REPRESENTATION SENSITIVITY IN SYMBOLIC EXECUTION BY LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models can execute symbolic procedures yet remain sensitive to how those procedures are represented. We study this gap using 256 deterministic table-execution programs with exact gold states. In the primary intervention, programs, table values, initial states, and decoding settings are fixed; only the names of row and column roles change. On gpt-oss-120b, replacing overlapping A/B role labels with disjoint P/Q labels raises accuracy from 82.4–86.7% to 98.0–98.8% across three sampling seeds. The direction persists under reversed condition order and across tested Qwen configurations. A subsequent 2×2 experiment crosses A/B versus W/X role labels with A–D versus W–Z value alphabets, preserving computations under a consistent symbol renaming. Across two sampling seeds, the preferred label pair reverses with the value alphabet: overlapping labels produce more errors within both alphabets. Mean within-alphabet overlap penalties are 7.81 and 10.94 percentage points, with paired bootstrap 95% intervals of [3.91, 11.72] and [7.03, 14.84]. All primary and crossed responses terminate naturally and are syntactically valid. A selected 12-root diagnostic cohort localizes the first visible error in ten failing traces to table-cell lookups. These results establish representation sensitivity and a replicated label–alphabet interaction in the tested setting, while leaving internal mechanisms unresolved. They motivate evaluating representation robustness alongside nominal symbolic-execution accuracy.

1 INTRODUCTION

Large language models (LLMs) can solve many tasks that require multi-step reasoning, planning, and symbolic manipulation, especially when prompted to produce intermediate reasoning traces (Wei et al., 2022). At the same time, benchmark results are known to move under apparently incidental changes to prompt wording, formatting, option order, or symbolic presentation (Sclar et al., 2024; Mizrahi et al., 2024; Pezeshkpour and Hruschka, 2024; He et al., 2024). This creates a practical reliability question that is distinct from whether a model can solve a task at all: *if two inputs denote exactly the same symbolic computation, how invariant is the model's execution to the representation used to encode that computation?*

This question is difficult to study on ordinary natural-language benchmarks. A paraphrase may subtly change pragmatics or difficulty, an evaluation heuristic may mis-score semantically equivalent outputs, and a model may fail because of search, context, or output-format constraints rather than symbolic execution itself. Recent work has highlighted both genuine prompt sensitivity and evaluation artifacts that can masquerade as sensitivity (Sclar et al., 2024; Mizrahi et al., 2024; Hua et al., 2025), while controlled symbolic generation has exposed brittleness in mathematical reasoning (Mirzadeh et al., 2025). We therefore study a deliberately narrow setting in which semantics and grading can be frozen exactly.

We introduce EXACTBENCH-RENDER, a synthetic state-execution benchmark with a deterministic oracle and a renderer-only intervention. Each root instance contains three explicit 4×4 lookup tables over symbols $\{A, B, C, D\}$, three registers, a one-bit mode, a mode-dependent mapping from two operators to the tables, and a short parsed instruction stream. The evaluated renderer pool uses a fixed mode within each program; the prompt retains both mode maps and the broader instruction contract. The model must execute the program and emit one final state. For paired conditions, every table cell, mode map, program instruction, initial state, gold state, and decoding parameter is identical.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Only the axis labels printed around the tables change. The LEGACY renderer reuses value symbols as axis-variable names (columns `B=A, ...`, rows `A=A, ...`); the neutral P/Q renderer uses disjoint role labels (columns `Q=A, ...`, rows `P=A, ...`).

Our experiments began as an exploratory failure-mining study on Qwen3.8-27B. After discovering a renderer confound, we explicitly separated the exploratory discovery chain from subsequent replication. The frozen LEGACY-versus-P/Q contrast was then evaluated across new seeds, a higher-precision Qwen configuration, and a separately developed gpt-oss-120b family. The central result is large: gpt-oss-120b produces 13.3–17.6% natural wrong answers under LEGACY but only 1.2–2.0% under P/Q across three seeds. Reversing condition order in the third seed preserves the effect. Qwen shows the same direction across three Q4 seeds and one Q8 run.

Our contributions are:

- **A controlled representation intervention.** We construct paired symbolic-execution prompts from the same semantic record and deterministic oracle; the rendered prompts differ only in the axis-role labels surrounding lookup tables.
- **Cross-seed and cross-family evidence.** The representation gap persists across frozen Qwen follow-ups and a separately developed gpt-oss-120b family, with exact paired comparisons and an order-reversal control. The errors are natural, syntactically valid completions rather than truncations or parser failures.
- **A crossed label/value follow-up.** A four-condition intervention separates label identity and value alphabet as experimental factors. The preferred labels reverse across alphabets in both seeds, with mean overlap penalties of 7.81 and 10.94 percentage points; internal mechanisms remain unresolved.
- **A bounded mechanism diagnostic.** In a 72-request exploratory trace study, all ten observed LEGACY failures first show an externally visible table-cell lookup error. We treat generated reasoning as behavioral evidence rather than a faithful record of internal computation, consistent with prior work on chain-of-thought faithfulness (Lanham et al., 2023; Lyu et al., 2023).

We show that, on a frozen symbolic execution task that the tested models nearly solve under a clean renderer, a semantics-preserving representation choice can increase observed execution error several-fold; for gpt-oss-120b, the observed LEGACY-to-P/Q error-rate ratio ranges from 7.4× to 15.0× across the three seeds.

2 EXACTBENCH-RENDER

2.1 TASK SEMANTICS

Each instance defines registers $R_0, R_1, R_2 \in \{A, B, C, D\}$, a mode $m \in \{0, 1\}$, and three total functions $T_0, T_1, T_2 : \{A, B, C, D\}^2 \to \{A, B, C, D\}$. Two operator names, `OPX` and `OPY`, are mapped to tables as a function of the current mode. The instruction set contains assignments of the form

$$R_i \leftarrow \mathrm{OPX}(R_j, R_k), \qquad R_i \leftarrow \mathrm{OPY}(R_j, R_k),$$

plus `SWITCH`, which toggles the mode immediately, and `HOLD`, which is a no-op. The prompt explicitly states that the instruction stream is already parsed and that no arithmetic, bitwise, ordering, or unstated semantics should be inferred.

A deterministic interpreter executes the program and produces the gold tuple (R_0, R_1, R_2, m). The requested answer is a single line such as

```
FINAL_STATE: R0=B,R1=C,R2=D;MODE=0.
```

We score exact semantic agreement after a strict parser. Outcomes are R (correct), W (valid but wrong), or N (invalid/unscorable); termination is separately labeled natural or by a resource limit. This separation matters because a token cap or malformed response is not counted as a reasoning error.

