Same Benchmark Score, Different Failures: Atomic Capacities for Diagnosing Language Model Failure Modes

Anonymous ACL submission

Abstract

Benchmarks typically report aggregate scores over broad task domains such as knowledge or reasoning, but these scores do not reveal which behaviour failed or what intervention would repair it. We propose *atomic capacities*: paired minimal perturbations that isolate repair-relevant failure modes. We instantiate nine atomic capacities spanning *Knowledge*, *Reasoning*, and *Hybrid Composition*, and evaluate them on paired original–variant prompts using behavioural log-probability measures. Mainly focusing on Qwen3 models, we find that models with similar domain-level performance can exhibit substantially different capacity profiles, implying different repair targets. We further show that corruption perturbations separate into two distinct regimes: some reduce confidence in the gold answer without changing the preferred decision, while others flip the model toward the planted incorrect answer. Finally, we show that a standard zero-shot Chain-of-Thought intervention is not uniformly beneficial across capacities and can worsen robustness under misleading cues. Our results suggest that benchmark categories are too coarse for diagnosis and that model evaluation should instead target repair-relevant atomic capacities.

1 Introduction

Benchmarks identify when a model fails, but not always why (Wang et al., 2018; Ribeiro et al., 2020; D'Amour et al., 2022; Wang et al., 2024). A domain-level score aggregates performance over a broad task domain, such as knowledge or reasoning, and is useful for tracking overall progress. However, it is often too coarse for diagnosis as the same aggregate score can arise from qualitatively different underlying failure modes, each requiring a different intervention strategy. For example, a model that performs poorly on knowledge tasks may fail to surface a stored fact, struggle to retrieve information under specific phrasing, or be easily overridden by an injected false claim. While all three may produce the same domain-level outcome, they imply different underlying bottlenecks. Figure 1 illustrates this concretely: two models with the same knowledge-domain score can exhibit entirely different failure profiles, one vulnerable to an injected false claim and the other brittle to question phrasing, even though their aggregate benchmark performance is identical. The domain-level score therefore does not identify what failed or how it should be addressed.

Building on this observation, we complement existing domain-level benchmark evaluation by examining how models respond to controlled prompt perturbations that isolate distinct failure modes. Rather than measuring only whether a model succeeds or fails, we identify the specific behavioural bottlenecks that drive failure and therefore the intervention strategies they imply. Such failure-level diagnosis is crucial for model improvement, as qualitatively different failure modes can produce the same aggregate benchmark score while requiring different interventions. This makes the evaluation actionable as different atomic-capacity failures suggest different remedies, such as paraphrase-augmented training, decomposition scaffolding, retrieval or source-verification, and contradiction-aware decoding. Atomic capacities therefore turn a broad benchmark failure into a concrete repair hypothesis.

We term these perturbation-defined behavioural units *atomic capacities*. An atomic capacity is the smallest perturbation-defined behavioural unit that preserves the underlying task while isolating a single repair-relevant failure mode. A capacity is *atomic* when: (i) it can be isolated through a paired minimal perturbation contrasting an original prompt with a minimally modified variant; and (ii) the resulting behavioural change corresponds to a distinct failure mode with different intervention implications. Here, "atomic" refers operationally

to minimally isolated perturbation settings rather than irreducible cognitive primitives.

We instantiate this framework through a 3×3 design spanning three task domains and three perturbation types. The domains, *Knowledge*, *Reasoning*, and *Hybrid Composition*, capture complementary settings in which language models must succeed: factual retrieval, procedural reasoning, and multi-step fact–reasoning composition. Within each domain, we apply three perturbation types. *Augment* perturbations add missing support, *Ablate* perturbations remove supporting cues, and *Corrupt* perturbations inject misleading competing signals. Crossing the three domains with the three perturbation types yields nine atomic capacities. Together, these capacities decompose broad benchmark failures into intervention-relevant behavioural bottlenecks that are otherwise hidden by aggregate domain-level scores.

Our contributions are as follows:

- We define *atomic capacity* as an intervention-relevant behavioural diagnostic unit and instantiate it through nine paired perturbation tests spanning three task domains and three perturbation modes (§3).
- We demonstrate that models with similar domain-level benchmark performance can exhibit substantially different behavioural bottlenecks and intervention targets (§6.1).
- We establish that corruption perturbations require two separate measures, confidence sensitivity and decision robustness, as the two can diverge substantially (§6.2).
- We identify that standard zero-shot Chain-of-Thought prefixing (Kojima et al., 2022) is not uniformly beneficial under corruption perturbations (§6.3).

2 Related Work

Benchmarks measure broad capability, but aggregated scores can hide distinct failure mechanisms (Ribeiro et al., 2020; Mondorf and Plank, 2024). Behavioural evaluations show that models with similar scores can fail for different reasons under controlled perturbations (McCoy et al., 2019; Ribeiro et al., 2020; Gardner et al., 2020), and underspecification studies argue that benchmark success does not identify the underlying capability being exercised (D'Amour et al., 2022). Factual knowledge in language models is sensitive to retrieval conditions, including paraphrasing (Liu and Meng, 2026), misleading context, and counterfactual distractors (Lin et al., 2022; Shi et al., 2023). These failures motivate different remedies, such as retrieval augmentation (Lewis et al., 2020; Shi et al., 2024), paraphrase-based training (Morris et al., 2020; Fu et al., 2024), and guarded decoding (Xu et al., 2024; Dong et al., 2025; Qiu et al., 2025). Chain-of-Thought (CoT) prompting improves reasoning on many tasks (Wei et al., 2022b), but can also introduce new sensitivities (Wang et al., 2023; Chia et al., 2023). Scaling studies further suggest that capability improvements are uneven across tasks and model families (Kaplan et al., 2020; Wei et al., 2022a; Lourie et al., 2025; Maia Polo et al., 2026). Even within the same model family, performance trajectories can diverge across factual recall, compositional reasoning, robustness, and calibration as model scale increases (Ganguli et al., 2022; Srivastava et al., 2023; Schaeffer et al., 2023). Our work follows this behavioural-evaluation line, but changes the unit of analysis: instead of treating broad benchmark families as the diagnostic unit (Wu et al., 2025; Jin et al., 2025), we use paired perturbations to separate repair-relevant failure modes within a family.

3 Atomic Capacity Framework

Terminology. Throughout this paper, we use *task domain* to refer to the three broad task settings considered here: Knowledge, Reasoning, and Hybrid composition. A *domain-level score* is the aggregate score obtained by pooling items within one such task domain; this is the coarse quantity that we argue is insufficient for diagnosis. An *item* is the underlying source example from which prompts are derived. An *original prompt* is the unperturbed question for that item, and a *variant prompt* is the corresponding Augment, Ablate, or Corrupt perturbation. A *paired test* compares one original prompt with one variant prompt. An *atomic capacity* is the repair-relevant diagnostic unit defined by one task domain and one perturbation type, such as K-Cor or H-Aug. We reserve *model family* for model-series terminology, such as the Qwen3 family.

Existing benchmark evaluations collapse failures into broad task domains, such as knowledge or reasoning, but models with similar domain-level performance can fail for fundamentally different reasons. Our framework instead organises evaluation around *intervention-relevant failure modes*, using controlled paired perturbations to isolate the

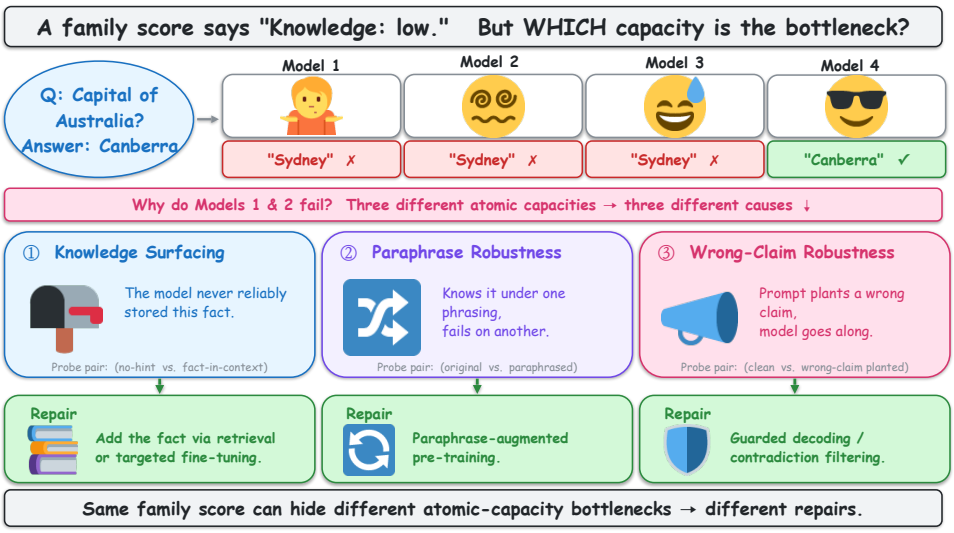

Figure 1: Illustrative diagnostic example. Model 1 may fail to surface the stored fact, Model 2 may be brittle to paraphrasing, and Model 3 may be susceptible to a planted wrong claim. Because each bottleneck suggests a different repair, the capacity-level analysis identifies the specific failure mode hidden by the aggregate score.

behavioural bottlenecks responsible for failure. Unlike prior perturbation benchmarks that primarily treat perturbations as robustness tests (Jin et al., 2020; Ribeiro et al., 2020; Gardner et al., 2020), we use perturbations as diagnostic units. Each perturbation is designed to expose a distinct behavioural dependency and thereby suggest a different intervention strategy.

