WHEN DO AGENT INTERVENTIONS HELP? MATCHING INTERVENTIONS TO RESIDUAL FAILURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern agent systems wrap a language model in skills, execution harnesses, recovery, reflection, and extra inference budget. These interventions, however, are usually evaluated in isolation or under a fixed configuration, obscuring whether an observed gain reflects intrinsic value or merely a match to the failures the surrounding system happens to leave behind. Rather than asking whether an intervention helps on average, we ask when each one helps and what failure it repairs. We run a matched factorial study over nine benchmarks, four models, four skill conditions, four harness configurations, and multiple inference budgets, for nearly 0.9 million agent trajectories. We find that intervention value is conditional rather than intrinsic: no single configuration is optimal everywhere, and the payoff of model, skill, harness, and budget interventions depends strongly on the task and its surrounding setup. This heterogeneity is structured by the residual failure, the failure that remains after the current system has done its best. Trajectory-level failure labels capture part of this structure, but diagnosis does not guarantee repair: even an observable execution failure need not be recoverable. These conditional patterns define capability boundaries and reveal real headroom for selective intervention. Yet the tested lightweight deployable routers fail to beat a fixed Harness policy on quality and cost, despite a far stronger outcome-aware oracle. Agent interventions should be judged by which failure they repair, under which configuration, and at what cost, not by average component gains alone.

1 INTRODUCTION

Modern agent systems increasingly combine a base language model with reusable skills, tool interfaces, environment observations, retry and recovery mechanisms, reflection, and additional inference budget; together, these surrounding components form an agent harness (Zhang et al., 2026). Systems such as ReAct, Reflexion, and Voyager already integrate several of these forms of support within a single execution stack (Yao et al., 2022; Shinn et al., 2023; Wang et al., 2023). Yet agent evaluations commonly treat the resulting stack as a single unit: benchmarks report task success or aggregate scores, while broader studies compare model backbones under largely fixed agent configurations (Liu et al., 2024; Ma et al., 2024a; Yao et al., 2024). These evaluations establish that a particular system performs better on a particular task, but they do not isolate which intervention produced the gain, which residual failure it addressed, or whether the same intervention remains useful under a different model, harness, or budget. Consequently, the literature provides system rankings and isolated improvements more readily than transferable design knowledge about when an intervention helps, becomes redundant, or causes interference.

When testing recent agent systems, we find that a stronger execution harness does not always improve performance (Figure 1a), that a Skill helping on one task can be ineffective or even harmful on another (Figure 1b), and that the strongest base Model does not necessarily win in every setting (Figure 1c). The variation also runs below the benchmark: instances grouped within one benchmark need not share a failure mode and can respond differently to the same intervention (Figure 1d), so a benchmark-level average collapses improved, unaffected, and degraded cases into one score. Together, these observations suggest that the value of an intervention is inherently task- and context-dependent. Rather than stacking increasingly powerful components by default, agent systems should select interventions matched to the bottlenecks present in a given task, which requires case-level analysis of failure trajectories and their underlying root causes.

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.

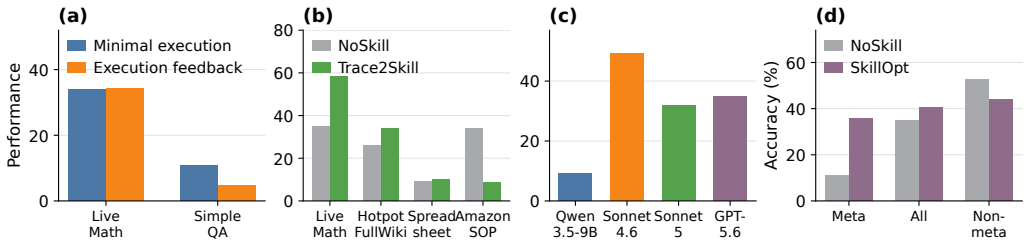


Figure 1: Motivating examples of intervention-matching and within-benchmark heterogeneity.

