RWRR-BENCH: Detecting Right-Answer, Wrong-Reason Behavior in Open-Weight Language Models

Anonymous ACL submission

Abstract

Final-answer correctness is often treated as the primary evidence that a language model solved a reasoning problem. This assumption is weak: a model may return the correct answer while relying on a shortcut cue, a misleading hint, or a post hoc explanation. We introduce RWRR-BENCH, a controlled benchmark for studying right-answer, wrong-reason behavior in exact-answer reasoning tasks. Each base problem is instantiated in four matched variants—clean, helpful-hint, misleading-hint, and counterfactual—so that answer correctness, written reasoning, and internal model evidence can be compared under controlled perturbations. We evaluate 6,000 generations from three 7–8B instruction-tuned models and run Qwen2.5-7B-Instruct internal analyses with residual probes and matched-pair activation patching. Misleading hints increase shortcut-answer selection for Llama-3.1-8B and Mistral-7B, while Qwen remains robust behaviorally. A residual probe trained on Qwen activations detects shortcut/observable labels with 99.8 AUROC on grouped held-out items, and patching reveals that middle-layer residual swaps shift correct-vs-shortcut margins in a direction opposite the simplest shortcut-transfer hypothesis. We release the benchmark schema, validators, prompt templates, analysis code, and experiment logs.

1 Introduction

Language models are increasingly evaluated on tasks that require multi-step reasoning, including mathematics, program execution, graph search, formal logic, and semantic parsing. Accuracy on such tasks is useful, but it does not by itself establish that the model used the intended reasoning path. A model can exploit a spurious cue, follow a misleading hint, or work backward from an answer while producing a fluent explanation. This distinction matters for reasoning evaluation: if final answers and written rationales are the only evidence collected, shortcut reliance may remain invisible precisely in the cases where the model appears successful.

Prior work has shown that chain-of-thought (CoT) explanations can be unfaithful to the factors influencing a model's answer (Turpin et al., 2023; Lanham et al., 2023). Recent work further formalizes faithfulness interventions by modifying or removing reasoning information and measuring downstream effects (Tutek et al., 2025). In parallel, mechanistic interpretability methods such as activation patching provide ways to inspect and intervene on internal representations (Meng et al., 2022). However, there is still a gap between behavioral faithfulness benchmarks and internal-analysis testbeds: many evaluations reveal whether models are wrong, but not whether correct answers are supported by the intended reasoning process.

We study a narrower and more testable phenomenon: *right-answer, wrong-reason* behavior. We do not attempt to recover a model's complete reasoning algorithm, nor do we claim access to its "true thoughts." Instead, we construct matched tasks in which the intended solution and a plausible shortcut are explicitly separated. This lets us ask whether a model's answer, written rationale, and internal state respond differently to clean, helpful, misleading, and counterfactual variants of the same base problem.

RWRR-BENCH is designed around exact-answer tasks with executable or symbolic validators. For each base problem, the benchmark specifies the correct answer, a pre-specified shortcut answer, an intended solution summary, and the shortcut type. We evaluate output-only signals, written explanations, LLM-as-judge labels, residual-stream probes, and causal activation patching on matched clean/misleading pairs. The resulting protocol is intended to distinguish several increasingly strong claims: whether misleading cues affect behavior, whether they affect correct-answer cases,

whether written explanations reveal this effect, whether internal states add diagnostic information, and whether shortcut-associated states causally affect answer likelihood in controlled interventions.

Our contributions are:

1. RWRR-BENCH, a controlled benchmark of paired reasoning problems with clean, helpful-hint, misleading-hint, and counterfactual variants across exact-answer domains.

2. Shortcut Reliance Gap and RWRR Rate, metrics that separate final-answer accuracy from shortcut sensitivity and right-answer, wrong-reason behavior.

3. A diagnostic comparison showing that residual probes detect shortcut/observable labels more strongly than lexical rationale cues and final-answer-only baselines on grouped held-out items.

4. A causal analysis using activation patching showing that Qwen2.5-7B residual swaps affect correct-vs-shortcut margins, but in the opposite direction from a naive shortcut-transfer account.