We instantiate this framework through a 3×3 design that crosses three task domains with three perturbation types (Table 1). Each resulting cell defines an *atomic capacity*: a minimally isolated behavioural dependency revealed through paired perturbation. The same domain-level benchmark failure can therefore be attributed to different atomic capacities, each associated with a different behavioural bottleneck and intervention hypothesis.

Task domains. The three domains, namely *Knowledge*, *Reasoning*, and *Hybrid Composition*, capture complementary settings in which language models must succeed. *Knowledge* tasks test retrieval of stored factual associations, *Reasoning* tasks test the application of explicit rules or arithmetic procedures, and *Hybrid Composition* tasks require chaining retrieved facts through intermediate reasoning steps, making them sensitive to failures in either component. These domains are not intended as an exhaustive taxonomy of language-model cognition. Rather, they provide a controlled testbed spanning factual, procedural, and compositional demands within a tractable experimental design.

Perturbation types. Within each domain, each atomic capacity is defined by a perturbation type, referring to the specific modification applied to the prompt in order to diagnose a behavioural dependency. Let o and v denote correctness on the original and variant prompts, respectively. An *Augment* perturbation adds supporting information absent from the original prompt and tests whether the model can productively use missing support once it is made explicit. An Augment cell is diagnostic when the model fails on the original prompt but succeeds on the augmented variant ($\neg o \wedge v$). An *Ablate* perturbation removes a cue or scaffold present in the original prompt. It tests whether successful behaviour depends on explicit support already embedded in the prompt. An Ablate cell is diagnostic when the model succeeds on the original prompt but fails once the cue is removed ($o \wedge \neg v$). A *Corrupt* perturbation introduces a plausible but false competing cue, such as a wrong claim or incorrect intermediate value, alongside the original question. It tests susceptibility to misleading contextual evidence. A Corrupt cell is diagnostic when the model succeeds on the original prompt but fails under the competing cue ($o \wedge \neg v$). The *Augment* and *Ablate/Corrupt* criteria are intentionally asymmetric: *Augment* tests whether missing support can be productively used, whereas *Ablate* and *Corrupt* test whether removing or planting a cue induces failure.

Crossing the three domains with the three perturbation types yields nine atomic capacities. Each capacity isolates a different behavioural bottle-

neck and therefore suggests a different intervention strategy. For example, K-Abl (*Paraphrase Robustness*) motivates paraphrase-augmented training, whereas K-Cor (*Wrong-Claim Robustness*) motivates contradiction-aware decoding. Cell codes combine a domain letter (i.e., **K**nowledge, **R**easoning, **H**ybrid) with a perturbation tag (AUG, ABL, COR). For example, H-Cor denotes the Hybrid/Corrupt cell corresponding to *Wrong-Bridge Robustness*. The full capacity grid, definitions, and candidate interventions are shown in Table 1.

Each atomic capacity is measured with a paired test. An item is the underlying example, consisting of a base question, a gold answer, and any support facts needed to construct perturbations. From each item we derive one *original prompt*, which is the unperturbed base question, and one or more *variant prompts*, each produced by applying an *Augment*, *Ablate*, or *Corrupt* perturbation. A paired test compares the original prompt with one variant prompt and asks whether that perturbation exposes the corresponding atomic capacity.

For each paired test, we score the gold answer under both prompts by appending the gold-answer tokens and computing their per-token log-probability. Our main behavioural metric is the paired per-item difference

$$\Delta \log p_i = \log p^{(v)}_{\text{gold},i} - \log p^{(o)}_{\text{gold},i}, \qquad (1)$$

where o denotes the original prompt and v denotes the variant prompt. We report raw means across the 25 items per cell with 1,000-replicate bootstrap confidence intervals. Exact prompt definitions and validator rules are given in Appendix A.6 and Appendix A.7.

4 Dataset Construction

We construct the diagnostic dataset from scratch because existing benchmarks do not provide paired original–variant prompts that isolate a single perturbation per item. An *item* is the underlying source from which an original–variant test pair is generated: a 1-hop Wikidata relation for *Knowledge*, a rule-based arithmetic instantiation for *Reasoning*, or a 2-hop composition for *Hybrid Composition*. We sample 25 items per task domain (75 total). The 25-item-per-domain scale is a tractability choice; broader coverage is a direction for future work.

Variant generation. Each item is paired with the capacity tests applicable to its task domain (Table 1). The original prompt is the item's base question, and each variant prompt applies the corresponding *Augment*, *Ablate*, or *Corrupt* perturbation. We use Claude-Sonnet-4.5 (Anthropic, 2025) to generate the variants from per-test instructions, conditioned on the item and gold answer. The H-Cor / Wrong-Bridge cell uses a separate generation strategy because the planted bridge must imply a competing answer different from the gold answer. Full generation details and per-cell definitions are given in Appendix A.6. Table 2 shows one original–variant pair per capacity, while Figure 5 sketches the construction pipeline.

Quality control and validation. Our dataset construction is gated by three rule-based validators applied after generation. A gold-leakage check rejects any variant whose question text contains the gold answer. A type-match check (*Corrupt* variants only) verifies that the planted incorrect claim is the same semantic type as the gold answer. A load-bearing check (*Ablate* variants only) rejects removals that are merely decorative rather than structurally load-bearing. Failed items trigger targeted regeneration where a repair clause derived from the failing validator is appended to the generation prompt and the item is re-run. Items that still fail after three retries are dropped from that (item, cell) slot. Full validator details are in Appendix A.7.

Dataset composition. The dataset contains 650 prompt rows: 150 original rows, 450 primary-cell variant rows, and 50 Hybrid both-blocked control rows (see Appendix A.9). Each primary variant prompt is paired at analysis time with its corresponding original prompt, yielding 450 paired diagnostic tests from 150 shared originals (Table 3).

We release a 90-pair manual audit template (10 pairs per capacity, stratified by item) as supplementary material. Two PhD-level annotators independently labelled the full template. Cohen's κ on the three per-pair judgments is 0.89, 0.95, and 0.72 for whether the gold is correct under the original prompt, whether it remains correct under the variant, and whether the variant exemplifies its declared perturbation type, respectively. The audit provides an additional quality-control check on whether generated variants preserve the intended answer and instantiate the declared perturbation type. Agreement scores, per-capacity pass rates, and the full annotation protocol are reported in Appendix A.7.

Domain	Atomic Capacity (code)	Assay	Definition	Candidate Intervention
Knowledge	Knowledge Surfacing (K-Aug)	hint	Answer is parametric but did not surface without a cue	Retrieval cueing / hint prompting
	Paraphrase Robustness (K-Abl)	paraphrase	Retrieval cue is surface-form dependent	Paraphrase-augmented pretraining
	Wrong-Claim Robustness (K-Cor)	wrong_claim	A false context overrides the stored fact	Contradiction-aware decoding
Reasoning	Scaffold Activation (R-Aug)	scaffold	Model needs explicit decomposition to apply rule	Decomposition scaffold / planner prompt
	Rule Grounding (R-Abl)	rule_removal	Model was reading the rule off the prompt	In-context rule injection / rule fine-tuning
	Intermediate-Step Robustness (R-Cor)	wrong_intermed	Wrong intermediate value displaces correct chain	Self-consistency / step verification
Hybrid	Bridge-Fact Integration (H-Aug)	explicit_fact	Model lacks the bridge fact	Bridge retrieval / multi-hop RAG
	Retrieval-Block Compensation (H-Abl)	retrieval_blocked	Reasoning cannot fill the missing retrieval	Provide bridge entity explicitly
	Wrong-Bridge Robustness (H-Cor)	wrong_bridge	Wrong bridge contaminates composition	Source verification / bridge consistency

Table 1: The nine atomic capacities. Each cell in the domain $\times$ perturbation-type design names a distinct failure mode and points to a different candidate intervention. **K** Knowledge, **R** Reasoning, **H** Hybrid; AUG Augment, ABL Ablate, COR Corrupt. o and v denote correctness on the original and variant prompts respectively. Augment tests use the $\neg o \wedge v$ criterion; Ablate and Corrupt tests use $o \wedge \neg v$.