Guided by these observations, we build a factorial matrix over $T \times M \times S \times H \times B$ (task instance, base model, skill condition, execution harness, and inference budget), isolating each intervention's effect while holding the surrounding configuration fixed. Across nine benchmarks we evaluate $|T| = 1{,}872$ test instances with $|M| = 4$ models, $|S| = 4$ skill/contrast conditions, and $|R| = 3$ repeats per cell; the four harness configurations run at a historical default budget, and two of them are additionally swept over three inference budgets ($|B| = 3$; Low/Medium/High). This gives a nominal **nearly 0.9 million** trajectories after a few structurally unavailable cells (Appendix Table 11). We use three repeats to capture execution-level nondeterminism, characterized in a controlled Spreadsheet diagnostic (Appendix A.6). For every task instance and intervention condition, we retain the full execution trajectory together with its failure information. This trajectory-level record allows us to connect intervention effects to the failures they address, determine when an intervention is beneficial, absorbed, ineffective, or harmful, and identify recurring relationships between failure modes and intervention utility across tasks and system configurations.

Our study yields three main findings. **1)** Intervention value is conditional rather than intrinsic: no agent configuration is universally optimal, and across tasks, models, skills, execution harnesses, and inference budgets, the value of an intervention depends strongly on the surrounding configuration and task context. **2)** We find that this heterogeneity is structured by the residual failure left by the current system. Trajectory-level failure information provides partial predictive signal for which intervention will help. **3)** These conditional effects induce distinct capability boundaries and substantial headroom for selective intervention, but the tested lightweight deployable routers fail to achieve a quality-cost Pareto improvement over a fixed Harness policy. Together, our results argue that agent interventions should be evaluated not by average gain alone, but by the **source configuration, residual failure, and matched intervention response**.

2 Related Work

Agent stacks, interventions, and their evaluation. Language-model agents combine a base model with scaffolds for reasoning, tool use, environment interaction, memory, and recovery, as in ReAct, Reflexion, and Voyager (Yao et al., 2022; Shinn et al., 2023; Wang et al., 2023), while deployed frameworks such as SWE-agent and OpenHands make the agent–environment interface and runtime explicit (Yang et al., 2024; Wang et al., 2025). Benchmarks including AgentBench, AgentBoard, and τ-bench evaluate such systems across tools and environments (Liu et al., 2024; Ma et al., 2024a; Yao et al., 2024), but their scores summarize the whole stack and can obscure which component produced a gain (Kapoor et al., 2024), while holistic evaluations hold the scaffold fixed and vary only the model (Liang et al., 2022); concurrent work instead treats the execution harness as a first-class object of study (Zhang et al., 2026). Agent behavior can be changed without touching model weights: reusable skills encode procedures optimized or distilled from prior experience (Yang et al., 2026; Ni et al., 2026; Chen et al., 2026b), and counterfactual trace auditing shows that aggregate pass rates understate how strongly a Skill changes behavior, motivating paired comparisons with and without the Skill (Zhou et al., 2026). Execution-time interventions instead alter how the agent acts through action feedback, recovery, or reflection (Yao et al., 2022; Shinn et al., 2023; Madaan et al., 2023), and additional test-time computation helps unevenly with difficulty and allocation rather than uniformly with budget (Snell et al., 2025). Prior work thus establishes that each intervention can matter; we study when its effect persists, disappears, or reverses once other components of the stack change.

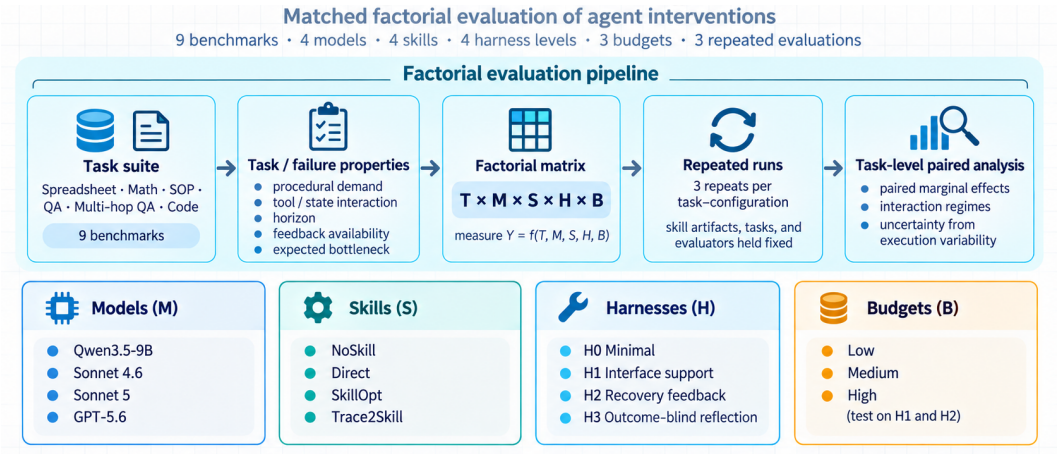


Figure 2: Matched factorial study design over $T \times M \times S \times H \times B$. Nine benchmarks are crossed with four Models, four Skill conditions, and four Harness levels (H0–H3), with Low/Medium/High Budget sweeps on two harnesses and three repeats per cell. Varying one axis at a time, the resulting matched task-instance outcomes feed the intervention, failure-mechanism, capability-boundary, and routing analyses.