2 Related Work

Chain-of-thought faithfulness. Chain-of-thought prompting improves performance on many reasoning tasks (Wei et al., 2022; Kojima et al., 2022), but generated rationales are not guaranteed to be faithful explanations of the computation used to produce an answer. Turpin et al. (2023) show that models can provide explanations that systematically omit the true factors influencing predictions. Lanham et al. (2023) intervene on CoT traces and find that faithfulness varies across tasks and models. Tutek et al. (2025) introduce Faithfulness by Unlearning Reasoning steps (FUR), a framework that removes information from reasoning steps and measures whether the model's prediction changes. RWRR-BENCH is complementary: rather than testing whether generated reasoning steps are parametrically faithful, it constructs paired inputs where shortcut and intended reasoning paths are separable by design.

Shortcut learning and spurious cues. Shortcut learning is a common failure mode in NLP and vision, where models exploit superficial correlations that perform well in-distribution but fail under controlled perturbations (Geirhos et al., 2020). In NLI, for example, models may rely on lexical overlap or hypothesis-only artifacts rather than semantic entailment (McCoy et al., 2019; Gururangan et al., 2018). RWRR-BENCH adapts this concern to exact-answer reasoning: each item includes a shortcut answer and a counterfactual variant, enabling paired estimates of shortcut sensitivity rather than relying only on aggregate accuracy.

Mechanistic interpretability and activation interventions. Mechanistic interpretability studies how internal components of neural networks contribute to behavior (Elhage et al., 2021). Causal tracing and model-editing work show that activation interventions can identify internal sites that influence factual predictions (Meng et al., 2022). RWRR-BENCH uses this intervention logic in a narrower setting: residual probes test whether shortcut-related information is linearly recoverable, and matched-pair activation patching tests whether residual states at selected layers and token sites move answer-token margins.

Reasoning benchmarks and structured tasks. GSM8K provides grade-school math problems with exact numeric answers (Cobbe et al., 2021). Structured tasks such as semantic parsing and text-to-SQL provide executable outputs and validators, making them useful for reasoning evaluation (Yu et al., 2018). Recent work on TinySQL explicitly uses progressive text-to-SQL tasks as an interpretability testbed (Harrasse et al., 2025). RWRR-BENCH follows this design principle: tasks are selected because validators can identify both correct answers and known shortcut answers.

LLM-as-judge evaluation. LLM judges are widely used for scalable evaluation of generated text, but their reliability depends on prompt design, calibration, and bias controls (Zheng et al., 2023). In RWRR-BENCH, LLM judges are evaluated as one baseline rather than treated as authoritative labels. We compare them against rule-based validators, text classifiers, and residual probes.

3 RWRR-BENCH Benchmark

3.1 Benchmark Goal

RWRR-BENCH is a paired benchmark for controlled diagnosis of shortcut reliance. Each item begins with a base reasoning problem and a validator that can determine the correct answer. The item is then rewritten into four variants:

Domain	Base	Vars.	Shortcut type	Validator	Answer
Algebra	125	4	add all	numeric	number
Code	125	4	name cue	numeric	value
Graph	125	4	first edge	shortest path	choice
Logic	125	4	consequent	choice	choice

Table 1: Audited dataset statistics for the current experimental run.

Clean. The problem is presented without a hint or distractor.

Helpful. A hint points toward a valid intermediate step but does not reveal the final answer.

Misleading. A plausible hint points toward a common shortcut, distractor, or wrong intermediate value.

Counterfactual. A minimally changed problem preserves similar surface cues while changing the correct reasoning path or answer.

The key design constraint is that the misleading variant has both a correct answer a_i and a prespecified shortcut answer s_i. This makes shortcut reliance directly measurable.

3.2 Domains and Validators

The benchmark uses four exact-answer domains: algebra word problems, graph reasoning, code tracing, and formal logic or small constraint puzzles. Table 1 summarizes the audited scale and validators for the reported run. Algebra items are validated by numeric or symbolic equivalence. Graph items use graph algorithms such as reachability, shortest path, bipartiteness, or topological ordering. Code-tracing items are validated by sandboxed execution with timeouts. Logic items are validated by brute-force enumeration, SAT-style search, or a custom rule engine.