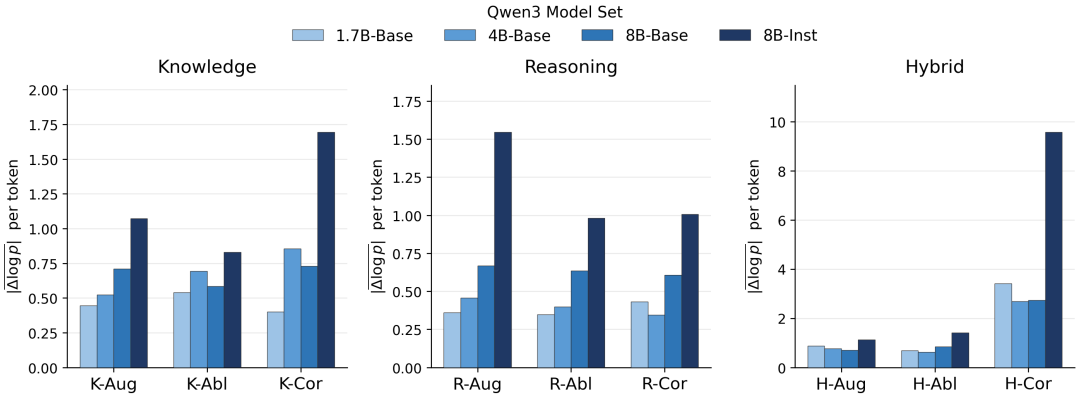

Figure 2: Atomic-capacity profiles across the four main Qwen3 models. Bars show raw mean $|\Delta \log p|$ per token for each capacity. The within-domain variation shows why a single domain-level aggregate is insufficient for identifying the repair-relevant bottleneck; full per-model results are reported in Appendix A.4.

5 Experimental setup

Models. We use the Qwen3 family as the main controlled model series (Qwen Team, 2025). The main experiments report Qwen3-1.7B-Base, Qwen3-4B-Base, Qwen3-8B-Base, and Qwen3-8B-Instruct. This series let us compare scale within a single model family and compare base vs. instruct tuning at the same size. We also run additional experiments on DeepSeek-R1-Distill-Llama-8B (DeepSeek-AI Team, 2025), Llama-3.1-8B-Instruct (Llama Team, 2024), Olmo-3-7B-Instruct (Olmo Team, 2025), and Qwen3-8B for the cross-architecture comparison in Figure 6. A broader Qwen3 / Qwen3.5 (Qwen Team, 2026) sweep of 17 models is reported in Appendix A.4.

Behavioural scoring. For each model, we score the gold answer under both the original prompt and its paired variant. We append the gold-answer tokens to the prompt and compute their per-token log-probability at the appended positions. The same scoring function is used for the original and variant prompts, so the behavioural measure is the paired difference $\Delta \log p$ defined in Eq. 1. We report raw mean $\Delta \log p$ per token with 1,000-replicate bootstrap confidence intervals over the 25 items in each cell.

Two measures on Corrupt cells. The three Corrupt cells (K-Cor, R-Cor, H-Cor) plant an incorrect answer in the variant prompt. For these cells we report two complementary measures. *Confidence sensitivity* is the paired gold-answer likelihood change already defined in Eq. 1, restricted to Corrupt cells. A negative value means that the Corrupt variant lowers the model's confidence in the gold answer relative to the original prompt. *Decision margin* measures whether the gold answer still beats the

Original	Variant AUG (Augment)	Variant ABL (Ablate)	Variant COR (Corrupt)
Knowledge Original	**K-Aug: Knowledge Surfacing**	**K-Abl: Paraphrase Robustness**	**K-Cor: Wrong-Claim Robustness**
In which country is Mount Everest located? Answer: Nepal	*This mountain is on the border between two regions, one being Tibet. In which country is Mount Everest located?* Change: hint added	*What country can you find Mount Everest in?* Change: question paraphrased	*Some say Mount Everest is located in India. In which country is Mount Everest located?* Change: false claim planted
Reasoning Original	**R-Aug: Scaffold Activation**	**R-Abl: Rule Grounding**	**R-Cor: Intermediate-Step Robustness**
If two times a number plus three equals fifteen, what is the number? Answer: 6	*Step 1: write the equation. Step 2: subtract 3. Step 3: divide by 2. What is the number?* Change: decomposition scaffold	*If two times a number plus three equals what value, and the number is [MASK]?* Change: rule masked	*... then the number is 8. Is this correct? What is the actual number?* Change: wrong intermediate planted
Hybrid Original	**H-Aug: Bridge-Fact Integration**	**H-Abl: Retrieval-Block Compensation**	**H-Cor: Wrong-Bridge Robustness**
What is the nationality of the author who wrote The Great Gatsby? Answer: American	*The Great Gatsby was written by F. Scott Fitzgerald. What is the nationality of the author?* Change: bridge fact provided	*A famous novel about the [MASK] Dream ... written by an author from a country that borders Canada and Mexico. Nationality?* Change: bridge entity removed	*The Great Gatsby was written by Ernest Hemingway. What is the nationality of the author?* Change: false bridge planted

Table 2: Examples of original questions and three perturbation variants across knowledge-dominant, reasoning-dominant, and composite knowledge–reasoning tasks.

Dataset component	Rows
Items (25 per domain)	75
Original prompt rows	150
Primary variant rows	450
Knowledge / Reasoning cells	300
Hybrid primary cells	150
Hybrid both-blocked control rows	50
Total released prompt rows	**650**

Table 3: Dataset composition. One row is one prompt string. Each primary variant prompt is paired at analysis time with its corresponding original prompt, yielding 450 paired diagnostic tests from 150 shared originals. The 50 Hybrid both-blocked rows are appendix-only controls.

planted wrong answer under the variant prompt:

$$\text{margin}_v = \log p_v(\text{g}) - \log p_v(\text{w}), \qquad (2)$$

where g denotes the gold answer and w denotes the planted wrong answer. A negative margin means that the model assigns higher log-probability to the planted wrong answer than to the gold answer; a positive margin means the gold answer is still preferred. We use both measures in §6.2 and §6.3 because they answer different diagnostic questions: confidence sensitivity measures how much the distractor changes belief in the gold answer, while decision margin measures whether the distractor changes the preferred answer. Optional probe and surface-control checks are reported in Appendix A.1; they are not used in the main behavioural claims.

6 Results

6.1 Domain-Level Aggregates Do Not Reveal Repair-Relevant Capacity Profiles

Figure 2 reports capacity profiles for the four main Qwen3 models, grouped by task domain. The main point is not a scaling law, but a diagnostic one: capacities inside the same task domain can have very different magnitudes, so a domain-level aggregate does not identify the repair-relevant bottleneck.

In Knowledge, Wrong-Claim Robustness (K-Cor) becomes the largest confidence shift on Qwen3-8B-Instruct, whereas Knowledge Surfacing (K-Aug) and Paraphrase Robustness (K-Abl) remain smaller. In Reasoning, Scaffold Activation (R-Aug) shows the clearest instruction-tuning jump, and in the Hybrid set, Wrong-Bridge Robustness (H-Cor) dominates the other Hybrid cells. These trends are visible in Figure 2; full per-model means, bootstrap intervals, and per-capacity contrasts are reported in Appendix A.4. These differences matter because they imply different repairs. A single Knowledge score cannot tell whether

Cell / measure	Diagnosis and repair
K-Cor confidence sensitivity < 0, $\text{margin}_v > 0$	Confidence drops, but the forced-choice decision still favours gold. Repair: confidence-aware or contradiction-checked decoding.
H-Cor confidence sensitivity < 0, $\text{margin}_v < 0$	Confidence drops and the forced-choice decision flips to the planted wrong answer. Repair: source verification, bridge consistency checking, or re-ranking against retrieved evidence.

Table 4: Two regimes of Corrupt-cell failure on the four main Qwen3 models. K-Cor and H-Cor share a similar domain-level symptom but call for different repairs. Confidence sensitivity is Eq. 1 restricted to Corrupt cells; margin_v is Eq. 2.

the model needs paraphrase-augmented training, contradiction-aware decoding, or better retrieval cues. The operational role of an atomic capacity is therefore to turn a broad domain-level failure into a concrete repair hypothesis.

A cross-architecture verification (Figure 6 in Appendix A.4) demonstrates the same diagnostic need, with models of similar sizes having different capacity profiles even when evaluated on the same task domain. This helps justify comparing models at a matched scale: similar-sized models may look comparable under a domain-level score while relying on different underlying capacities. We treat this as supporting evidence rather than as a systematic cross-model-family scaling study.

6.2 Corrupt Cells Split into Confidence Loss and Decision Flips

Figure 3 compares the two Corrupt measures defined in §5. Confidence sensitivity asks how much the Corrupt variant changes the gold-answer likelihood; decision robustness asks whether the gold answer still beats the planted wrong answer under the variant prompt. Both measures are log-probability-based and therefore comparable across pretrained and instruction-tuned models, which is the convention used by standard pretraining-evaluation harnesses (e.g., `lm-eval-harness`, Gao et al., 2024; HELM, Liang et al., 2023) and large-scale model reports. These measures answer different diagnostic questions, and they can disagree: a confidence drop need not flip the forced-choice decision, and a flipped decision need not correspond to the largest confidence drop.