Evaluated pool and generation. The renderer experiments use 256 *static-control* roots: each program contains two `HOLD` instructions at positions 3 and 6 and no `SWITCH`. The prompt still defines `SWITCH` and displays both mode maps, but execution retains its initial mode throughout. Thus the renderer evidence concerns stateful register updates and lookup/alias resolution, not dynamic mode switching. Each root contains 48 table cells (three 4×4 tables), three four-valued registers, and one binary mode. Across roots there are 128 possible register–mode combinations; within a root, only the registers change.

The supplied generator shuffles each table row as a permutation of A–D, so table cells are not sampled independently. Each mode maps `OPX`/`OPY` to two distinct tables, and the two maps must differ. Program lengths are drawn from 7–9; each assignment samples an operator, destination, and two source registers. The generator first retains dynamic parent programs for which at least one of three alternative switch semantics disagrees with the oracle, then replaces both switches with holds to form matched static controls. The evaluated pool therefore inherits this parent-selection procedure rather than constituting an unfiltered sample of arbitrary programs. The supplied generator reproduces the saved dynamic and static pools exactly with generator seed 20260913. Sampling seeds reuse these same 256 roots.

Table 1: Verified composition of the 256-root renderer pool. Counts are over programs or instructions as indicated.

Quantity	Value
Programs of length 7 / 8 / 9	90 / 94 / 72
Mean / median program length	7.93 / 8
Lookup assignments per program	5–7
All instructions	2,030
Lookup assignments	1,518 (74.78%)
`HOLD` instructions	512 (25.22%)
`SWITCH` instructions	0
Initial mode 0 / 1	127 / 129 roots

2.2 Renderer intervention

The main intervention changes only the printed names of the row and column roles. Table 2 illustrates the two conditions for one table.

The semantic contract always says that the first argument selects the row and the second selects the column. Thus `A=` and `B=` in LEGACY are not values or assignments; they are axis-variable labels that happen to overlap the value alphabet. P/Q retains the compact equation-like syntax but uses neutral, non-overlapping role labels. Semantic equivalence here is formal under this stated contract; we do not claim that the two strings have equal pragmatic clarity or equal difficulty. Any difference in surface clarity induced by the labels is part of the representation variable being measured.

The overlapping `A/B` notation is intentionally minimal: it isolates role/value name reuse rather than proposing a realistic table format. In conventional software, the same surface symbol can be reused across typed fields because a parser resolves its role outside the rendered string. A relevant deployment analogue arises when structured state is serialized into text across tool or agent interfaces, potentially after abbreviation, aliasing, flattening, or token-driven compaction. The experiment therefore probes whether a semantics-preserving serialization choice can cease to be behaviorally neutral once execution is delegated to an LLM.

For every paired run we verify that `source_id`, a semantic `root_hash`, and the computed gold state match exactly across conditions. This is the basis for root-level paired analysis.

2.3 Inference and statistics

All primary runs use sampling parameters $T = 1.0$, top-$p = 0.95$, top-$k = 20$, and an 8192-token completion ceiling. Qwen3.8 is run with the explicit `low` setting of its reasoning-effort interface (Qwen Team, 2026). This setting is held fixed from discovery through replication; the study estimates renderer sensitivity at this operating point, rather than sensitivity across reasoning budgets. gpt-

Table 2: The frozen renderer intervention. Cell values, program, mode maps, initial state, and gold state are identical across the pair.

LEGACY: overlapping role/value namespace					P/Q: disjoint role/value namespace				
	B=A	B=B	B=C	B=D		Q=A	Q=B	Q=C	Q=D
A=A	C	A	D	B	P=A	C	A	D	B
A=B	B	D	C	A	P=B	B	D	C	A
A=C	A	D	B	C	P=C	A	D	B	C
A=D	C	A	D	B	P=D	C	A	D	B

Table 3: Exploratory Qwen Q4 renderer factorial, seed 17. Natural wrong rate is shown; A/COL additionally produced three non-natural/invalid completions.

Renderer	Natural W	Error rate
LEGACY A/B	20/256	7.81%
ROW/B	11/256	4.30%
A/COL	39/256	15.23%
ROW/COL	7/256	2.73%
X/Y	7/256	2.73%
P/Q	5/256	1.95%

oss-120b is served with low reasoning effort through `llama.cpp`; the model has 117B total and 5.1B active parameters per token (OpenAI, 2025). Runtime settings and the model/server identifiers available at experiment time are reported in Appendix E.

For a fixed seed and root set, we report the 2×2 paired outcome table between LEGACY and P/Q and the two-sided exact McNemar/binomial statistic on discordant pairs. We use these as paired finite-sample diagnostics over the observed roots. The inferential emphasis is on effect size, direction, and replication under a frozen contrast rather than on a single thresholded test. Reported p-values are not multiplicity-adjusted and are not used as the sole basis for any claim; the adaptive discovery analyses are labeled separately from frozen follow-ups. We do *not* pool repeated seeds into an iid significance test: the same 256 generated roots recur across seeds, and the generator itself induces structure, including dynamic-parent filtering before construction of static controls. Aggregated rates across seeds are therefore descriptive summaries only. Appendix B adds per-seed paired bootstrap intervals for the error-rate difference, conditional on resampling the observed roots.

3 EXPLORATORY DISCOVERY AND FREEZING THE RENDERER CONTRAST

Exploratory failure mining on Qwen revealed that an earlier branch had unintentionally changed the table renderer. We isolated that representation change on 256 fixed semantic roots, then froze the contrast for subsequent replication. The seed-17 discovery results below are adaptive.

Renderer isolation. With 256 semantic roots frozen, changing only the ambiguous A=/B= axis labels to explicit ROW=/COL= reduced Qwen Q4 natural wrong answers from 20/256 (7.81%) to 7/256 (2.73%). The paired table contained 18 W $\rightarrow$R and 5 R $\rightarrow$W transitions (exact McNemar $p = 0.0106$).

Exploratory label variants. We next varied the row and column labels separately and added a compact X=/Y= condition. Results were non-monotonic (Table 3): fixing only the row label improved the error rate, while the asymmetric hybrid A=/COL= made it worse and produced three non-natural completions. Fully disjoint ROW/COL and compact X/Y both reached 2.73%. This pattern is inconsistent with a simple claim that "more explicit English labels are always better" but does not separate namespace overlap from specific label identities, tokenization, or asymmetric notation.