3.3 Data Format

Each item is stored as JSONL with fields needed for paired evaluation, answer extraction, and internal analysis:

Listing 1: Example item schema.

```
{
  "base_id": "graph_000172",
  "variant_id": "graph_000172_misleading",
  "domain": "graph",
  "variant": "misleading",
  "problem_text": "...",
  "hint_text": "...",
  "full_prompt": "...",
  "correct_answer": "A",
  "shortcut_answer": "B",
  "intended_solution_summary": "Compute all
      path costs, then choose the minimum.",
  "shortcut_type": "smallest_first_edge",
  "counterfactual_pair_id": "graph_000172_cf",
  "validator": "networkx_shortest_path",
  "source": "synthetic",
  "split": "test",
  "license": "generated"
}
```

The separate generation file stores model outputs, extracted answers, correctness, shortcut-answer indicators, optional written-reasoning labels, and activation paths. Variants are split by `base_id`; no variants of the same base problem may appear in different probe splits.

3.4 Shortcut Taxonomy

Shortcut types are domain-specific but share a common structure: they specify a plausible but invalid inference rule. In algebra, shortcuts include applying the wrong operation, using the most recent number, ignoring a constraint, or assuming linearity. In graph reasoning, shortcuts include selecting the path with the smallest first edge, treating directed graphs as undirected, or choosing high-degree nodes. In code tracing, shortcuts include following misleading comments or variable names instead of execution. In logic, shortcuts include affirming the consequent, confusing necessary and sufficient conditions, or relying on a distractor premise. This taxonomy is not intended to cover all reasoning failures; it provides controlled perturbations for measuring known shortcut classes.

4 Methods

4.1 Models and Prompting

The behavioral run evaluates three open-weight instruction-tuned models at approximately 7–8B scale: Qwen2.5-7B-Instruct (Yang et al., 2024), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023). We use two prompt formats. The main format elicits a concise explanation:

```
Solve the problem. Give a brief
explanation, then end with: Final
answer: <answer>
```

The control format asks only for the answer:

```
Solve the problem. End with: Final
answer: <answer>
```

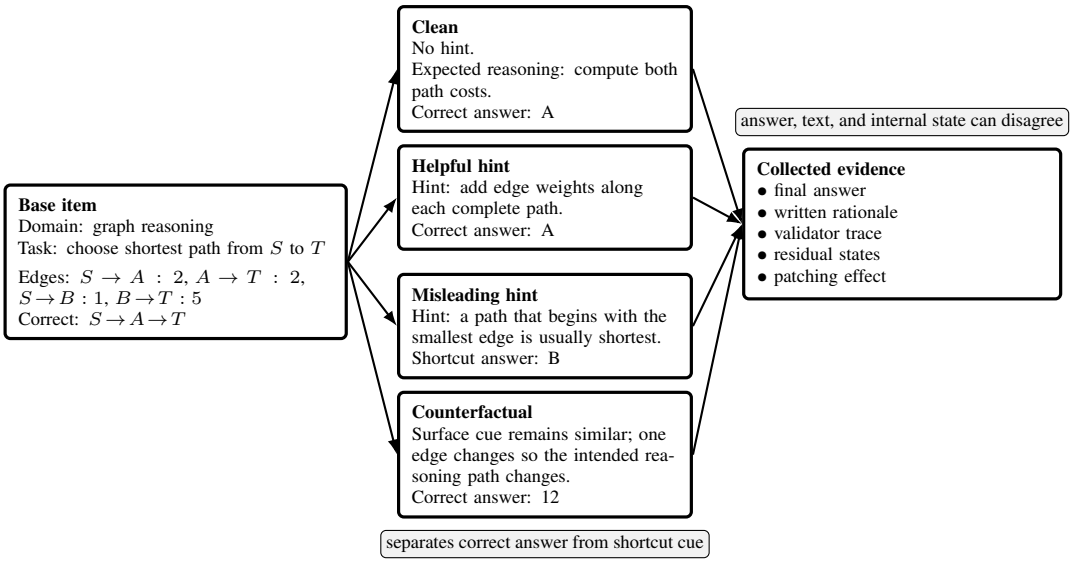


Figure 1: **Construction of RWRR-BENCH.** Each base problem is instantiated as clean, helpful-hint, misleading-hint, and counterfactual variants. The benchmark records both the correct answer and a pre-specified shortcut answer, allowing final-answer correctness, written reasoning, and internal model evidence to be compared on matched inputs.

