We utilized a strong judge model (DeepSeek-V3/Gemini-Flash) to evaluate reasoning integrity and classify error types.

1. Reasoning Integrity Score (RIS) Binary Judge Used to determine if a specific step S_t is valid given Context C.

> ”You are a strict verifier. Your task is to determine if the 'Generated Step' is logically and factually supported by the 'Context'. Context: {context} Generated Step: {step} Is the 'Generated Step' fully and correctly supported by the 'Context'? Respond with only 'Yes' or 'No'.”

2. Failure Mode Classification Used to categorize why a specific step failed.

> ”You are an error analyst. The ’Generated Step’ was deemed flawed (incorrect). Given the ’Context’, classify the primary error in the ’Generated Step’. Categories: [1. Factual Error, 2. Logical Leap, 3. Numerical Miscalculation, 4. Other] Context: {context} Generated Step: {step} Output only the category name.”

3. RAG Misuse Classification Used to detect if the model ignored or Hallucinated based on retrieved context.

> ”You are an error analyst. Determine if the ’Generated Step’ misuses the ’Context’. Misapplication: references context but uses it incorrectly (e.g., logical error, misquote, misinterpretation). Correct: uses the context correctly. Irrelevant: does not use the context at all. Respond with only ’Misapplication’, ’Correct’, or ’Irrelevant’.”

C. Implementation Details

- **Generator Models:**
  - `mistralai/Mistral-7B-Instruct-v0.2`
  - `meta-llama/Llama-3-8B-Instruct`
  - `qwen/qwen-2.5-7b-instruct`
- **Datasets:** GSM8K (Math), HotpotQA (Multi-hop QA), ARC-Challenge (Reasoning).
- **Sampling:** Temperature was set to default (varied by provider, typically 0.7-1.0) with standard top-p sampling.
- **Judge Model:** `deepseek/deepseek-v3.1-terminus` and `google/gemini-2.5-flash-lite` were used for automated evaluation.
- **Distilled Verifier:** A 4-layer MLP (512-256-128-1) trained on sentence embeddings (all-MiniLM-L6-v2) concatenated with verbosity features (step count, trace length).