Finally, P/Q replaced conventional X/Y with neutral P/Q. Its 5/256 wrong rate remained low, while 19 LEGACY failures recovered and only 4 roots regressed (one root failed in both), $p = 0.00260$. At

Table 4: Paired LEGACY versus P/Q results. W→R denotes roots wrong under LEGACY and correct under P/Q; R→W is the reverse. †Qwen Q4 seed 17 belongs to the adaptive discovery sequence; all other rows evaluate the frozen contrast. All listed rows have 256/256 natural, valid completions in both conditions.

Model	Seed	LEGACY W	P/Q W	W→R	R→W	p_{McN}
Qwen Q4	17†	20 (7.81%)	5 (1.95%)	19	4	2.60×10^{-3}
Qwen Q4	113	18 (7.03%)	5 (1.95%)	17	4	7.20×10^{-3}
Qwen Q4	991	16 (6.25%)	7 (2.73%)	15	6	7.84×10^{-2}
Qwen Q8	17	16 (6.25%)	2 (0.78%)	15	1	5.19×10^{-4}
gpt-oss-120b	17	34 (13.28%)	3 (1.17%)	34	3	1.23×10^{-7}
gpt-oss-120b	113	45 (17.58%)	3 (1.17%)	44	2	3.08×10^{-11}
gpt-oss-120b	991	37 (14.45%)	5 (1.95%)	35	3	6.68×10^{-8}

this point we froze LEGACY versus P/Q as the replication contrast. Because this discovery sequence is adaptive, we do not treat the seed-17 Qwen result as an independently pre-specified confirmatory test; the subsequent seed and model-family runs are the main replication evidence.

4 REPLICATION ACROSS SEEDS AND MODEL FAMILIES

4.1 QWEN REPLICATION

On Qwen3.8-27B UD-Q4_K_XL, LEGACY remains worse than P/Q across two additional sampling seeds: 18/256 versus 5/256 at seed 113 and 16/256 versus 7/256 at seed 991. The paired discordance favors P/Q at seed 113 (17 W →R versus 4 reverse, $p = 0.00720$) and again at seed 991 (15 versus 6, $p = 0.0784$). The seed-991 contrast has the same direction but does not reach the conventional unadjusted 0.05 threshold; its 95% paired bootstrap interval includes zero (Appendix B). These Qwen results provide weaker statistical evidence than the gpt-oss contrasts; seed-17 Q4 also belongs to adaptive discovery.

A Q8 configuration at seed 17 shows the same direction: 16/256 LEGACY errors (6.25%) versus 2/256 P/Q errors (0.78%), with 15 W →R and one reverse transition ($p = 5.19 \times 10^{-4}$). This does not establish quantization independence—the configurations are not a controlled quantization study—but it shows that the observed contrast is not restricted to the single tested Q4 artifact. Across the two frozen Q4 follow-up seeds and the Q8 run, the absolute LEGACY–P/Q error-rate gaps are 5.08, 3.52, and 5.47 percentage points, respectively; the adaptive seed-17 Q4 discovery gap is 5.86 points.

4.2 REPLICATION IN A SEPARATELY DEVELOPED MODEL FAMILY

We next evaluated a separately developed reasoning model family, gpt-oss-120b. Earlier Mistral and Gemma pilots had shown that a renderer comparison is hard to interpret when the model already fails frequently under the clean representation; those unfavorable pilots are reported in Appendix C. For subsequent family replication we therefore required a near-solved P/Q pilot before spending the full evaluation budget. This screen was adopted after the Mistral and Gemma pilots rather than preregistered. gpt-oss-120b met this criterion (15/16 correct on the first 16 P/Q roots at seed 17); the full 256-root paired evaluation covers seeds 17, 113, and 991. The screen is part of model selection, not an independent replication. Table 4 summarizes the discovery contrast and frozen follow-ups.

The gpt-oss contrast is large in every seed. P/Q accuracy is 98.83%, 98.83%, and 98.05%, while LEGACY accuracy is 86.72%, 82.42%, and 85.55%. The absolute error-rate gaps are 12.11, 16.41, and 12.50 percentage points, and the corresponding relative reductions in error when moving from LEGACY to P/Q are 91.2%, 93.3%, and 86.5%. Descriptively, across the three repeated seeds, LEGACY produces 116 wrong events out of 768 root-runs (15.10%) and P/Q 11/768 (1.43%), a 10.5× ratio; we do not attach a pooled iid p-value to this aggregate.

Order control. The first two gpt-oss full runs used P/Q before LEGACY. To test the simple alternative that a second run degrades because of runtime state, cache effects, thermal drift, or another temporal factor, seed 991 deliberately reversed the order: LEGACY first, then P/Q, without restarting the server. LEGACY still produced 37/256 errors before P/Q produced 5/256. This does not eliminate

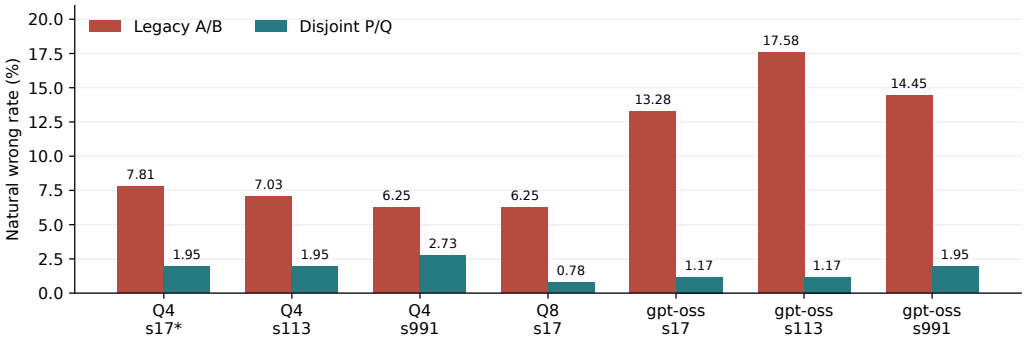

Figure 1: Natural wrong rates for the frozen LEGACY and P/Q renderers. The asterisk marks Qwen Q4 seed 17, which belongs to adaptive discovery; subsequent Qwen rows and all gpt-oss full runs evaluate the frozen contrast.

every temporal confound, and only one seed used the reversed order, so it is a targeted control rather than a fully counterbalanced design. It is, however, inconsistent with the direct explanation that the condition run second is systematically worse.