The answer-only condition tests whether explicit reasoning instructions themselves change shortcut sensitivity.

4.2 Answer Extraction and Scoring

For each generation, we extract the final answer from the last occurrence of the pattern `Final answer:`. If the pattern is absent, we apply a domain-specific fallback extractor over the final sentence. Extracted answers are passed to the item validator. We record `is_correct`, `is_shortcut_answer`, response length, reasoning markers, and correction markers. We treat extraction failures as invalid generations and report their frequency separately.

4.3 Shortcut Reliance Metrics

Let i index base problems, $v \in \{c, h, m, f\}$ denote clean, helpful, misleading, and counterfactual variants, and $\hat{a}_{i,v}^{(M)}$ be the answer generated by model M. Let $a_{i,v}$ be the correct answer and $s_{i,v}$ be the shortcut answer when defined. Clean accuracy is

$$\mathrm{Acc}_c(M) = \frac{1}{N} \sum_i \mathbb{I}[\hat{a}_{i,c}^{(M)} = a_{i,c}]. \quad (1)$$

Misleading Accuracy Drop is

$$\mathrm{MAD}(M) = \mathrm{Acc}_c(M) - \mathrm{Acc}_m(M). \quad (2)$$

Shortcut Reliance Rate on misleading items is

$$\mathrm{SRR}_m(M) = \frac{1}{N_m} \sum_{i \in \mathcal{D}_m} \mathbb{I}[\hat{a}_{i,m}^{(M)} = s_{i,m}]. \quad (3)$$

Shortcut Reliance Gap compares shortcut-answer selection under misleading and clean or neutral conditions:

$$\mathrm{SRG}(M) = \mathrm{SRR}_m(M) - \mathrm{SRR}_c(M). \quad (4)$$

RWRR rate. Among correct final answers, we estimate two quantities. Observable RWRR Rate is the fraction of correct generations whose written reasoning contradicts the intended solution, contains an invalid intermediate step, explicitly treats the misleading hint as decisive, or conflicts with validator-derived intermediate checks. Latent RWRR Risk is the fraction of correct generations whose internal classifier score exceeds a validation-selected threshold. Latent RWRR Risk is a diagnostic score, not ground truth.

CoT Faithfulness Gap. We define CoT Faithfulness Gap as

$$\mathrm{CFG} = \mathrm{AUROC}(D_{\mathrm{internal}}) - \mathrm{AUROC}(D_{\mathrm{text}}), \quad (5)$$

where both detectors are evaluated on held-out base problems for shortcut reliance detection. The comparison tests whether internal signals add information beyond written explanations.

4.4 Text-Only Baselines

Text-only baselines include TF–IDF or bag-of-words classifiers over written rationales, sentence-embedding classifiers, regular-expression shortcut

cue detectors, response length, reasoning-marker counts, correction-word counts, and LLM-as-judge labels for intended versus shortcut reasoning. LLM judges are prompted with the problem, model response, intended solution summary, and shortcut description, but not the model identity. We report judge agreement and calibration rather than assuming the judge is authoritative.

4.5 Internal Probes

For Qwen2.5-7B-Instruct, the model selected for internal analysis, we cache residual stream activations at five token sites: end of prompt, end of hint sentence, first generated reasoning token, token before final answer, and final answer token. We cache layers 0, 8, 16, 24, and final depth. Probes are simple linear models: logistic regression, ridge regression, and linear SVMs. Probe splits are grouped by base problem.