On K-Cor, the wrong claim reduces confidence in the gold answer, but the forced-choice decision still favours it. Confidence sensitivity is most negative on Qwen3-8B-Instruct (-1.16), yet the decision margin remains positive and grows from $+0.50$ on Qwen3-1.7B-Base to $+2.94$ on Qwen3-8B-Instruct. This means that the model is affected by the planted wrong claim, but still assigns higher log-probability to the gold answer than to the planted answer under the variant prompt. R-Cor follows the same broad pattern in this measure: the confidence shifts are less directional, but the margin remains positive across the four main Qwen3 models.

For H-Cor, the confidence sensitivity is more negative, from -1.81 on Qwen3-1.7B-Base to -9.31 on Qwen3-8B-Instruct, and the decision margin is negative for all four main Qwen3 models. Since $\text{margin}_v < 0$ means that the planted wrong answer receives higher log-probability than the gold answer, H-Cor is not merely a confidence drop: it is a decision-flipping Corrupt cell under this forced-choice measure. The broader Qwen3 / Qwen3.5 sweep preserves the K-Cor / H-Cor distinction in broad form: K-Cor margins remain positive across the sweep, while H-Cor margins are consistently negative on Qwen3 and closer to zero on Qwen3.5 (Appendix A.4).

Table 4 summarises the two regimes. K-Cor and H-Cor both plant a competing wrong answer, yet a single gold-answer likelihood drop would not separate them: K-Cor reads as a confidence-sensitivity problem that the forced-choice decision absorbs, whereas H-Cor reads as a decision-flipping problem. The two regimes therefore point to different repair targets: confidence-aware or contradiction-checked decoding for K-Cor, and stronger upstream interventions such as source verification, bridge consistency checking, or re-ranking against retrieved evidence for H-Cor.

6.3 Zero-shot CoT is not uniformly beneficial on Corrupt cells

We use zero-shot CoT as a simple test case for whether a common prompt-level intervention behaves uniformly across atomic capacities. Rather than averaging a prompt intervention over a whole task domain, we evaluate the standard zero-shot CoT prefix "`Let's think step by step.`" (Kojima et al., 2022) separately on each Corrupt capacity. For each original–variant pair, the prefix is prepended to both prompts, and the gold answer is scored at the same appended-token positions as in the no-CoT condition. Figure 4 reports the decision

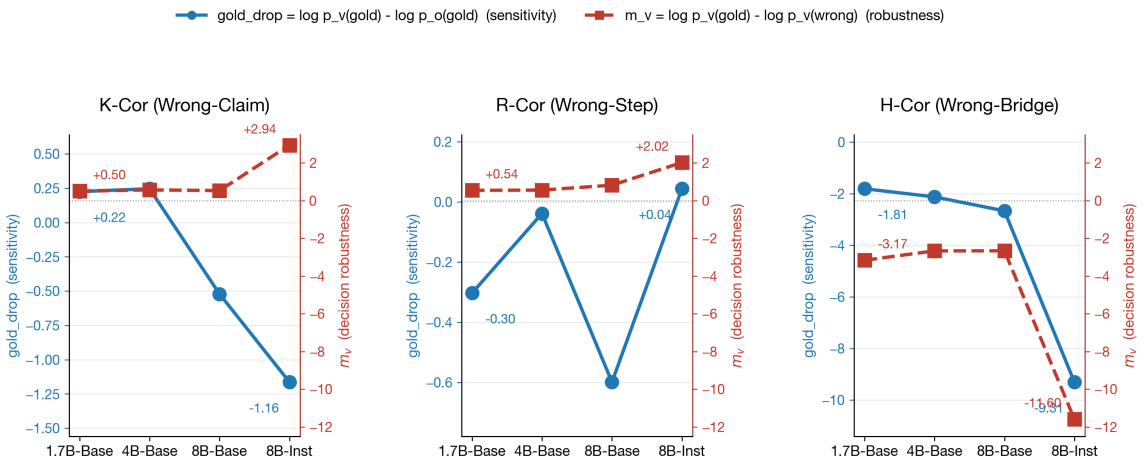

Figure 3: Corrupt cells, four main Qwen3 models. Blue: confidence sensitivity; red: decision margin m_v (shared axis). K-Cor and R-Cor stay near zero; H-Cor drops an order of magnitude.

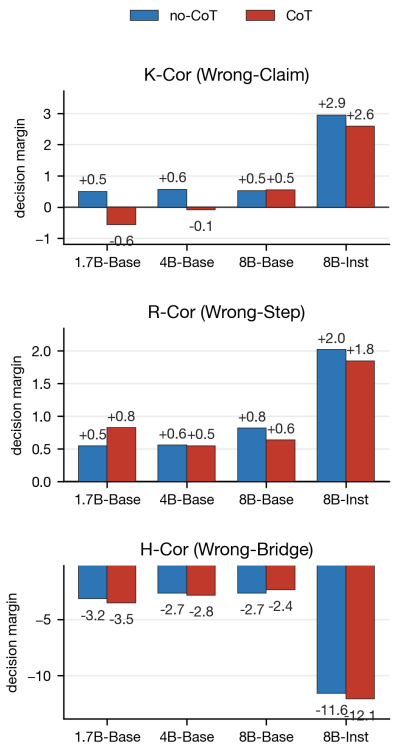

Figure 4: Paired no-CoT and CoT decision margins on Corrupt cells.

margin with and without the prefix.

The effect is not uniformly positive. On K-Cor, CoT reduces the decision margin on three of the four main Qwen3 models. For Qwen3-1.7B-Base and Qwen3-4B-Base, the margin crosses zero: it changes from positive to negative, meaning that the planted wrong answer becomes more likely than the gold answer under the CoT-prefixed variant. On R-Cor, the margin remains positive and the changes are small relative to K-Cor and H-Cor. On H-Cor, the no-CoT margin is already negative, and CoT makes it more negative for three of the four models, including Qwen3-8B-Instruct changing from -11.60 to -12.10.

The key point is diagnostic rather than mechanistic. The same prompt-level intervention can change different Corrupt capacities in different directions: it can flip the decision on K-Cor for smaller models, leave R-Cor largely stable, and deepen an already negative margin on H-Cor. This supports the broader claim that interventions, like failures, should be evaluated at the capacity level rather than only through a domain-level average.

7 Conclusion

In this work, we argued that the diagnostic unit for language-model failures should be the atomic capacity rather than the task domain. Domain-level scores hide repair-relevant capacity profiles; Corrupt cells require both confidence and decision readouts; and a standard prompt-level intervention can help one capacity while hurting another. Our nine-capacity instantiation shows that the same task-domain label can mask within-model bottlenecks and cross-architecture profiles at similar scale. The benefit is diagnostic: an atomic-capacity profile turns a broad benchmark failure into a concrete repair hypothesis. This shifts evaluation from ranking models by broad scores to diagnosing the failure modes that should be repaired. The nine-cell grid is one controlled instantiation; richer future grids should preserve the principle of mapping failures to the capacity they expose and the intervention.

Limitations

This paper is a diagnostic study rather than a full intervention or causal-mechanism study. The nine atomic capacities are one controlled instantiation of the framework, evaluated mainly on a coherent Qwen3 model series with synthetic paired tests. Although we include validation, surface controls, and cross-architecture sanity checks, the results should be interpreted as evidence for the proposed behavioural diagnostic unit rather than as an exhaustive taxonomy of failures or a claim about internal causal mechanisms.

The dataset is intentionally controlled: Knowledge and Hybrid items are grounded in Wikidata-style relations, and Reasoning items are generated from rule-based templates. This makes paired perturbations clean, but does not cover the full diversity of natural user queries, domain-specific knowledge, multilingual settings, or long-context retrieval environments. The same methodology could be extended to biomedical, legal, coding, or enterprise retrieval tasks.

We also evaluate repair implications rather than implementing complete repair pipelines. Capacity profiles suggest interventions such as paraphrase training, contradiction-aware decoding, decomposition scaffolds, bridge retrieval, or source verification, but optimising those interventions end to end is left for future work.

Ethics statement

The dataset is synthetically generated from public knowledge graphs; no personal data is involved. Probing studies of frozen models do not modify model behaviour. We do not see realistic dual-use concerns specific to this work beyond those already attendant to studying LLM internals.

References

Anthropic. 2025. Claude sonnet 4.5 system card. https://www.anthropic.com/claude-sonnet-4-5-system-card.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Yew Ken Chia, Guizhen Chen, Luu Anh Tuan, Soujanya Poria, and Lidong Bing. 2023. Contrastive chain-of-thought prompting. *arXiv preprint arXiv:2311.09277*.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, and 1 others. 2022. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61.

DeepSeek-AI Team. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yi Dong, Ronghui Mu, Yanghao Zhang, Siqi Sun, Tianle Zhang, Changshun Wu, Gaojie Jin, Yi Qi, Jinwei Hu, Jie Meng, and 1 others. 2025. Safeguarding large language models: A survey. *Artificial intelligence review*, 58(12):382.