Composition, failure analysis, and observability. Deployed jointly, these interventions cannot be valued from isolated comparisons: one-factor-at-a-time designs cannot estimate interactions, and attributed value depends on the reference configuration (Box et al., 2005; Shapley, 1952; Faigle & Kern, 1992), a weakness that has inflated apparent progress elsewhere (Henderson et al., 2018; Musgrave et al., 2020). Concurrent studies report the consequence: an all-components scaffold is often suboptimal (Liu, 2026), performance depends on scaffold compatibility and the full model–harness–environment configuration (Chen et al., 2026a), and rankings shift under re-implementation (Zhu et al., 2026b). A parallel line diagnoses where executions fail, categorizing errors and localizing decisive steps (Cemri et al., 2026; Zhu et al., 2025; Zhang et al., 2025; Liu et al., 2026; Deshpande et al., 2025; Barke et al., 2026; Fan et al., 2026), or asks whether the record exposes enough evidence to detect the failure at all (Balusu, 2026; Zhu et al., 2026a; Zhao, 2026).

We differ in three ways. First, scope: prior studies vary a subset of the axes and hold the rest fixed, whereas to our knowledge ours is the only study that crosses all five at once, task instance, Model, Skill, Harness, and Budget. Second, evidence: we do not stop at scores. Every execution leaves a trajectory, and we attach a failure label to each source execution before any matched outcome is inspected, which ties an intervention to the failure it acts on. Third, the question: pairing a labelled failure with matched outcomes lets us ask whether that failure is *addressable*, meaning whether it improves when one axis changes and whether the repair costs executions that already worked. This design also lets us test claims rather than inherit them: Liu (2026) report interference fading with model capability, whereas our patterns persist across the three stronger API models (Appendix Table 17).

3 Experimental Design

Following the intervention-matching gap and within-benchmark heterogeneity observed in the Introduction, we design the experiment around task context rather than treating the agent stack as a single unit. We construct a matched factorial matrix over $T \times M \times S \times H \times B$: each task instance T is evaluated under **four** base Models M, **four** Skill conditions S, and **four** Harness configurations H, and for two of those harnesses under **three** inference Budgets B, with three repeats per cell. The Model axis spans backbones of increasing nominal capability, Qwen3.5-9B, Sonnet 4.6, Sonnet 5, and GPT-5.6, which lets us ask whether the value of an external Skill, Harness, or Budget transfers across model strength. Figure 2 summarizes the design: the four intervention axes, which are the factors of the design, are motivated in turn below. Appendix A.1 gives the complete operational definitions and resource ceilings.

Table 1: Human-authored descriptive task properties and task-structure intervention hypotheses. Proc. = procedural; Fb. = feedback; Med. = medium. "Expected intervention" records a design hypothesis.

Benchmark	Proc. demand	Tool/ state	Horizon	Fb. demand	Fb. avail.	Expected bottleneck	Expected intervention
Spreadsheet	Med.	High	Med.	High	High	Execution/tool	H2; H3
LiveMath	Low	Low	Med.	Low	Low	Reasoning/decision rule	Skill; Model
UCSB SOP	High	High	Med.	High	High	Procedural	Skill; H2
Amazon SOP	High	High	High	High	High	Execution/tool; horizon	H2; H3; Budget
SimpleQA	Low	Low	Low	Low	Low	Factual knowledge/recall	Model
SQuAD	Low	Low	Low	Low	High	Evaluation/answer contract	H1
Hotpot Context	Low	Low	Med.	Low	Med.	Evaluation/answer contract	H1
Hotpot FullWiki	Med.	High	High	High	Med.	Horizon/termination	Skill; H2; Budget

3.1 BENCHMARKS

We select **nine** benchmarks so that the suite spans distinct capabilities and, by construction, distinct dominant failure modes, rather than repeatedly stressing the same one. This diversity is what makes a matched intervention study meaningful: a single benchmark would confound an intervention's value with the one bottleneck it happens to impose, whereas a capability-diverse suite lets us observe when the same intervention helps, saturates, or backfires as the underlying difficulty changes. Table 1 records our manual, human-authored annotation of each benchmark's task properties and hypothesized bottleneck, which we use only to diversify benchmark selection and never as evidence in the failure analysis; the per-benchmark split accounting is given in Appendix Table 10. The specific capability each benchmark exercises, and why it makes a meaningful test bed, is described in Appendix A.2.