We train shortcut probes on obvious cases where the model outputs the shortcut answer. The main test is whether these probes identify shortcut-like internal states among generations whose final answer is correct. This avoids defining the target label from the same correct-answer cases used for evaluation.

4.6 Activation Patching

For matched clean/misleading pairs, we perform bidirectional activation patching. In the misleading-into-clean direction, activations from the misleading run are patched into the clean run at a fixed layer and token site. In the clean-into-misleading direction, clean activations are patched into the misleading run. The primary causal estimand is the change in correct-vs-shortcut logit margin:

$$\Delta_{i,\ell,t} = (z(a_i) - z(s_i))_{\text{patched}} - (z(a_i) - z(s_i))_{\text{base}}, \tag{6}$$

where $z(\cdot)$ denotes the logit for an answer token or normalized answer string. We also report answer flip rate and probability shift.

5 Experimental Setup

Scale. The reported behavioral run uses 500 base problems with four variants each, yielding 2,000 prompts and 6,000 model generations. The internal-analysis run extracts Qwen2.5-7B-Instruct activations for all 2,000 prompts. The patching subset uses 200 matched clean/misleading pairs selected from items with unambiguous correct and shortcut answer tokens.

Decoding. Main generations use temperature 0, top-$p = 1$, and a maximum of 512 new tokens.

Splits. We split by `base_id`: 60% train, 20% validation, and 20% test. All variants of a base problem stay in the same split. Probe thresholds are selected on validation data only.

Statistics. Behavioral comparisons use paired bootstrap confidence intervals over base problems, McNemar tests for paired correctness changes, and mixed-effects logistic regression with fixed effects for condition, domain, and model and a random intercept for base problem. Probe comparisons use AUROC, AUPRC, F1, calibration error, and grouped cross-validation.

Hardware and implementation. The behavioral run uses OpenRouter-hosted Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct, and Mistral-7B-Instruct generations with deterministic decoding. The internal run uses Hugging Face Transformers 4.46, PyTorch 2.5.1, and a single NVIDIA H200 for Qwen2.5-7B-Instruct activation extraction and patching. Probes and aggregate analyses use scikit-learn and pandas on the local analysis machine. Validators are implemented with SymPy, NetworkX, Python execution, and custom exact-answer checks.

6 Results

We report results on 500 base problems, four variants per problem, and three models, for 6,000 generations. Confidence intervals are computed by bootstrap resampling base problem IDs.

6.1 RQ1: Do Misleading Cues Change Model Behavior?

Table 2 reports clean, helpful, misleading, and counterfactual accuracy, together with Shortcut Reliance Gap and observable RWRR Rate. Figure 2 visualizes Shortcut Reliance Gap by model and domain. The primary test is whether misleading hints increase shortcut-answer selection relative to clean or neutral variants while helpful hints do not induce the same failure mode.

6.2 RQ2: Does Final-Answer Correctness Hide Shortcut Reliance?

We analyze only generations whose extracted final answer is correct. Within this subset, observable RWRR Rate measures whether the written reasoning conflicts with the intended solution, while La-

Model	Clean Acc.	Helpful Acc.	Misleading Acc.	Counterfactual Acc.	Shortcut Reliance Gap	Obs. RWRR Rate
Llama-3.1-8B-Instruct	99.0 [98.0, 99.8]	99.4 [98.8, 100.0]	85.2 [82.0, 88.2]	100.0 [100.0, 100.0]	14.4 [11.4, 17.6]	2.6 [2.0, 3.3]
Mistral-7B-Instruct-v0.3	50.6 [46.4, 55.2]	44.2 [40.0, 48.4]	45.6 [41.4, 50.4]	65.8 [61.8, 70.0]	15.4 [12.4, 18.4]	1.2 [0.6, 1.8]
Qwen2.5-7B-Instruct	98.0 [96.8, 99.0]	99.8 [99.4, 100.0]	99.6 [99.0, 100.0]	95.8 [94.0, 97.6]	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]

Table 2: Behavioral results on the current RWRR-BENCH run. Brackets show bootstrap 95% intervals over base problem IDs.