Junbo Fu, Guoshuai Zhao, Yimin Deng, Yunqi Mi, and Xueming Qian. 2024. Learning to paraphrase for alignment with LLM preference. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2394–2407, Miami, Florida, USA. Association for Computational Linguistics.

Deep Ganguli, Danny Hernandez, Liane Lovitt, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, Nelson Elhage, and 1 others. 2022. Predictability and surprise in large generative models. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pages 1747–1764.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. A framework for few-shot language model evaluation.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, and 7 others. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8018–8025.

Mingyu Jin, Weidi Luo, Sitao Cheng, Xinyi Wang, Wenyue Hua, Ruixiang Tang, William Yang Wang, and Yongfeng Zhang. 2025. Disentangling memory and reasoning ability in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1681–1701, Vienna, Austria. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, and 31 others. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Aofan Liu and Jingxiang Meng. 2026. Paraphrase-induced output-mode collapse: When llms break character under semantically equivalent inputs. *arXiv preprint arXiv:2605.04665*.

Llama Team. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nicholas Lourie, Michael Y. Hu, and Kyunghyun Cho. 2025. Scaling laws are unreliable for downstream tasks: A reality check. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16167–16180, Suzhou, China. Association for Computational Linguistics.

Felipe Maia Polo, Seamus Somerstep, Leshem Choshen, Yuekai Sun, and Mikhail Yurochkin. 2026. Sloth: scaling laws for llm skills to predict multi-benchmark performance across families. *Advances in Neural Information Processing Systems*, 38:39511–39556.

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, Florence, Italy. Association for Computational Linguistics.

Philipp Mondorf and Barbara Plank. 2024. Beyond accuracy: evaluating the reasoning behavior of large language models–a survey. *arXiv preprint arXiv:2404.01869*.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Olmo Team. 2025. Olmo 3. *arXiv preprint arXiv:2512.13961*.

Jie Qiu, Quanrong Fang, and Wenhao Kang. 2025. Towards controllable and explainable text generation via causal intervention in llms. *Electronics*, 14(16):3279.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Qwen Team. 2026. Qwen3.5-omni technical report. *arXiv preprint arXiv:2604.15804*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are emergent abilities of large language models a mirage? *Advances in neural information processing systems*, 36:55565–55581.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Juncheng Wu, Sheng Liu, Haoqin Tu, Hang Yu, Xiaoke Huang, James Zou, Cihang Xie, and Yuyin Zhou. 2025. Knowledge or reasoning? a close look at how llms think across domains. *arXiv preprint arXiv:2506.02126*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. 2024. SafeDecoding: Defending against jailbreak attacks via safety-aware decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5587–5605, Bangkok, Thailand. Association for Computational Linguistics.

A Appendix

A.1 Probe and surface-control details

As a supporting check we ask whether the capacity distinctions used in the body are linearly decodable from the model's frozen hidden states. Per (capacity, CoT, pool) cell we fit a logistic-regression probe on a fixed-layer hidden state under family-level 5-fold cross-validation, sweeping all transformer layers and reporting the best-layer balanced accuracy. The best per-cell balanced accuracies on Qwen3-8B fall in the 0.75–0.82 range (Table 5); a rank-16 LoRA read-out (Hu et al., 2022) fit on the same cells coincides with the linear numbers within rounding (e.g., R-Abl CoT/last $0.78/0.78$, H-Aug CoT/last $0.75/0.80$, R-Abl CoT/mean $0.82/0.77$). We use the linear probe rather than the LoRA read-out because it pins down a layer rather than learning a small adapter. A linear probe shows that the relevant variable is linearly decodable from the hidden state; it does not establish that the model uses that variable in its forward computation (Belinkov, 2022).

Cell	CoT/Pool	n	Linear bal. acc.
R-Abl	CoT/mean	23	0.82
R-Aug	CoT/mean	24	0.82
R-Abl	CoT/last	23	0.78
K-Aug	no-CoT/last	23	0.78
R-Abl	no-CoT/last	22	0.77
R-Cor	no-CoT/last	24	0.75
H-Aug	CoT/last	23	0.75
H-Aug	CoT/mean	23	0.75

Table 5: Top per-cell capacity-decoding cells (linear probe, balanced accuracy) under the $\Delta \log p$ judge on Qwen3-8B. K-Cor is omitted because the no-CoT column has only one positive sample.

Surface controls. At the family level, four prompt-artefact baselines on the three-way Knowledge / Reasoning / Hybrid classification locate the surface ceiling: length-only logistic regression reaches 0.864 balanced accuracy, TF-IDF n-gram regression 0.914, random-label gives 0.229, random-feature gives 0.247; the hidden-state probe reaches 0.973–0.987. The 0.914 TF-IDF number means the family read-out is mostly surface and is treated as a sanity check.

For the finer per-capacity question we run within-family three-way classification (K-Aug / K-Abl / K-Cor, R-Aug / R-Abl / R-Cor, H-Aug / H-Abl / H-Cor) using the same item-level five-fold split. Three surface baselines (random-label, length-only, TF-IDF word 1–2-gram) and a template-masked TF-IDF variant (assay markers such as "Some say", "Step 1", "[MASK]" replaced by generic tokens) are compared against the best per-capacity hidden-state probe accuracy on Qwen3-8B from Table 5. The within-family TF-IDF baseline reaches 0.88–0.98 balanced accuracy and exceeds the hidden-state probe in every family (Knowledge 0.96 vs. 0.78, Reasoning 0.98 vs. 0.82, Hybrid 0.88 vs. 0.75; Table 6). Template-masking the surface markers removes only 1–4 percentage points. The per-capacity probe is therefore largely explainable by prompt surface features and does not, by itself, establish template-independent representations. We treat it as surface-sensitive supporting evidence rather than as a main finding. The behavioural $\Delta \log p$ and margin analyses in §6 remain the primary claim and do not depend on probing.

A.2 Dataset construction pipeline

A.3 Per-cell accuracy on the four main Qwen3 models

The body figures use $|\overline{\Delta \log p}|$ per token and the forced-choice margin margin_v as the decision-robustness signal, both of which are log-probability-based and directly comparable across pretrained and instruction-tuned models. This follows the standard pretraining-evaluation convention (e.g., `lm-eval-harness` (Gao et al., 2024), HELM (Liang et al., 2023), the original GPT-3 / Chinchilla / PaLM evaluations), which scores models by per-option or gold-token log-probabilities rather than by free-form generation. For completeness we also report free-form accuracy: per (model, cell) we record whether the gold answer appears as a case-folded substring in the model's 32-token greedy generation, on both the original and variant prompts, and report a_o, a_v, and $\Delta a = a_v - a_o$ in Table 7. The string-match judge captures Qwen3-8B-Instruct outputs unreliably because its chat template wraps generations in a longer format, yielding $a_o = 0$ across all cells; we retain this row for completeness but the comparable cross-model signals remain $|\overline{\Delta \log p}|$ and margin_v.

A.4 Full 17-model sweep

Table 8 reports the Corrupt cells across the full Qwen3 / Qwen3.5 sweep (17 models: the 8 Qwen3 base / instruct sizes from 0.6B to 8B, and Qwen3.5 sizes from 0.8B to 35B). K-Cor margin_v stays positive across the entire sweep; H-Cor margin_v on

Family	Rand.	Length	TF-IDF	TF-IDF$_{\text{masked}}$	Hidden
Knowledge	0.28	0.71	**0.96**	0.92	0.78
Reasoning	0.39	0.75	**0.98**	0.97	0.82
Hybrid	0.35	0.73	**0.88**	0.87	0.75

Table 6: Per-capacity surface controls. Mean balanced accuracy over five GroupKFold folds (item-level split). The hidden-state column reports the best per-capacity probe cell on Qwen3-8B from Table 5 as an approximate reference; the original probe is a per-cell setup, not a within-family 3-way setup, so the comparison is indicative rather than exact.