Completion length. A secondary descriptive signal is unusually stable: mean gpt-oss completion length under LEGACY versus P/Q is 536.3 vs. 355.9 tokens (seed 17), 528.3 vs. 350.5 (seed 113), and 539.5 vs. 354.0 (seed 991), approximately 1.5× in each run. We treat token count as an observable cost correlate, not as a direct measure of internal reasoning effort.

4.3 CROSSING ROLE LABELS WITH THE VALUE ALPHABET

The original renderer contrast changes label identity and overlap together. In a subsequent follow-up, we crossed row/column labels A/B versus W/X with value alphabets A–D versus W–Z on the same 256 roots. The latter alphabet is a bijective renaming $A \mapsto W$, $B \mapsto X$, $C \mapsto Y$, $D \mapsto Z$, applied consistently to table values, initial registers, the semantic contract, answer examples, and gold states. Program structure and table functions are preserved up to this renaming. Within each alphabet only axis-role labels vary. Two interpreters agree on all gold states, every transformation round-trips to the original prompt, and A/B with A–D is byte-identical to LEGACY.

All four conditions were newly evaluated on gpt-oss-120b at seed 17 and then replicated at seed 113, with low effort and the main-study sampler and completion ceiling (2,048 requests). Conditions are interleaved within each root using four balanced sequences; each condition occupies each position 64 times per seed. The executable protocol fixed the conditions, schedule, and contrasts before seed 17, following the earlier experiments; seed 113 was selected as a replication after inspecting seed 17. All responses were valid and naturally terminated, with no request errors or retries. Table 5 reports both runs; historical LEGACY responses are not substituted for these reruns.

Table 5: Crossed follow-up on gpt-oss-120b, low effort. Each cell evaluates the same 256 roots under a consistent value renaming. Labels are literal row/column prefixes in both alphabets: A/B overlaps A–D, whereas W/X overlaps W–Z (bold cells); the other two cells are disjoint.

Seed	Row/column labels	Values A–D	Values W–Z
17	A/B	**32/256 (12.50%)**	34/256 (13.28%)
17	W/X	13/256 (5.08%)	**55/256 (21.48%)**
113	A/B	**36/256 (14.06%)**	34/256 (13.28%)
113	W/X	8/256 (3.13%)	**62/256 (24.22%)**

At seed 17, the within-alphabet overlap penalties are 7.42 percentage points for A–D and 8.20 for W–Z. At seed 113, both penalties are 10.94 points. The preferred label pair therefore reverses across alphabets in both seeds. Mean overlap penalties are 7.81 points (95% paired bootstrap interval [3.91,

11.72]) and 10.94 points [7.03, 14.84], respectively. The corresponding differences of label effects are 15.63 points [7.81, 23.44] and 21.88 points [14.06, 29.69]. Appendix B gives each within-alphabet interval and discordance test. We analyze seeds separately because they reuse the same roots.

The byte-identical A/B–A–D prompt yields 34 versus 32 errors in the original and crossed runs at seed 17, and 45 versus 36 at seed 113: absolute differences of 0.78 and 3.52 percentage points. These cross-run discrepancies do not isolate stochastic variability from schedule or runtime differences, and two comparisons cannot characterize a run-level variance distribution. The paired bootstrap intervals quantify conditional root-resampling uncertainty, not uncertainty across repeated runs.

A descriptive 2×2 decomposition clarifies this crossover (Figure 2, Appendix B). Marginal A/B versus W/X error totals are 66 versus 68 at seed 17 and 70 versus 70 at seed 113, each out of 512 responses. A–D versus W–Z totals are 45 versus 89 and 44 versus 96: alphabet marginal differences of 8.59 and 10.16 points. Thus the observed label marginal is small while the label–alphabet interaction is large; marginal equality does not establish label equivalence. For fixed A/B, the alphabet marginal difference minus the mean overlap penalty equals the observed alphabet-substitution difference: $8.59 - 7.81 = 0.78$ and $10.16 - 10.94 = -0.78$ points (rounded). This cancellation is an algebraic decomposition of the same cells, not independent mechanistic validation. Overlap occupies the diagonal of this design, so its contrast is aligned with the interaction; tokenization and learned symbol associations remain unresolved.

5 Exploratory mechanism analysis

The final-answer results establish a renderer effect but do not identify where execution diverges. We therefore ran a deliberately small diagnostic study rather than another large benchmark sweep. We selected 12 roots: seven that had failed under LEGACY in all three earlier gpt-oss seeds and five roots that had been correct in both conditions. This cohort is intentionally enriched for susceptibility and is *not* population-representative.

Each root was evaluated under both renderers at three new pre-fixed sampling seeds (1701, 1702, 1703), for 72 requests total. Order was varied as LEGACY →P/Q, P/Q →LEGACY, and LEGACY →P/Q across the three seeds. P/Q was correct on all 36 root-seed trials. LEGACY produced 3/12, 3/12, and 4/12 natural wrong answers in the three seeds, respectively. Among the seven susceptibility-selected roots, LEGACY failed 9/21 trials; among five control roots it failed 1/15. Again, these rates describe the selected cohort only.

We logged the model's generated reasoning trace and, after final correctness was known, manually compared the first externally observable incorrect state transition against the deterministic interpreter. All ten final LEGACY failures first exhibited a wrong table-cell lookup. Examples include reporting the wrong value for $T_2(B, B)$, $T_1(D, A)$, or $T_0(B, B)$ while otherwise referring to the intended table/arguments. The first observed errors occurred at different instruction positions even for the same root across seeds, compatible with stochastic susceptibility rather than a single fixed "bad cell."

This diagnostic must be interpreted cautiously. The cohort was selected using earlier outcomes, and trace inspection was performed after final correctness was known; it is therefore a localization analysis, not an unbiased estimate of mechanism frequency. Generated chain-of-thought is also not guaranteed to faithfully expose a model's internal computation (Lanham et al., 2023). Indeed, two matched P/Q runs that ultimately answered correctly contained a transient intermediate lookup error that was later overwritten. The informative distinction is a wrong cell value despite referring to the intended table and arguments, rather than merely an error on a lookup instruction. Lookup assignments constitute 74.78% of instructions. Appendix F gives oracle-path exposure counts, but the selected, repeated-root trace cohort does not support a direct enrichment test against the whole-pool baseline. These ten annotations locate the observed errors without estimating a population mechanism frequency.