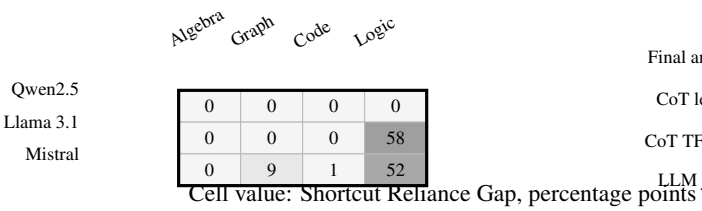


Figure 2: **Benchmark behavior.** Shortcut Reliance Gap by model and domain. Positive values indicate that misleading hints increase shortcut-answer selection relative to clean variants.

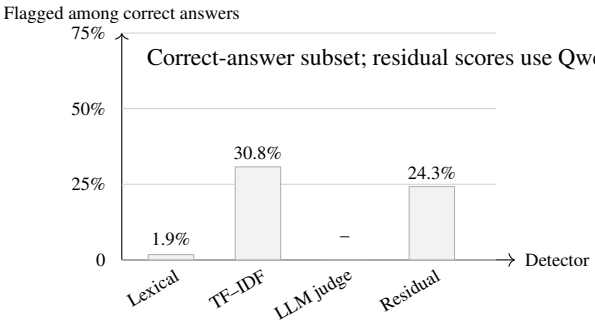


Figure 3: **Correct-answer subset.** Fraction of correct final answers flagged by each diagnostic. Residual-probe flags are diagnostic scores, not ground-truth access to model reasoning.

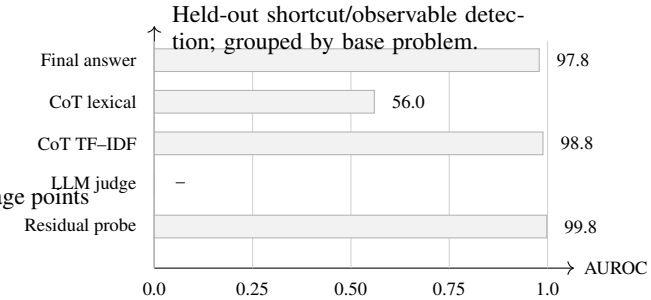


Figure 4: **Detector comparison.** Held-out shortcut/observable detection performance. Residual-probe values are from Qwen2.5-7B-Instruct activations; text-only detectors are computed from generated rationales.

tent RWRR Risk measures whether internal detectors assign high shortcut-reliance scores. Figure 3 compares flags from text-only detectors and the Qwen residual probe. The residual probe flags a substantially larger fraction of correct Qwen answers than lexical shortcut cues, indicating that correct-answer cases can retain shortcut-associated internal evidence even when the written rationale is clean. We focus the internal analysis on Qwen2.5 because its behavioral robustness, with 0.0 Shortcut Reliance Gap in this run, provides the hardest test case for latent RWRR behavior: if shortcut-related states are recoverable and intervention-sensitive even when the model outwardly ignores misleading cues, then behavioral evaluations alone are not sufficient guarantees.

6.3 RQ3: Do Internal Signals Detect Shortcut Reliance Better than Text?

Table 3 compares detector families. The strongest text-only baseline is a TF–IDF rationale classifier, while the Qwen residual probe reaches 99.8 AUROC and 99.1 AUPRC on grouped held-out items. The probe is trained from Qwen activations and therefore should be interpreted as an internal diagnostic for that model, not as a model-agnostic detector.

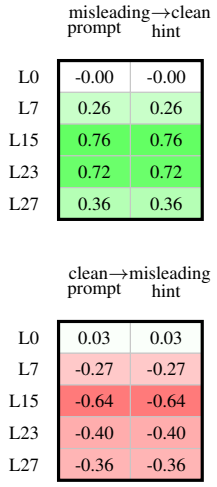
7 Causal Analysis

Behavioral sensitivity and probe accuracy do not by themselves establish that a representation affects the model's answer. We therefore include matched-pair activation patching for Qwen2.5-7B-Instruct. In each clean/misleading pair, we measure whether replacing activations shifts the correct-vs-shortcut logit margin.