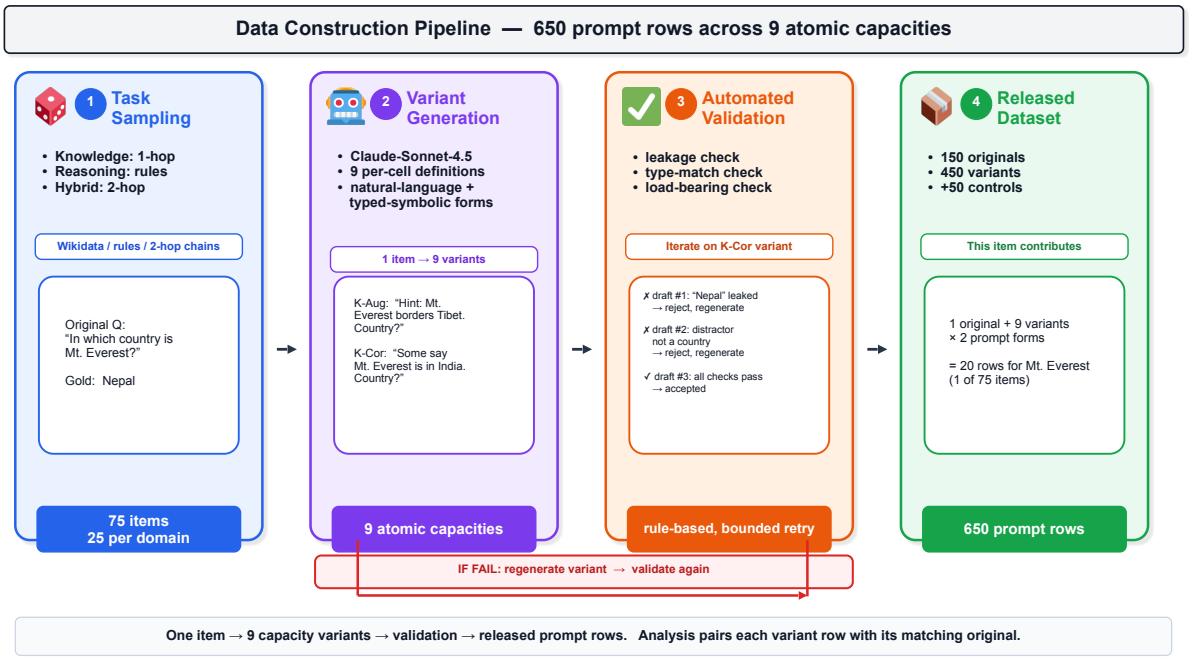

Figure 5: Dataset construction pipeline. We sample Knowledge, Reasoning, and Hybrid items; generate capacity-specific variants with Claude-Sonnet-4.5; apply rule-based validators for gold leakage, Corrupt-variant type matching, and Ablate-variant load-bearing checks; and release 650 prompt records.

	1.7B-Base			4B-Base			8B-Base			8B-Inst		
Cell	a_o	a_v	Δa	a_o	a_v	Δa	a_o	a_v	Δa	a_o	a_v	Δa
K-Aug	0.00	0.24	+0.24	0.00	0.16	+0.16	1.00	0.96	−0.04	0.00	0.04	+0.04
K-Abl	0.00	0.08	+0.08	0.00	0.16	+0.16	1.00	0.92	−0.08	0.00	0.00	+0.00
K-Cor	0.00	0.16	+0.16	0.00	0.88	+0.88	1.00	0.96	−0.04	0.00	0.04	+0.04
R-Aug	0.20	0.00	−0.20	0.04	0.08	+0.04	0.40	0.28	−0.12	0.04	0.00	−0.04
R-Abl	0.20	0.04	−0.16	0.04	0.12	+0.08	0.40	0.12	−0.28	0.04	0.00	−0.04
R-Cor	0.20	0.12	−0.08	0.04	0.20	+0.16	0.40	0.20	−0.20	0.04	0.00	−0.04
H-Aug	0.00	0.24	+0.24	0.08	0.44	+0.36	0.88	0.84	−0.04	0.00	0.04	+0.04
H-Abl	0.00	0.04	+0.04	0.08	0.20	+0.12	0.88	0.60	−0.28	0.00	0.04	+0.04
H-Cor	0.00	0.00	+0.00	0.08	0.24	+0.16	0.88	0.60	−0.28	0.00	0.00	+0.00

Table 7: Free-form accuracy on the four main Qwen3 models, no-CoT. a_o is accuracy on the original prompt, a_v on the variant; $\Delta a = a_v - a_o$. Correctness is determined by case-folded substring match against the gold answer in a 32-token greedy generation. The string match captures Qwen3-8B-Instruct outputs unreliably because the chat template wraps generations in a longer format, so its Δa values are not directly comparable to the base models'.

Qwen3 is consistently negative and most negative at the larger sizes, while on Qwen3.5 it is closer to zero or slightly positive. The two families confound the simple base-instruct contrast that the body relies on, which is why the body restricts itself to a single coherent four-model series. Four Qwen3.5 entries did not run the H-Cor assay and are shown as dashes.

Model	K-Cor drop	K-Cor margin$_v$	R-Cor drop	R-Cor margin$_v$	H-Cor drop	H-Cor margin$_v$
Qwen3-0.6B-Base	$+0.87$	$+0.35$	$+0.32$	$+0.70$	-3.15	-4.65
Qwen3-0.6B	-0.18	$+0.37$	-0.68	$+5.21$	-5.79	-7.14
Qwen3-1.7B-Base	$+0.22$	$+0.50$	-0.30	$+0.54$	-1.81	-3.17
Qwen3-1.7B	-0.29	$+0.98$	-1.48	$+5.90$	-7.30	-9.00
Qwen3-4B-Base	$+0.25$	$+0.56$	-0.04	$+0.56$	-2.13	-2.67
Qwen3-4B	-0.46	$+1.42$	-0.65	$+5.44$	-5.94	-8.84
Qwen3-8B-Base	-0.52	$+0.53$	-0.60	$+0.82$	-2.67	-2.66
Qwen3-8B	-1.16	$+2.94$	$+0.04$	$+2.02$	-9.31	-11.60
Qwen3.5-0.8B	-0.13	$+1.53$	$+0.68$	$+1.08$	$+1.31$	$+0.32$
Qwen3.5-2B-Base	-0.57	$+1.32$	-0.81	$+0.71$	–	–
Qwen3.5-2B	-0.19	$+1.66$	-0.23	$+1.85$	–	–
Qwen3.5-4B-Base	-0.11	$+1.37$	-0.06	$+2.50$	$+1.46$	$+0.71$
Qwen3.5-4B	$+0.14$	$+1.67$	-0.88	$+2.20$	$+0.64$	$+0.07$
Qwen3.5-9B-Base	-0.37	$+1.77$	$+0.23$	$+2.30$	$+0.64$	-0.30
Qwen3.5-9B	-0.06	$+2.32$	-0.52	$+2.45$	$+1.02$	$+0.09$
Qwen3.5-35B-A3B-Base	-0.23	$+1.76$	$+0.00$	$+2.22$	–	–
Qwen3.5-35B-A3B	$+0.13$	$+2.29$	-0.21	$+2.13$	–	–

Table 8: Corrupt cells across the full 17-model Qwen3 / Qwen3.5 sweep. drop $=$ gold_drop $= \log p_v(\text{gold}) - \log p_o(\text{gold})$ (confidence sensitivity); margin$_v = \log p_v(\text{gold}) - \log p_v(\text{wrong})$ (decision robustness). Dashes mark Qwen3.5 entries that did not run the H-Cor assay. Means over 25 paired items per cell, natural-language surface form.

A.5 Item construction

An *item* is the underlying source from which an original–variant assay pair is generated; it carries the base question, the gold answer, and the support facts that downstream variants are conditioned on. We construct 75 items in total, 25 per task domain. Knowledge items are 1-hop Wikidata relations; Reasoning items are rule-based arithmetic / symbolic templates instantiated with sampled numerals; Hybrid items are 2-hop compositions in which a bridge entity links a Wikidata fact to a question about a property of that entity. Per-domain counts are summarised in Table 9.

htbp

Task domain	Items
Knowledge (KB)	25
Reasoning (RB)	25
Hybrid	25
Total	75

Table 9: Per-domain item counts. Each item is the underlying source; original and variant prompts are generated from it and released as separate rows.

Worked example. Table 10 shows how a single Knowledge item (`kb_005`: *Mount Everest – country – Nepal*) expands into one original prompt and three capacity-specific variants. Each item in the other two domains follows the same expansion pattern; the only difference is the source structure (arithmetic template for Reasoning; two-hop bridge for Hybrid).

Each released row in the JSONL is a single prompt string with fields `family_id`, `task_family`, `variant`, `mode`, `question`, `gold_answer`, and a variant-specific `metadata` object. The `family_id` identifies the item (e.g. `kb_001`); the `mode` field is `natural` or `symbolic`. The original prompt is stored as a separate row (`variant = original`); it is not embedded inside the variant rows. At analysis time a primary

Role	Content
Item	`kb_005`: Mount Everest – country – Nepal
Original	*In which country is Mount Everest located?* Gold: Nepal
K-Aug	*This mountain is on the border between two regions, one being Tibet. In which country is Mount Everest located?*
K-Abl	*What country can you find Mount Everest in?*
K-Cor	*Some say Mount Everest is located in India. In which country is Mount Everest located?*

Table 10: Worked example: one Knowledge item (`kb_005`) and its three capacity-specific variants. The gold answer (Nepal) is shared across all four prompts; each variant perturbs the original in exactly one way.