6 Related work

Prompt and representation sensitivity. Meaning-preserving prompt choices can substantially alter measured LLM performance. Sclar et al. (2024) report large swings under formatting variants, Mizrahi et al. (2024) argue for multi-prompt evaluation, and option order and serialization templates

can also change accuracy (Pezeshkpour and Hruschka, 2024; He et al., 2024). More specifically, Zhu et al. (2026) hold answer-bearing content fixed while changing local discourse-role labels and observe large shifts in whether models adopt that content; Narin (2024) likewise reports that symbolic variable naming affects mathematical problem solving. At the same time, Hua et al. (2025) show that heuristic scoring can exaggerate sensitivity by misclassifying semantically correct outputs. Our setting targets the intersection of these concerns: it uses a meaning-preserving representation intervention inside a deterministic symbolic executor task, while eliminating open-ended scoring ambiguity through a four-symbol state oracle and an identical exact parser across conditions. Unlike discourse-wrapper studies, the changed labels here denote local row/column roles inside the computation itself rather than the epistemic status of external context.

Symbolic and controlled reasoning benchmarks. BIG-bench helped establish broad capability testing across symbolic and algorithmic tasks (Srivastava et al., 2023). More targeted work studies symbolic operations directly (Dave et al., 2024) or converts non-natural symbolic representations to natural language to improve performance (Wang et al., 2024). GSM-Symbolic uses controllable symbolic templates to expose variance across instantiations of mathematical problems (Mirzadeh et al., 2025). EquiBench evaluates whether models recognize program-semantic equivalence under transformations including variable renaming, and finds evidence that surface syntax still influences judgments (Wei et al., 2025). Sui et al. (2024) study structural table understanding and input representations. Our intervention isolates local axis-role names rather than comparing broad table formats: the underlying symbolic instance is fixed, including every table cell and instruction, while only two axis-role names are changed. This permits direct paired root-level attribution within the benchmark contract.

Neural-symbolic execution and reasoning traces. Systems such as BINDER explicitly separate LM-generated symbolic forms from deterministic execution (Cheng et al., 2023), illustrating the reliability advantages of delegating exact operations to a formal executor. Our study instead asks what happens when an LLM itself performs the symbolic execution. Chain-of-thought can improve reasoning performance (Wei et al., 2022), but generated traces need not faithfully reflect internal computation (Lanham et al., 2023); approaches such as Faithful CoT explicitly introduce deterministic solvers to obtain stronger guarantees (Lyu et al., 2023). This motivates our distinction between exact final-state evidence and exploratory trace diagnostics.

7 Discussion and limitations

Representation robustness is a separate axis from capability. The clean P/Q condition is important because it provides direct evidence that the task is within the tested model's practical capability on these roots: gpt-oss-120b solves roughly 98–99% under P/Q. The LEGACY degradation therefore cannot be summarized as "the model is too weak for the benchmark." Conversely, models with poor P/Q baselines are not useful confirmatory instruments for this particular effect; Appendix C reports two such capability pilots rather than counting them as supporting or contradicting replications.

Capability gating and model selection. The second-family replication should not be read as a random sample of model families. Two earlier non-Qwen pilots performed poorly even under P/Q and therefore could not cleanly distinguish representation sensitivity from general task failure (Appendix C). We report those pilots explicitly. For later family testing, the clean-control gate was used as an instrumentation criterion: a model had to demonstrate that the benchmark was practically solvable before its LEGACY–P/Q difference was interpreted as renderer-specific evidence. This strengthens within-model attribution but narrows the population to models capable of the clean task.

Behavioral interaction versus internal mechanism. The crossed follow-up (Section 4.3) strengthens the overlap hypothesis: the better label pair reverses when the value alphabet changes in both seeds, and overlapping labels are worse within each alphabet. Nevertheless, alphabet substitution with A/B fixed does not consistently improve accuracy, and the earlier Qwen factorial is non-monotonic. Disjoint labels also differ descriptively: on A–D, W/X yields 13 and 8 errors at seeds 17 and 113, versus 3 and 3 for P/Q. These are different run batches, not a controlled W/X–P/Q contrast. Disjointness alone therefore does not establish near-perfect reliability; the evidence supports an overlap-associated interaction, not a universal overlap-only explanation. Neither this behavioral intervention nor the

selected trace study distinguishes tokenization, learned assignment conventions, attention, or internal role binding.

Implication for structured agent interfaces. Our results do not establish failure in arbitrary multi-agent systems, but they identify a concrete validation concern: a representation transformation can preserve a software-level schema contract without preserving LLM behavior. This matters wherever structured state is exchanged through independently designed interfaces or compact serializations. Representation-robustness testing can therefore complement nominal accuracy by checking whether decisions remain invariant under semantics-preserving changes to role names or serialization.

Scope and generalization. We evaluate one static-control table-execution pool, two principal model families, a small set of local inference artifacts, and three sampling seeds. The 256 roots are generated by one benchmark family and should not be treated as iid draws from all possible symbolic tasks. Model families also differ in templates, weights, and runtime details, so absolute error rates should not be interpreted as a controlled comparison between Qwen and gpt-oss. All primary runs use low reasoning effort; persistence at higher effort, under alternative samplers, and on independently generated roots remains untested. Future work should test independently generated semantics, additional families that pass the clean-control capability gate, deterministic or per-request reproducible decoding paths, and internal activation-level diagnostics.

Adaptive discovery. The renderer hypothesis was discovered through iterative benchmark development on Qwen seed 17. We expose this sequence rather than presenting the final contrast as pre-specified. The frozen follow-up evidence comprises new Qwen seeds, Qwen Q8, the gpt-oss runs, and the reversed-order control. The later crossed experiment was designed in response to the remaining overlap/identity ambiguity and was replicated at a second seed on the same roots. This distinction is essential for avoiding post-hoc overstatement.

8 Conclusion

Formally identical symbolic programs can exhibit sharply different LLM execution reliability when only the lookup-role labels are changed. In ExactBench-Render, replacing an overlapping `A/B` role namespace with neutral disjoint `P/Q` labels reduces natural wrong-answer rates from 13–18% to roughly 1–2% on gpt-oss-120b across three seeds, with the same direction on Qwen configurations. The same direction persists under a condition-order reversal; in the reported paired runs the wrong answers do not arise from truncation or invalid formatting, and the clean representation is near-solved. A small trace study further localizes observed failing trajectories to table-cell lookups, while also illustrating why generated reasoning should not be treated as a faithful internal trace. The crossed follow-up replicates a reversal of preferred role labels across value alphabets at two seeds, with overlapping labels worse within each alphabet; alphabet replacement alone does not guarantee improvement. These findings motivate reporting representation robustness alongside nominal task accuracy when evaluating LLMs used for structured execution.