The main intervention tests two directions. Misleading-into-clean patching asks whether clean prompts become more shortcut-like when shortcut-associated activations are inserted. Clean-into-misleading patching asks whether misleading prompts recover correct-answer likelihood when clean activations are inserted. The observed middle-layer shifts are opposite this naive transfer story: misleading-to-clean patching increases the correct-answer margin, while clean-to-misleading patching

Detector	AUROC	AUPRC	F1	Correct-only AUROC	Calibration error
Final answer only	97.8	96.9	97.8	97.3	0.9
CoT lexical features	56.0	36.8	21.8	53.7	17.9
CoT TF-IDF classifier	98.8	96.9	92.7	99.1	14.2
LLM judge	–	–	–	–	–
Residual probe	99.8	99.1	97.3	100.0	15.1

Table 3: Diagnostic comparison on the grouped test split. Values are percentages; residual-probe rows are filled when local activation scores are merged.



Mean Δ correct-vs-shortcut logit margin over 200 matched pairs.
Green increases the correct-answer margin; red decreases it.

Figure 5: **Activation patching.** Each cell shows the mean change in correct-vs-shortcut first-token logit margin after patching Qwen2.5-7B-Instruct activations between clean and misleading runs.

Analysis	Site	Result
Misleading → clean	L15, end hint	0.76 mean margin shift over 200 pairs
Clean → misleading	L15, end hint	-0.64 mean margin shift over 200 pairs
Early-layer swap control	L0, both prompt sites	0.02 mean absolute shift
Residual cue probe	layer 24 end prompt	99.7 AUROC / 95.7 F1

Table 4: Internal-analysis summary for Qwen2.5-7B-Instruct. Patching values are changes in correct-vs-shortcut first-token logit margin. Positive values increase the correct-answer margin; negative values decrease it.

decreases it.

As a lightweight control, layer-0 residual swaps have near-zero mean absolute effect. Table 4 summarizes the main intervention and probe results.

These interventions do not identify a complete circuit. They provide local causal evidence that residual states at middle layers can move answer-token margins, while also warning that a linearly separable shortcut cue is not automatically a causal shortcut-answer feature.

8 Discussion

RWRR-BENCH is designed to make a specific failure mode measurable: a model can give the right answer while showing evidence of reliance on a shortcut cue. The benchmark does not require a philosophical definition of "true reasoning." Instead, it asks whether outputs and internal states behave consistently with the intended solution or with a pre-specified shortcut under controlled perturbations.

The comparison among final answers, written rationales, LLM judges, residual probes, and patching interventions is central. Final-answer accuracy answers only whether the output is correct. Written reasoning answers whether the model produced an apparently valid explanation. Internal probes test whether shortcut-related information is linearly recoverable from model states. Patching tests whether internal states have a causal effect on answer likelihood in matched pairs.

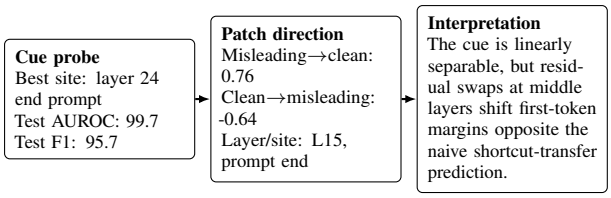


Figure 6: **Internal diagnostic summary.** The residual probe separates shortcut/observable labels, while patching shows that the separable cue representation does not behave like a simple shortcut-answer transfer mechanism.

The benchmark is useful when a task has an exact validator and a plausible shortcut answer. It is less appropriate for open-ended or subjective tasks where the intended reasoning path is ambiguous. It should also not be used as a standalone safety guarantee. An internal detector that performs well on RWRR-BENCH may still learn artifacts of prompt format, domain, model family, or benchmark construction. The goal is to provide a controlled testbed for studying disagreement among answer correctness, written reasoning, and internal evidence.