3.2 SKILLS

A Skill is a reusable, natural-language procedure supplied to the model before execution and frozen for the whole study, so that any measured difference reflects the supplied procedure rather than online adaptation. We compare **four** Skill conditions that differ in how the procedure is constructed and in the execution evidence available during construction: NOSKILL supplies no procedure and serves as the reference; DIRECT supplies a one-shot procedure synthesized from the task specification without any rollout evidence; SKILLOPT (Yang et al., 2026) optimizes the procedure through outcome-guided iteration on training rollouts; and TRACE2SKILL (Ni et al., 2026) distills a reusable procedure from prior execution traces. This design is meaningful because several benchmarks are limited by a missing reusable procedure rather than by model capability or execution support: LiveMath rewards a narrow answer-selection rule, UCSB SOP rewards encoded ordering and verification constraints, and full-wiki HotpotQA rewards a multi-hop search-and-integration policy. Contrasting NOSKILL, DIRECT, SKILLOPT, and TRACE2SKILL therefore separates whether value comes from having any procedure at all, from outcome-guided optimization, or from trajectory distillation. All artifacts are frozen and transferred unchanged across Models, so a Skill cannot silently adapt to the target; Appendix Figure 7 and Appendix A.1 detail the construction pathways and provenance.

3.3 HARNESS

A Harness changes the information and actions available during execution rather than supplying a task policy beforehand, and the **four** levels form a deliberate progression from minimal execution to increasing feedback and repair. H0 is a minimal task-native wrapper; H1 adds a standardized output or tool interface (Patil et al., 2024); H2 additionally exposes executable observations and errors and permits bounded retry or local repair (Yao et al., 2022); and H3 adds one outcome-blind revision without access to evaluator outcomes or gold answers (Madaan et al., 2023). This progression is meaningful because our state-changing tool workflows, such as SpreadsheetBench and Amazon SOP, can fail during execution even when the overall plan is correct, so exposing execution feedback and permitting bounded recovery should help without changing the model, following the action–observation principle of ReAct (Yao et al., 2022); the answer-interface-sensitive benchmarks, such as SQuAD and provided-context HotpotQA, instead motivate the H1 contract; and H3 tests whether one further outcome-blind revision adds value once execution support is present. Because H2 jointly changes failure observability and the opportunity to act on it, an H2 gain cannot be attributed separately to error exposure or to retry; Appendix Figure 8 and Table 7 make the levels explicit.

3.4 BUDGET

We treat inference Budget as an independent axis with Low, Medium, and High ceilings crossed only with H1 and H2, while H0–H3 are also run at each benchmark's historical Default configuration. The sweep jointly raises completion-token, turn, and timeout ceilings, so it tests whether a trajectory is resource-feasible and where performance saturates rather than isolating token-only scaling; excessive ceilings can raise cost without improving outcomes.

3.5 PROTOCOL AND REPORTED QUANTITIES

Repeats. We use three repeats per cell to capture execution-level nondeterminism, whose magnitude we characterize in a controlled SpreadsheetBench diagnostic (Appendix A.6). **Evaluation.** Every benchmark keeps its native evaluator and native success threshold, and we compare matched task-instance and model configurations that change exactly one axis while holding the rest fixed (Eq. 1). Native scores are never pooled across benchmarks; cross-benchmark summaries use equal-weight macro averaging so that large test sets do not dominate. Task instances, frozen Skill artifacts, and evaluators are held fixed, and we use frozen train, method-validation, harness-validation, and test splits, with test instances locked and excluded from Skill construction and selection (Appendix A.3). Because the opportunity tables report observed positive changes rather than gains certified to exceed execution noise, we distinguish observed change, noise-qualified improvement, and success crossing throughout, and report the calibration status of each in Appendix A.6 and Appendix A.5. **Within-axis envelopes.** To summarize how much room each axis has on a benchmark, we report four within-axis observed envelopes: Δ_S^+, Δ_H^+ and Δ_B^+ are the largest positive gain that any alternative Skill, any Harness level above H0, or any Budget endpoint above Low achieves over its own reference, each clipped at zero, while Δ_M is the best-to-worst span across the four Models because no Model is a natural baseline. All four are maxima over a candidate set and use different reference cells and different candidate counts, so each bounds the room on one axis and they are not a ranking across axes; Eq. 2 gives the formal definitions.