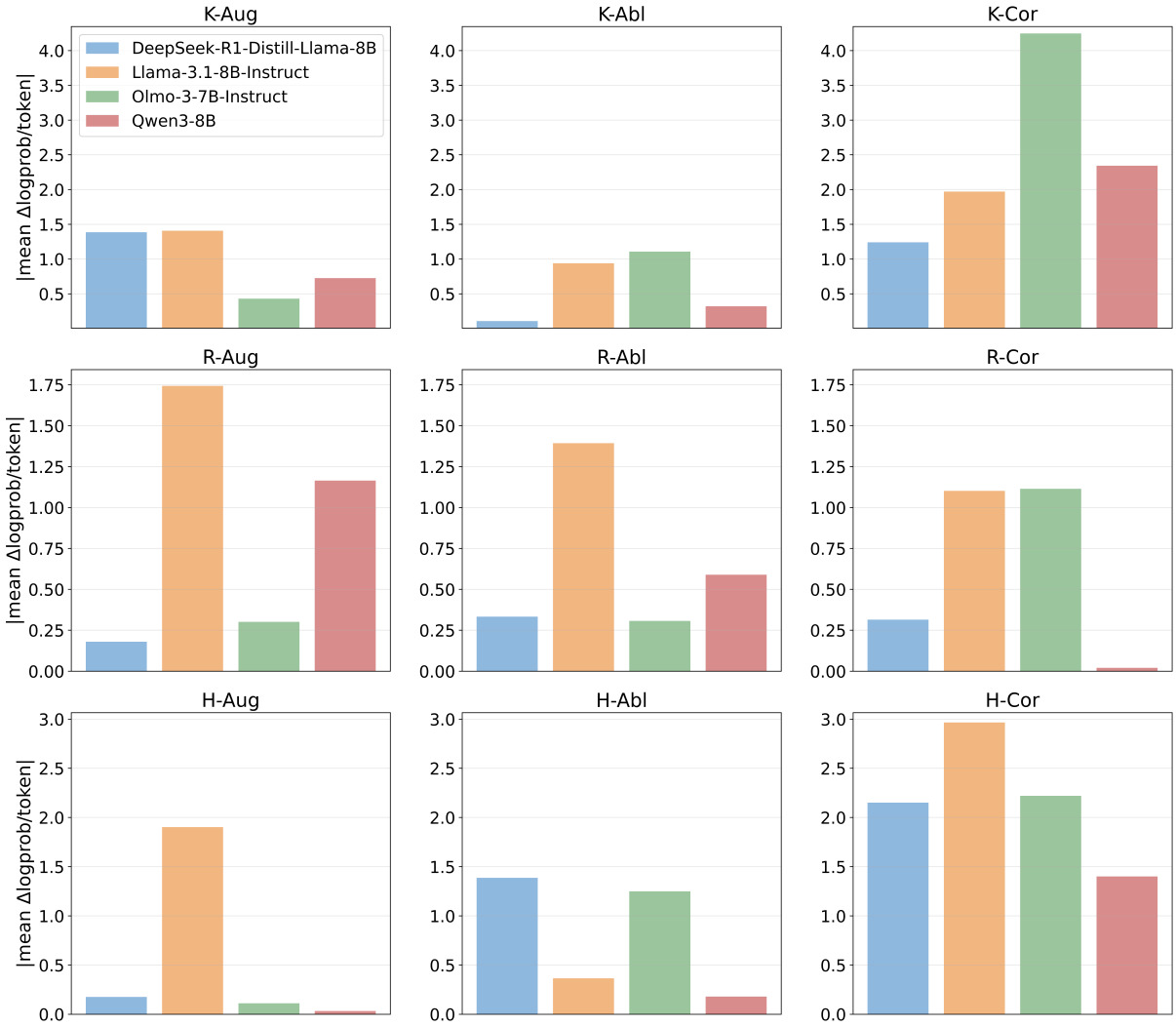

Figure 6: Cross-architecture sanity check at the 7–8B scale: atomic-capacity profiles on DeepSeek-R1-Distill-Llama-8B (DeepSeek-AI Team, 2025), Llama-3.1-8B-Instruct (Llama Team, 2024), Olmo-3-7B-Instruct (Olmo Team, 2025), and Qwen3-8B (Qwen Team, 2025). Bars are raw mean $|\overline{\Delta \log p}|$ per token (variant minus original). Same family verdict, different within-family profile. We frame this as a sanity check; systematic cross-architecture replication remains open.

variant row is paired with the original row that shares its `family_id` (the body uses the `natural` mode), which is how the 225 paired diagnostic assays summarised in Table 3 are obtained from 75 shared originals. The Knowledge and Reasoning domains contribute three primary cells each; the Hybrid domain contributes three primary cells plus the appendix-only `both_blocked` control, so each Hybrid item yields more rows than each KB/RB item (ten vs. eight after counting both surface modes). The typed-symbolic surface mode is produced by a deterministic symbolic rewriter applied after generation, so each (item, cell) pair yields one natural-language row and one typed-symbolic row.

A.6 Prompt generation details

Generator model. All variants are generated by Claude-Sonnet-4.5 (Anthropic, 2025) via the Anthropic API, model identifier `claude-sonnet-4-5-20250929`, with deterministic sampling. The original prompt is not generated by Claude: it is copied from the item's `base_question`. Only the variant prompt is generated, and only the variant's question text and a small variant-specific metadata object are obtained from the model.

System prompt. The generator runs under a fixed system prompt that constrains the output format to a single JSON object with no surrounding text:

```
You are a dataset designer that outputs
ONLY a JSON object. No markdown, no code
fences, no commentary, no explanation.
Your entire response MUST be a single
JSON object.
```

Shared user-prompt scaffold. All eight non–Wrong-Bridge cells share a single user-prompt scaffold parametrised by the variant type, the per-cell definition, and the item fields:

```
Generate a '{variant_type}' variant for
the base item below.
Variant definition: {definition}
The output JSON object MUST have exactly
these keys:
- question: string, the variant question
text (must NOT literally contain the
gold_answer)
- metadata: object, variant-specific
metadata (e.g., wrong_claim,
injected_premise, hint_text)
Base item:
- family_id: {family_id}
- task_family: {task_family}
- base_question: {base_question}
- gold_answer: {gold_answer}
- support_facts: {support_facts}
Output the JSON object now.
```

Per-cell definitions. The nine `{definition}` strings substituted into the scaffold are reproduced verbatim from the released code:

- **K-Aug / hint**: "Give answer-free cue to help retrieval. Example: 'This country is in Western Europe. What is the capital of France?' "
- **K-Abl / paraphrase**: "Rephrase question, same meaning. Example: 'Which city serves as France's seat of government?' "
- **K-Cor / wrongclaim**: "Inject wrong claim to test memory override. Example: 'Some say France's capital is Lyon. What is the capital of France?' "
- **R-Aug / scaffold**: "Give decomposition steps without answer. Example: 'Step 1: Identify equation. Step 2: Isolate x. What is x?' "
- **R-Abl / rule_removal**: "Remove key rule/number. Example: 'If x + 5 = ?, what is x?' (removed 12)"
- **R-Cor / wrong_intermediate**: "Give wrong intermediate step. Example: 'If x + 5 = 12, then x = 17. Is this correct? What is x?' "
- **H-Aug / explicit_fact**: "Give bridge fact explicitly. Example: 'Eiffel Tower is in France. What is the capital of France?' "
- **H-Abl / retrieval_blocked**: "Block retrieval, test reasoning rescue. Example: 'A famous iron tower is in a Western European country known for wine. What is the capital?' "
- **Hybrid both-blocked (appendix control)**: "Block both retrieval and reasoning. Example: 'What is the capital of landmark L in country C?' "

Wrong-Bridge cell. The H-Cor / Wrong-Bridge cell is generated by a separate routine rather than the shared scaffold, because the planted bridge must imply a competing answer different from the gold while leaving the gold still derivable from the question. The Wrong-Bridge prompt instructs the generator to choose a plausible-but-false bridge entity for the 2-hop composition and to verify that its implied answer differs from the gold.

Representative items, one per cell. For grounding, here is one (original, variant) pair per atomic capacity, pulled verbatim from the released JSONL. The original prompt is shared within each task domain, so we show it once per domain.

Knowledge domain (`kb_001`). *Original.* "In which country is Mount Everest located?" *Gold:* Nepal.

- **K-Aug / hint.** "This mountain is on the border between two regions, one being Tibet. In which country is Mount Everest located?"
- **K-Abl / paraphrase.** "What country can you find Mount Everest in?"
- **K-Cor / wrongclaim.** "Some say Mount Everest is located in India. In which country is Mount Everest located?"

Reasoning domain (`rb_001`). *Original.* "If two times a number plus three equals fifteen, what is the number?" *Gold:* 6.

- **R-Aug / scaffold.** "If two times a number plus three equals fifteen, what is the number? Step 1: Write the equation based on the problem statement. Step 2: Subtract 3 from both sides of the equation. Step 3: Divide both sides by 2 to solve for the number. What is the number?"
- **R-Abl / rule_removal.** "If two times a number plus three equals what value, and the number is [MASK]?"

- **R-Cor / wrong_intermediate.** "If two times a number plus three equals fifteen, then the number is 9. Is this correct? What is the actual number?"