Reproducibility statement

The accompanying package provides the benchmark generator, deterministic interpreters, paired prompt pools, inference runners, and analysis scripts. Per-instance final-answer records cover all seven main-table comparisons: 2,048 Qwen responses and 1,536 gpt-oss responses. Root identities, semantic hashes, oracle states, parsed answers, and paired counts were checked against these records. The package also contains all 72 reasoning traces from the diagnostic cohort and all 2,048 raw responses from the two-seed crossed follow-up, together with its fixed schedule and protocol. Appendix E gives inference settings and record coverage; Appendix B defines the statistics; Appendix A supplies a complete prompt; Appendix F reports oracle-path exposures.

AI use disclosure

The authors directed the study, determined the benchmark and experimental protocol, executed the experiments, evaluated alternative explanations, and selected the final analyses and claims. Generative

AI assisted with hypotheses, experimental design, benchmark and analysis code, result interpretation, literature discovery, and manuscript drafting and editing. Quantitative claims were checked against deterministic oracles, saved artifacts, and analysis scripts. The authors reviewed AI-assisted work, references, and factual claims and take responsibility for the final manuscript, code, results, and conclusions.

REFERENCES

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Binding language models in symbolic languages. In *International Conference on Learning Representations*, 2023.

Neisarg Dave, Daniel Kifer, C. Lee Giles, and Ankur Mali. Investigating symbolic capabilities of large language models. *arXiv preprint arXiv:2405.13209*, 2024.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X. Wang, and Sadid Hasan. Does prompt formatting have any impact on LLM performance? *arXiv preprint arXiv:2411.10541*, 2024.

Andong Hua, Kenan Tang, Chenhe Gu, Jindong Gu, Eric Wong, and Yao Qin. Flaw or artifact? Rethinking prompt sensitivity in evaluating LLMs. In *Proceedings of EMNLP*, pages 19889–19899, 2025.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In *Proceedings of IJCNLP-AACL*, pages 305–329, 2023.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. In *International Conference on Learning Representations*, 2025.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? A call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024.

Ali Emre Narin. Math problem solving: Enhancing large language models with semantically rich symbolic variables. In *Proceedings of the 2nd Workshop on Mathematical Natural Language Processing @ LREC-COLING 2024*, pages 19–24, 2024.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of NAACL*, pages 2006–2017, 2024.

Qwen Team. Qwen3.8-27B model card. `https://huggingface.co/Qwen/Qwen3.8-27B`, 2026.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, 2024.

Aarohi Srivastava et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets LLM: Can large language models understand structured table data? A benchmark and empirical study. In *Proceedings of WSDM*, 2024.

Yile Wang, Sijie Cheng, Zixin Sun, Peng Li, and Yang Liu. Speak it out: Solving symbol-related problems with symbol-to-language conversion for language models. In *ICLR Workshop: How Far Are We From AGI*, 2024.

Anjiang Wei, Jiannan Cao, Ran Li, Hongyu Chen, Yuhui Zhang, Ziheng Wang, Yuan Liu, Thiago S. F. X. Teixeira, Diyi Yang, Ke Wang, and Alex Aiken. EquiBench: Benchmarking large language models' reasoning about program semantics via equivalence checking. In *Proceedings of EMNLP*, pages 33868–33881, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 2022.

Jianguo Zhu, Xiangmei Li, and Wenjie Liu. Discourse-role labels as presentation-time variables for context use in large language models. *arXiv preprint arXiv:2606.04109v2*, 2026.

A COMPLETE EVALUATED PROMPT AND ORACLE EXECUTION

The following is the LEGACY prompt for root `CTL41-0001`; long prose lines are wrapped for typesetting. Machine-readable paired prompts are supplied in `evidence/pools/`. The P/Q version changes only table row prefixes `A=` to `P=` and column prefixes `B=` to `Q=`. No instruction or table value changes.

```
SEMANTIC CONTRACT
Register values are only the four symbols A, B, C, D.
T0, T1, and T2 are lookup tables. For Tn(x,y), use x as the ROW label and y as the COLUMN
label.
OPX and OPY do not have fixed meanings: their table is determined by the CURRENT mode.
SWITCH toggles MODE 0 <-> MODE 1 immediately. The new mode affects subsequent APPLY
instructions only.
HOLD does nothing and leaves both registers and mode unchanged.
The instruction stream is already parsed. No arithmetic, bitwise operation, ordering, or
unstated meaning exists.

START MODE: 1
START REGISTERS: R0=B, R1=C, R2=A

MODE MAP
MODE 0: OPX=T0, OPY=T2
MODE 1: OPX=T1, OPY=T0

LOOKUP TABLES
T0:
      B=A  B=B  B=C  B=D
A=A   B    D    C    A
A=B   A    C    D    B
A=C   C    A    B    D
A=D   D    A    B    C

T1:
      B=A  B=B  B=C  B=D
A=A   D    A    C    B
A=B   D    C    B    A
A=C   A    C    D    B
A=D   C    B    A    D

T2:
      B=A  B=B  B=C  B=D
A=A   D    B    C    A
A=B   B    A    D    C
A=C   B    C    A    D
A=D   A    D    B    C

PROGRAM
1. R1 := OPX(R0, R2)
2. R1 := OPX(R0, R2)
3. HOLD
4. R2 := OPX(R0, R1)
5. R2 := OPY(R2, R0)
6. HOLD
7. R0 := OPY(R1, R2)
8. R2 := OPY(R2, R0)

Compute the exact final register state and final mode.
End with exactly one line in this format:
FINAL_STATE: R0=A,R1=B,R2=C;MODE=0
Use the actual final symbols and mode in that line.
```

The oracle begins at $(B, C, A, 1)$. The register triples after steps 1–8 are (B, D, A), (B, D, A), (B, D, A), (B, D, A), (B, D, D), (B, D, D), (C, D, D), and (C, D, B). Mode stays 1. In particular, step 7 reads $T_0(D, D) = C$ and changes R_0 from B to C; step 8 uses that updated value to read $T_0(D, C) = B$. This evaluated example demonstrates state propagation without a mode transition.