4 RESULTS

We report the factorial outcomes in the order the questions stack: first whether any configuration wins everywhere, then what the task, the observed failure, and the tested action set each say about which intervention to reach for, and finally whether that structure can be exploited prospectively. We start with the full result matrix and dip into it step by step, from the benchmark-level response down to the individual labelled failure. Appendix D complements this axis-by-axis reading with a case study of each benchmark, from its task properties to its best observed configuration.

4.1 NO UNIVERSALLY OPTIMAL AGENT CONFIGURATION.

No single configuration is best everywhere: interventions solve different problems rather than supply uniformly stronger computation, and within one benchmark the same intervention can help one subgroup of task instances and harm another. The complete 512-cell Default-budget surface is reported in Table 2. Model rankings are not context-free, since different Models become optimal across Benchmark×Skill conditions and their ordering shifts with the surrounding Harness or Skill (Figure 3a). Skills, Harnesses, and Budgets show the same conditionality, with large gains on some tasks but little benefit or even degradation on others depending on the task structure and existing configuration (Figure 3b). This heterogeneity also appears within individual Benchmark×Model blocks, where changing a single intervention can produce large improvements or severe regressions (Figure 3c). Adding pairwise interaction terms improves held-cell prediction on seven of eight benchmarks (mean leave-one-cell-out R^2 0.729 to 0.812; paired sign-flip $p = 0.0078$), so these effects are not independent additive component gains. Subgroup responses can even oppose each other within one benchmark: on LiveMath with Sonnet 5, TRACE2SKILL gains 6.60 points on the 53 meta-option instances, those carrying LiveMath's recurring answer choice that is correct whenever it appears, but loses 4.93 on the 71 without it, averaged over H0–H3, leaving a whole-benchmark effect near zero (Figure 3d).

Table 2: Combined Default-budget Skill × Harness performance matrix. Rows follow the Benchmark × Skill layout; columns group each Model into H0–H3 Harness conditions. Each entry is the mean benchmark-native performance on a 0–100 scale over three complete repeats. **Bold blue**, a red-shaded cell, and a box mark the row, block and benchmark maxima respectively.