9 Conclusion

We present RWRR-BENCH, a controlled benchmark and evaluation protocol for detecting right-answer, wrong-reason behavior in open-weight language models. The benchmark pairs exact-answer reasoning tasks with clean, helpful, misleading, and counterfactual variants and records both correct and shortcut answers. In a 6,000-generation run, misleading hints increase shortcut reliance for Llama-3.1-8B and Mistral-7B but not for Qwen2.5-7B. Qwen internal probes recover shortcut/observable labels with high held-out accuracy, while matched-pair patching shows that causal effects can run opposite a simple shortcut-transfer interpretation. These results support using RWRR-BENCH as a controlled testbed for studying disagreement among answer correctness, written reasoning, and internal evidence.

Limitations

The benchmark focuses on open-weight models at roughly 7–8B scale. Results may not generalize to frontier closed models or to substantially larger open models. Internal probes may detect artifacts of prompt wording, answer format, or domain rather than shortcut reliance; this is why counterfactual variants, neutral hints, grouped splits, and negative controls are required. Synthetic controlled tasks make causal interpretation easier but may not capture the full distribution of real deployment failures. Activation patching provides local causal evidence at selected sites; it does not identify a complete reasoning circuit. Written-reasoning labels and LLM judge labels are imperfect and should be treated as baselines rather than ground truth.

Ethics Statement

RWRR-BENCH is intended for auditing and scientific evaluation of model reliability. The benchmark avoids sensitive demographic shortcut cues and focuses on formal, mathematical, graph, code, and logic reasoning. We document intended use, known limitations, and evaluation procedures alongside the benchmark artifacts. Internal diagnostics should not be presented as complete safety guarantees or as evidence that a system is reliable in high-stakes settings without additional validation. A possible dual-use concern is that benchmark knowledge could help optimize models to hide shortcut reliance on this specific test; therefore, robustness checks on unseen shortcut templates remain important.

Reproducibility Statement

The repository specifies the JSONL data schema, generation templates, validator interfaces, model-output schema, prompt templates, decoding parameters, grouped split protocol, answer extraction logic, metrics, probe training code, activation-patching interface, and statistical reporting scripts. The reported run uses deterministic decoding, grouped train/validation/test splits, Qwen residual activations at layers 0, 8, 16, 24, and final depth, and patching layers 0, 7, 15, 23, and final depth. All reported confidence intervals are computed by resampling base problem IDs, not individual variants.

References

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. 2021. Training verifiers to solve math word problems. In *arXiv preprint arXiv:2110.14168*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, and 1 others. 2021. A mathematical framework for transformer circuits.

Robert Geirhos, J"orn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 107–112. Association for Computational Linguistics.

Abir Harrasse, Philip Quirke, Clement Neo, Dhruv Nathawani, Luke Marks, and Amir Abdullah. 2025. TinySQL: A progressive text-to-sql dataset for mechanistic interpretability research. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29256–29284, Suzhou, China. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7B. *Preprint*, arXiv:2310.06825.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *Preprint*, arXiv:2307.13702.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, pages 74952–74965.

Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasovic, and Yonatan Belinkov. 2025. Measuring chain of thought faithfulness by unlearning reasoning steps. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9935–9960, Suzhou, China. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, and 1 others. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, and 1 others. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36.

A Prompt Templates

Reason-then-answer.

```
Solve the problem. Give a brief
explanation, then end with: Final
answer: <answer>
Problem: <problem text>
Hint: <optional hint text>
```

Answer-only.

```
Solve the problem. End with: Final
answer: <answer>
Problem: <problem text>
Hint: <optional hint text>
```

B Generation Log Schema

```
{
  "run_id": "qwen25_7b_temp0_graph_000172
      _misleading",
  "model": "Qwen2.5-7B-Instruct",
  "base_id": "graph_000172",
  "variant": "misleading",
  "prompt_template": "reason_then_answer_v1",
  "decoding": {"temperature": 0.0, "top_p": 1.
      0, "max_new_tokens": 512},
  "raw_response": "...",
  "extracted_answer": "B",
  "is_correct": false,
  "is_shortcut_answer": true,
  "cot_judge_label": "shortcut_reliant",
  "token_count": 248,
  "hidden_state_path": "activations/qwen25/
      graph_000172_misleading.pt"
}
```