B ADDITIONAL STATISTICAL DETAIL

For each seed, let b be the number of roots correct under LEGACY but wrong under P/Q, and c the number wrong under LEGACY but correct under P/Q. We report the exact two-sided McNemar/binomial value

$$p = \min\left(1, 2\sum_{i=0}^{\min(b,c)} \binom{b+c}{i} 2^{-(b+c)}\right).$$

The statistic is computed separately for each repeated seed. We do not pool repeated evaluations of the same roots into a nominal sample size of 768.

Table 6 reports 95% paired percentile bootstrap intervals for $\Delta = (c-b)/256$. For each seed we resample 256 paired outcomes with replacement, using 200,000 replicates. Since the statistic depends only on the four paired cell counts, a multinomial draw from $(c, b, WW, RR)/256$ exactly implements the empirical paired bootstrap without recovering root identities. The reproducible script uses NumPy's default RNG with seeds 20260917 through 20260923 in table order. These intervals assume exchangeable root resampling and describe uncertainty under that empirical distribution; they do not account for generator-level dependence, adaptive selection, or variation across new task families. Seeds are analyzed separately. They supplement, rather than replace, the exact discordance tests.

Table 6: Legacy minus P/Q natural wrong rate, in percentage points, with 95% paired percentile bootstrap intervals. †Adaptive discovery contrast.

Model	Seed	Difference	95% interval
Qwen Q4	17†	5.86	[2.34, 9.38]
Qwen Q4	113	5.08	[1.56, 8.59]
Qwen Q4	991	3.52	[0.00, 7.03]
Qwen Q8	17	5.47	[2.73, 8.59]
gpt-oss-120b	17	12.11	[7.81, 16.80]
gpt-oss-120b	113	16.41	[11.72, 21.09]
gpt-oss-120b	991	12.50	[8.20, 17.19]

Crossed follow-up contrasts. Let $e_{i,0}, e_{i,1}, e_{i,2}, e_{i,3}$ be binary error indicators for root i in conditions A/B–A–D, A/B–W–Z, W/X–A–D, and W/X–W–Z, respectively. Define $d_{i,A} = e_{i,0} - e_{i,2}$, $d_{i,W} = e_{i,3} - e_{i,1}$, and $d_{i,O} = (d_{i,A} + d_{i,W})/2$. Reported penalties are 100 times their root means. The interaction contrast is $\overline{d}_A + \overline{d}_W = 2\overline{d}_O$, taking A/B minus W/X on A–D and subtracting the same label contrast on W–Z. We resample entire four-condition root vectors together using 20,000 percentile bootstrap replicates (Python `random.Random`, seed 20260917). The same conditional-root limitations as above apply. Each seed is analyzed separately; the 512 root-seed quartets are not treated as independent roots. All 256 quartets per seed are natural and valid, so semantic-error and total-failure analyses coincide; no complete-case exclusions were necessary. Full raw responses, including generated reasoning, are retained, but this revision does not claim a full-cohort first-divergence analysis.

Table 7: Crossed follow-up: overlap minus disjoint error in percentage points. Intervals resample paired root quartets; p-values are two-sided exact McNemar diagnostics, unadjusted. "O-only/D-only" counts roots failing only with overlapping/disjoint labels.

Seed	Values	Difference	95% interval	O-only/D-only	p
17	A–D	7.42	[2.73, 12.50]	30/11	0.00432
17	W–Z	8.20	[1.95, 14.45]	44/23	0.0139
113	A–D	10.94	[6.25, 15.63]	34/6	8.36×10^{-6}
113	W–Z	10.94	[4.69, 17.19]	48/20	9.14×10^{-4}

Descriptive factorial decomposition. Let p_0, p_1, p_2, p_3 be the four condition error rates in the order above. The W/X-minus-A/B label marginal is $L = (p_2 + p_3 - p_0 - p_1)/2$, the W–Z-minus-A–D alphabet marginal is $A = (p_1 + p_3 - p_0 - p_2)/2$, and the mean overlap contrast is $O = (p_0 + p_3 - p_1 - p_2)/2$. The interaction is $2O$, and the fixed-A/B alphabet substitution obeys $p_1 - p_0 = A - O$ exactly. Marginals average over the two tested levels of the other factor; their 512 responses represent 256 repeated roots, not 512 independent instances. This descriptive decomposition introduces no additional significance test and does not identify an internal mechanism.

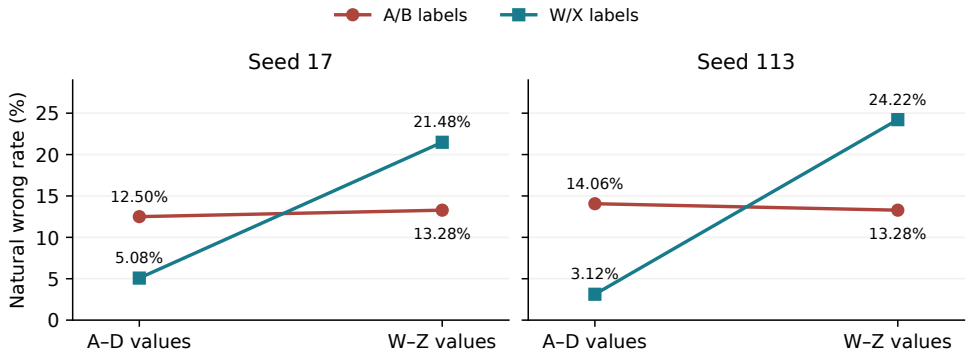

Figure 2: Role-label preference reverses across value alphabets in both crossed runs. Each point summarizes the same 256 roots. A/B overlaps A–D and W/X overlaps W–Z; the other combinations are disjoint. Lines connect observed error rates, not fitted predictions. Paired contrast intervals are reported in Table 7.

C CAPABILITY-GATE PILOTS NOT USED AS CONFIRMATORY EVIDENCE

We piloted two additional non-Qwen model families on 16 paired roots. Mistral-Small-3.2-24B-Instruct-2506-Q8_0 produced 13/16 LEGACY wrong answers and an effective 10/16 P/Q errors after accounting for one parser-whitespace artifact; Gemma-3-27B-IT-Q8_0 produced 10/16 LEGACY and 8/16 P/Q wrong answers under the corrected embedded-template configuration. Because neither model established a near-solved P/Q baseline, these pilots do not isolate a renderer-specific reliability degradation and are not treated as confirmatory evidence. They are included to make model-selection behavior visible rather than silently omitting unfavorable pilots. After these pilots, later cross-family replication used a clean-control screen: 0–1 wrong out of 16 was considered a strong pass, 2/16 a pass, 3/16 borderline, and 4/16 or more a stop criterion. The first 16 gpt-oss-120b P/Q records at seed 17 contain one wrong answer, meeting this criterion; these roots are included in the full study.