Benchmark	Skill / contrast	Qwen3.5-9B				Sonnet 4.6				Sonnet 5				GPT-5.6			
		H0	H1	H2	H3	H0	H1	H2	H3	H0	H1	H2	H3	H0	H1	H2	H3
Spreadsheet	NoSkill	12.5	9.3	24.9	30.4	48.6	49.2	51.1	**53.1**	33.5	31.9	34.5	34.2	35.1	35.0	36.0	37.5
	Direct	14.3	13.3	28.6	28.8	45.8	43.3	45.8	**48.9**	30.1	31.4	34.0	28.2	33.7	33.0	35.0	36.0
	SkillOpt	8.0	9.3	22.5	26.2	34.2	39.6	**40.4**	36.1	30.2	30.5	33.2	29.0	33.3	34.0	34.8	37.3
	Trace2Skill	9.3	10.1	22.7	25.1	34.5	36.7	38.0	**39.3**	26.1	25.7	26.9	24.5	33.2	33.7	34.5	36.5
LiveMath	NoSkill	34.1	34.9	34.4	36.3	31.7	32.0	30.9	31.7	40.1	38.7	37.9	35.2	52.2	**53.5**	51.9	49.2
	Direct	33.3	32.3	33.3	30.9	28.5	29.8	31.5	29.8	36.3	36.6	38.4	37.4	50.0	**51.9**	49.7	51.1
	SkillOpt	40.3	40.6	39.8	40.6	31.5	33.6	33.1	35.5	34.7	37.4	37.1	32.5	55.4	55.1	**58.6**	54.3
	Trace2Skill	53.2	58.1	56.5	54.6	33.9	34.4	33.6	37.1	40.6	39.0	36.8	35.5	61.6	**62.1**	57.8	60.5
UCSB SOP	NoSkill	52.8	51.0	52.6	59.2	61.7	60.2	60.2	60.2	66.7	65.9	66.7	64.2	69.9	69.3	**70.3**	68.4
	Direct	54.1	57.5	55.2	60.4	59.0	58.7	59.4	63.0	66.5	65.5	66.1	61.3	68.2	**68.6**	68.2	66.9
	SkillOpt	66.3	66.3	65.9	62.9	67.4	67.2	65.9	65.1	68.6	67.8	69.3	67.6	72.6	**73.1**	71.6	70.3
	Trace2Skill	62.1	61.9	61.9	62.1	58.9	57.5	60.6	62.9	65.1	66.1	65.0	64.0	**71.8**	70.3	71.2	68.2
Amazon SOP	NoSkill	28.3	33.9	88.6	95.1	82.7	82.7	**100.0**	99.6	2.6	4.2	58.9	87.8	58.5	57.1	**100.0**	99.0
	Direct	1.7	1.7	1.7	1.7	90.8	80.2	**99.4**	99.2	2.6	1.7	85.7	89.5	57.8	57.8	95.7	99.0
	SkillOpt	8.8	6.7	94.8	98.2	**100.0**	**100.0**	**100.0**	**100.0**	1.7	9.0	61.5	80.0	59.1	58.5	**100.0**	99.3
	Trace2Skill	19.6	8.6	80.6	93.3	77.1	82.9	**100.0**	99.9	9.4	6.8	58.5	85.1	57.1	55.9	98.0	99.3
SimpleQA	NoSkill	10.7	5.1	4.7	5.0	33.8	30.1	29.9	29.7	31.6	29.2	30.4	29.6	**69.4**	67.8	68.0	68.8
	Direct	8.0	6.3	6.1	6.4	28.2	29.7	30.1	29.3	27.6	25.0	24.6	25.7	65.0	66.1	67.0	**69.2**
	SkillOpt	11.3	6.3	6.3	5.9	33.4	30.1	30.1	32.9	27.7	30.9	29.3	30.8	67.6	68.0	66.6	**68.4**
	Trace2Skill	9.9	6.0	5.4	5.6	33.3	29.8	29.0	31.1	30.3	31.2	31.2	28.0	67.8	67.4	66.9	**68.8**
SQuAD	NoSkill	10.7	36.6	78.6	79.3	0.0	85.4	85.6	86.3	1.8	**90.0**	89.0	89.4	23.9	63.4	63.1	68.3
	Direct	46.7	76.4	75.9	76.3	42.4	80.4	80.3	82.0	37.6	**89.2**	88.3	87.4	59.3	73.6	74.4	77.2
	SkillOpt	29.0	77.0	77.6	77.4	0.0	82.4	82.1	82.6	50.3	**87.1**	86.9	84.7	49.7	63.8	64.1	66.6
	Trace2Skill	76.9	55.0	80.8	79.2	16.1	86.7	86.9	87.2	69.3	**91.0**	90.6	90.7	72.8	80.6	80.8	81.1
Hotpot Context	NoSkill	18.6	59.3	59.7	58.6	0.0	66.2	66.2	**67.8**	3.4	64.3	64.6	65.8	22.6	62.0	61.9	61.3
	Direct	21.3	55.2	55.3	57.3	16.6	66.8	66.9	**69.2**	9.9	62.9	62.7	64.8	29.7	64.6	64.3	60.9
	SkillOpt	0.0	44.0	47.9	59.7	0.0	68.2	68.4	**68.6**	0.0	63.9	63.0	64.9	0.0	13.6	12.2	12.2
	Trace2Skill	26.2	58.4	58.6	59.2	1.0	68.6	68.1	**68.7**	17.6	62.8	62.1	65.4	28.0	61.6	62.0	61.3
Hotpot FullWiki	NoSkill	24.2	25.9	26.8	26.9	29.1	30.4	29.3	31.1	31.1	30.9	31.3	**33.2**	21.0	22.6	23.1	19.9
	Direct	1.4	3.9	28.4	**29.8**	26.7	26.1	26.7	28.0	27.2	26.7	27.8	27.4	27.2	27.2	27.2	26.8
	SkillOpt	21.1	24.0	34.9	**35.6**	19.8	20.7	21.0	23.9	20.1	24.0	23.6	24.4	15.2	20.0	17.1	16.8
	Trace2Skill	32.7	34.1	36.0	39.1	44.0	41.6	41.7	42.7	41.1	**45.0**	43.2	41.9	31.6	30.1	32.6	28.9

Take-away 1: There is no universally best configuration

No configuration is best everywhere, and the benchmark average hides even more: across benchmarks the best Model, Skill, Harness, or Budget keeps changing, and even inside one benchmark a single intervention can help some instances while hurting others, so an intervention's value is conditional rather than intrinsic.