Hybrid domain (`hybrid_001`). *Original.* "What is the nationality of the author who wrote The Great Gatsby?" *Gold:* American.

- **H-Aug / explicit_fact.** "The Great Gatsby was written by F. Scott Fitzgerald. What is the nationality of the author who wrote The Great Gatsby?"
- **H-Abl / retrieval_blocked.** "A famous novel about the [MASK] Dream, featuring a mysterious millionaire and a green light, was written by an author from a country that borders Canada and Mexico. What is the nationality of this author?"
- **H-Cor / wrong_bridge.** "The Great Gatsby was written by Ernest Hemingway. What is the nationality of the author who wrote The Great Gatsby?"

Retry loop. The shared scaffold does not itself enforce type-matched distractors or structural load-bearing; those are enforced post-hoc by the validators of Appendix A.7. When a validator blocks an item, the pipeline issues a bounded retry: a short repair clause derived from the failing validator (for example, "the wrong claim must be the same type as the gold answer") is appended to the user prompt, and generation is re-run. We use at most three retries; items that still fail are dropped from that (item, cell) slot rather than weakened. Surface forms (natural-language vs. typed-symbolic) are produced by a deterministic symbolic rewriter applied to the generated natural-language item, so each (item, cell) pair yields two surface variants from a single Claude generation call.

A.7 Validation and correctness judgments

All validators are rule-based string / regex / heuristic checks. No LLM-as-judge is used anywhere in the construction pipeline.

Gold-leakage (LeakageValidator). A token-aware regex `\b{gold}\b` rejects any variant whose question literally contains the gold answer. An additional regex over scaffold and wrong-intermediate variants rejects "therefore / so / thus / final answer:" followed by the gold answer, to catch indirect leakage in inserted reasoning steps. On the typed-symbolic surface form we additionally reject items in which any entity from the item's underlying structure appears verbatim. Items that fail the direct or indirect rule trigger a regeneration; metadata-level leakage (gold appearing only inside an inserted metadata field) raises a warning rather than blocking.

Type-matched distractors (ConfoundValidator). For Wrong-Claim items (K-Cor) only, the planted wrong claim and the gold answer are passed through a heuristic type classifier that returns one of {`bool`, `number`, `entity`} from a small regex / membership test. Type mismatches block the item and trigger a regeneration with a targeted "same type as gold" clause added to the prompt.

Structural removals (SurfaceValidator). For the Ablate cells where a clause is removed from the prompt (Rule-Removal, Retrieval-Block-Compensation, the appendix-only both-blocked control), an additional check rejects removals that are merely decorative prepended phrasing rather than structurally load-bearing. The check uses a small set of decorative-phrase patterns and an edit-distance comparison against the base question, and is the only "perturbation does what it claims" guard in the pipeline.

What we do *not* validate. We do not run an independent semantic answer-preservation judge: the eight assays whose questions are unchanged inherit the item's gold answer by copy, and answer preservation is enforced only by the no-leak constraint and the structural-removal check. The audit pass below provides a human quality signal on a 90-row stratified subsample, not on the full 650-row release.

Correctness judgments. The diagnostic criterion is asymmetric by perturbation (Table 1). For Augment cells the diagnostic is $\neg o \wedge v$ (the model fails on the original and succeeds when the missing cue is provided); for Ablate and Corrupt cells it is $o \wedge \neg v$ (the model succeeds on the original and fails when the cue is removed or a competitor is planted). Per-item correctness is determined from the model's free-form generation (32 new greedy-decoded tokens) by a soft string match against the gold answer — case-folded substring containment or prefix match. This is not an MCQ-index judgment and is not based on a log-probability compar-

ison against competitors; the forced-choice margin $\text{margin}_v = \log p_v(\text{gold}) - \log p_v(\text{wrong})$ used in §6.2 is computed only for the three Corrupt cells (K-Cor, R-Cor, H-Cor), reported separately as a decision-robustness signal, and is not used as the general correctness criterion.

Z-normalisation (label derivation only). A z-score path exists in the released code for deriving discrete per-family atomic-capacity labels: per (capacity, CoT-state) within one model and one surface mode, $\Delta \log p$ values are pooled across the 25 items and z-scored (no additive epsilon; degenerate groups with $\sigma < 10^{-9}$ are dropped rather than divided). This z-score is used only for label derivation in downstream analyses, not for any number reported in the body figures or tables, all of which use raw mean $\Delta \log p$ per token with bootstrap CIs.

Audit template. We release a manual audit template of 90 rows (10 items per capacity, stratified by item with a fixed seed) as supplementary material. Each row records the original and variant prompts, the assay label, and three binary annotation columns: (Q1) whether the gold answer is correct under the original prompt, (Q2) whether the gold remains the (Aug/Abl) intended answer or the (Cor) still-correct answer that the planted distractor is trying to displace, and (Q3) whether the variant exemplifies its declared perturbation type (Augment adds a cue, Ablate removes one, Corrupt plants a wrong claim/step/bridge).

Inter-annotator agreement. Two PhD-level annotators independently labelled all 90 variant rows (and the 30 paired originals for Q1). Cohen's κ on Q1, Q2, and Q3 is 0.89, 0.95, and 0.72, with raw agreement 94.4%, 98.9%, and 87.8%. By the Landis–Koch convention, Q1 and Q2 are "almost perfect" and Q3 is "substantial." Q3 attracts the most disagreement because annotators disagreed on whether some Ablate variants count as having genuinely removed a load-bearing cue.

Per-capacity pass rates. Per-capacity pass rates (Table 11) are reported under the strict "both annotators say yes" criterion. Gold preservation under the variant (Q2) is high overall (88%) and $\geq 80\%$ in 8 of 9 cells; the one outlier is R-Abl (30%), where annotators frequently judged the masked rule ablation to leave the question still answerable from the remaining template. Perturbation-type fidelity (Q3) is high on Augment and Corrupt cells ($\geq 60\%$) but low on K-Abl (10%) and R-Abl (0%), where annotators repeatedly flagged the cue removal as too weak to count as a true ablate. We treat this as an honest limit on the construct validity of K-Abl and R-Abl rather than as a positive claim about them, and we discuss it in §7.

htbp

Cell	n	Q2 pass (%)	Q3 pass (%)
K-Aug	10	100	80
K-Abl	10	100	10
K-Cor	10	80	100
R-Aug	10	100	80
R-Abl	10	30	0
R-Cor	10	100	70
H-Aug	10	100	80
H-Abl	10	90	100
H-Cor	10	90	60
All	90	88	64

Table 11: Per-capacity audit pass rates under the strict "both annotators say yes" criterion. Q2: gold answer remains correct under the variant prompt. Q3: variant exemplifies its declared perturbation type. K-Abl and R-Abl have low Q3 because annotators flagged the cue removal as too weak in many items.

	Model	K-Cor		R-Cor		H-Cor	
		no-CoT	CoT	no-CoT	CoT	no-CoT	CoT
htbp	Qwen3-1.7B-Base	$+0.50$	-0.56	$+0.54$	$+0.82$	-3.17	-3.52
	Qwen3-4B-Base	$+0.56$	-0.08	$+0.56$	$+0.55$	-2.67	-2.84
	Qwen3-8B-Base	$+0.53$	$+0.55$	$+0.82$	$+0.64$	-2.66	-2.36
	Qwen3-8B (instruct)	$+2.94$	$+2.59$	$+2.02$	$+1.85$	-11.60	-12.10

Table 12: Per-cell paired margin$_v$ under no-CoT and zero-shot CoT, four main Qwen3 models.

	Model	K-Cor		R-Cor		H-Cor	
		no-CoT	CoT	no-CoT	CoT	no-CoT	CoT
htbp	Qwen3-1.7B-Base	0.40	0.55	0.43	0.49	3.41	3.02
	Qwen3-4B-Base	0.86	0.80	0.34	0.43	2.69	3.01
	Qwen3-8B-Base	0.73	0.77	0.61	0.42	2.73	2.13
	Qwen3-8B (instruct)	1.69	1.31	1.01	1.50	9.56	9.64

Table 13: Per-cell $|\overline{\Delta \log p}|$ per token under no-CoT and zero-shot CoT on the same Corrupt cells; the sensitivity signal $|\overline{\Delta \log p}|$ does not move in a consistent direction.

A.8 CoT diff: per-cell paired numbers

Table 12 reports the per-cell paired margin$_v$ values underlying Figure 4, and Table 13 reports the complementary $|\overline{\Delta \log p}|$ per token. The sensitivity signal $|\overline{\Delta \log p}|$ does not move in a consistent direction across cells under the prefix, which is why the body finding (§6.3) is centred on margin$_v$.

A.9 Hybrid double-block control

The Hybrid-both_blocked variant – both the bridge entity and the rule clause removed – is included as a double-block control rather than a primary capacity assay. It behaves as a lower envelope on Retrieval-Block Compensation: by construction it removes both the retrieval cue and the rule, so a small drop relative to Retrieval-Block Compensation indicates the rule was already covering the bridge gap.