D MECHANISM-STUDY FIRST DIVERGENCES

Table 8 lists all ten final LEGACY failures in the 72-request mechanism cohort.

Table 8: First externally observable divergence in generated LEGACY traces. "Exact" is the deterministic table value.

Seed	Root	Step	Lookup	Trace / exact
1701	0044	9	$T_2(B, B)$	C / D
1701	0142	7	$T_1(D, A)$	C / B
1701	0189	8	$T_2(B, B)$	A / D
1702	0044	2	$T_2(C, D)$	D / A
1702	0142	8	$T_0(B, D)$	B / C
1702	0148	1	$T_0(B, B)$	A / D
1703	0044	9	$T_2(B, B)$	C / D
1703	0138	1	$T_2(C, C)$	C / A
1703	0189	1	$T_2(C, A)$	C / D
1703	0248	2	$T_0(B, B)$	B / D

Two correct P/Q traces for root 0044 contained a transient wrong lookup that was later overwritten. This is why we interpret reasoning traces as behavioral diagnostics rather than faithful internal-state records.

E Reproducibility details

Qwen. The principal discovery/replication model was Qwen3.8-27B served through `llama.cpp`, using a UD-Q4_K_XL GGUF for the three-seed Q4 experiments and a Q8 GGUF for the precision replication. Qwen runs used the explicit `low` reasoning-effort setting, $T = 1.0$, top-$p = 0.95$, top-$k = 20$, maximum completion 8192, and one request at a time.

gpt-oss. The gpt-oss artifact was locally labeled `gpt-oss-120b-F16.gguf`. The released model itself uses MXFP4 for the majority MoE weights (OpenAI, 2025); the local filename should therefore not be read as "all 117B parameters are FP16." The authors manually confirmed the original main-study server commit as `67672dc5b76f8bc17785a19d3dc6d1463fc2902c` and model SHA-256 as `2d1f0298ae4b6c874d5a468598c5ce17c1763b3fea99de10b1a07df93cef014f`. The saved server command uses `llama.cpp` with `--jinja --reasoning-effort low --parallel 1`, context 16384, batch 2048, micro-batch 512, and flash attention enabled. Requests used $T = 1.0$, top-$p = 0.95$, top-$k = 20$, and maximum completion 8192. OpenAI's released gpt-oss guidance recommends $T = 1.0$, top-$p = 1.0$ (OpenAI, 2025); we retained top-$p = 0.95$ to keep the paired study on the same fixed sampler used by the benchmark protocol. Because both renderers use identical decoding settings within every pair, the renderer contrast is internal to this configuration rather than a claim about optimal gpt-oss sampling. Seeds 17 and 113 ran P/Q then LEGACY; seed 991 reversed order without restarting the server. Reported seeds identify the recorded sampling configurations; bitwise repeatability across reruns or runtime builds was not established.

Crossed follow-up provenance. Both crossed runs record the server model identifier `gpt-oss-120b-F16.gguf`, context 16384, and the request parameters above, but do not independently record the loaded weight checksum or server commit. Root order uses seed 20260917. With condition indices as in Appendix B, within-root orders cycle through $(0, 1, 3, 2)$, $(1, 2, 0, 3)$, $(2, 3, 1, 0)$, and $(3, 0, 2, 1)$. Each ordered adjacent condition pair occurs 64 times within roots. The saved request/response events contain 1,024 starts and 1,024 saves per seed, without errors. The rerun A/B–A–D counts are 32 and 36, compared with the original counts of 34 and 45 at seeds 17 and 113. Sampling seeds label stochastic runs; exact repeatability across runs or builds was not established.

Record coverage. All main-table runs retain per-instance final-answer text, parsed states, root hashes, gold states, completion-token counts, and termination labels. The original gpt-oss main-study runner saved final-answer content but not the separate reasoning field. Reasoning is retained for all 72 diagnostic-cohort responses and all 2,048 crossed-follow-up responses. Final-answer comparisons are reproducible from the main-study records; trace-localization claims refer to the diagnostic cohort.

Pair integrity. For every primary paired comparison, analysis scripts verify identical `source_id` sets, identical root hashes, and identical gold states before computing transition counts. All main-table gpt-oss and Qwen comparisons reported in Table 4 contain only valid, naturally terminated outputs.

F Oracle-path lookup exposures

We execute each of the 256 semantic roots once with the deterministic oracle and log both argument values *before* each assignment. This produces 1,518 lookup events; repeated sampling seeds do not multiply the semantic exposure denominator. The two independent interpreters in the supplied generator agree on every final state, and all root hashes, gold states, and paired prompt transformations were checked. Table 9 summarizes argument frequencies.

Table 9: Argument exposure along correct execution paths. Percentages use 1,518 lookups as denominator.

Argument	A	B	C	D
Row x: count	383	332	405	398
Row x: percent	25.23	21.87	26.68	26.22
Column y: count	362	369	426	361
Column y: percent	23.85	24.31	28.06	23.78

The joint counts for rows and columns ordered A–D are

$$\begin{pmatrix} 183 & 58 & 81 & 61 \\ 46 & 174 & 66 & 46 \\ 73 & 61 & 206 & 65 \\ 60 & 76 & 73 & 189 \end{pmatrix}.$$

Equal-argument lookups account for 752/1,518 events (49.54%); $x = y = B$ accounts for 174/1,518 (11.46%). These frequencies are not uniform: operands may refer to the same register, and register values evolve along the program. Consequently a uniform-letter null is not an appropriate default baseline for error localization.

Six of the ten selected failing-trace annotations have $x = B$, and five have $x = y = B$. These are descriptive counts only: the cohort was selected using earlier failures, roots recur across seeds, and first errors are conditioned on survival through preceding instructions. The whole-pool exposure table is therefore not a matched risk-set denominator for those ten events. Establishing enrichment would require cohort- and position-matched analysis with dependence handled at root level. The deterministic oracle confirms the table, arguments, and exact value at all ten annotated positions. The accompanying diagnostic-cohort traces make the reported behavioral annotations available for inspection; these annotations do not constitute an automated full-cohort error taxonomy.