4.2 Which Interventions Help Under Which Task

Tables 2 and 3 separate the benchmarks by the axis that carries their largest response. SimpleQA is the clearest Model-led pattern ($\Delta_M = 60.85$ against $\Delta_S^+ = 0.13$ and $\Delta_H^+ = 0.00$): it asks for factual knowledge that is generally absent from the task context, and neither a procedural Skill nor execution feedback supplies a fact the base model does not hold. UCSB SOP is the clearest Skill-led pattern ($\Delta_S^+ = 5.55$ against $\Delta_H^+ = 0.00$), where the task rewards following an explicit decision procedure. On Spreadsheet and Amazon SOP, Harness is the axis with traction among the external interventions ($\Delta_H^+ = 5.54$ and 48.03, both against $\Delta_S^+ = 0.00$), because a reasonable plan can still fail locally during execution and H2 can expose that failure and retry it. Which harness level carries the gain separates two mechanisms: on Spreadsheet and Amazon SOP it sits at H1→H2 ($+4.8$ and $+42.2$ points, averaged over Models and Skills), where executable feedback and bounded recovery act, whereas on SQuAD and Hotpot Context it sits at H0→H1 ($+39.5$ and $+46.7$), where the harness supplies the answer interface rather than repairing an execution error (Appendix C.4 gives the per-benchmark bottleneck profiles).

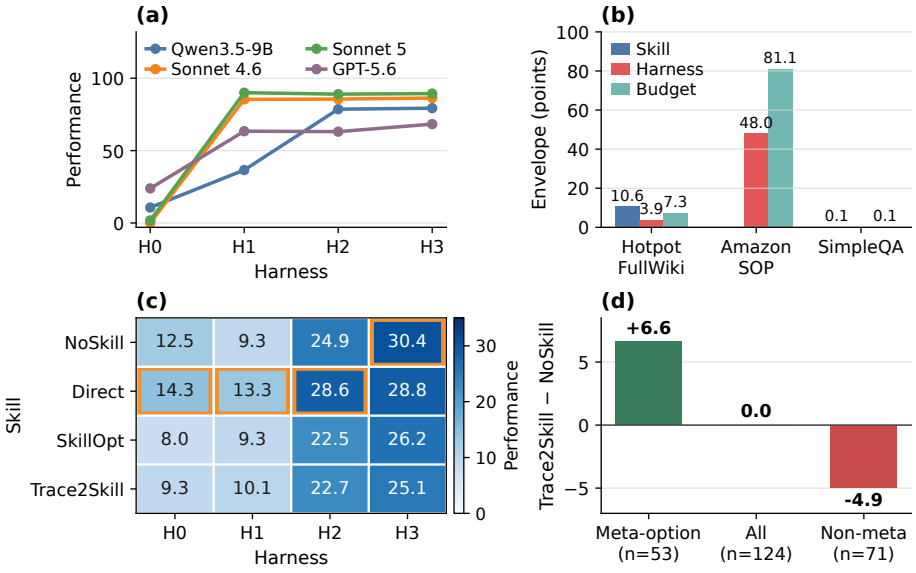


Figure 3: Intervention-response heterogeneity.

Table 3: Benchmark-level within-axis observed envelopes in percentage points (Eq. 2). Blue marks the largest envelope per row; the columns are not comparable across axes (Section 3.5).

Benchmark	Δ_S^+	Δ_H^+	Δ_B^+	Δ_M
Spreadsheet	0.00	5.54	4.18	24.33
LiveMath	8.15	0.79	2.67	22.26
UCSB SOP	5.55	0.00	NA	10.42
Amazon SOP	0.00	48.03	81.13	53.10
SimpleQA	0.13	0.00	0.10	60.85
SQuAD	17.13	44.34	0.24	11.88
Hotpot Context	1.71	48.18	0.00	9.57
Hotpot FullWiki	10.58	3.92	7.26	6.99

Take-away 2: A task's bottleneck already points to the axis

Because an intervention's value is conditional, a task's dominant bottleneck already indicates which axis to reach for: tasks bottlenecked by missing knowledge call for a stronger model, and tasks bottlenecked by execution or interface errors call for a richer harness.

4.3 Which Interventions Help Under Which Failure

Table 4 labels why the source configuration failed. These labels are read off each saved trajectory. The composition matches the task reading: SimpleQA's labeled failures are 74.84% Factual knowledge/recall, UCSB SOP is 78.57% Procedural, and Spreadsheet and Amazon SOP are 99.83% and 100% Execution/tool. Because six of the eight benchmarks are effectively single-root, the composition alone says no more than benchmark identity does. The label earns its place only when tested against matched outcomes.

Appendix C.6 turns that correspondence into a test: a mapping from failure label to axis, Procedural→Skill, Execution, State and answer contract→Harness, and Reasoning, Factual knowledge and Abstention→Model, is frozen before any outcome is loaded, then each source failure is replayed under the mapped action and under the mean of the other three axes. For each frozen source-failure pattern, Figure 4 plots the failed-source score, the mean of the three non-aligned axes, and the semantically mapped action over 832 Qwen NoSkill+H1+Medium failures, 53 procedural, 465 execution and 314 capability. Two mappings hold: Execution→Harness gains +6.54 points over

Table 4: Root-cause composition of the 2,225 labeled NoSkill+H1+Medium source failures, with n the benchmark denominator. *Hypotheses from the Qwen3.5-9B judger (Appendix C.6).

Benchmark	n	Proc.	Reas.	Exec.	State	Horiz.	Fact.	Abst.	Contr.
Spreadsheet	588	0.00	0.00	99.83	0.00	0.17	0.00	0.00	0.00
LiveMath	305	0.33	73.44*	0.00	26.23*	0.00	0.00	0.00	0.00
UCSB SOP	266	78.57	0.00	7.52	13.91	0.00	0.00	0.00	0.00
Amazon	113	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
SimpleQA	783	0.00	0.00	0.00	0.00	0.00	74.84*	24.78*	0.38
SQuAD	159	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
Hotpot Context	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
Hotpot FullWiki	10	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00

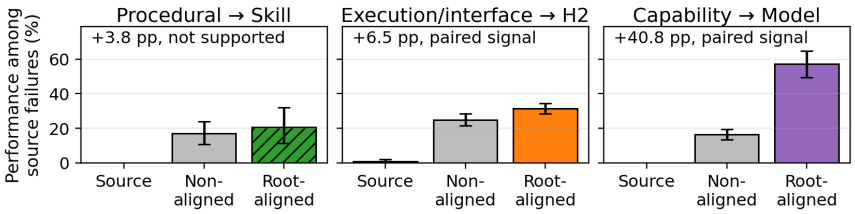


Figure 4: Root-aligned versus non-aligned intervention performance.

the non-aligned mean (95% CI $[+2.21, +10.77]$, seven benchmarks) and Capability→Model $+40.78$ ($[+32.28, +49.11]$, two). Procedural→Skill moves the predicted way at $+3.77$, but its interval spans zero ($[-5.66, +14.47]$) on a single benchmark. Root-aligned means taxonomy-aligned, not empirically optimal: the mapped Harness action still does not beat the strongest alternative. Appendix Table 22 defines the eight root meanings.

We also find that aligning the intervention to the failure is not sufficient, because an exposed, actionable failure converts into a durable repair on some benchmarks and almost never on others, and observability alone does not predict whether a recovery intervention will help (Appendix C.7).

Take-away 3: Match the failure, then actually repair it

Failure type points to the fix, missing capability to a stronger model and execution errors to a richer harness, though missing procedure does not map to a skill; but even a confirmed match becomes a real recovery only when the exposed error is repaired and that repair survives to the final answer.

4.4 What Capability Boundaries Remain, and Can They Be Routed?

Figure 5 gives the uniform decomposition. Each comparison selects one action per axis, enumerating all 3×3 Skill–Model pairs uniformly with H2 and High budget as singleton actions, and an axis crosses when its selected action reaches native success. Panel (a) shows 95% task-cluster bootstrap intervals; root labels are abbreviated and per-root denominators are in Table 23, where the single Horizon/termination case is descriptive. No tested axis reaches success in 59.6% of comparisons. Model alone succeeds in 17.3% and several axes succeed in 16.6%, while Harness-only, Skill-only, and Budget-only crossings account for 3.3%, 2.3%, and 1.0%. Having one action available on every axis therefore does not make a failed source recoverable. The cases that do admit several valid actions can reach the same outcome at different costs (Appendix C.8).

Whether that headroom is usable depends on choosing the right action in advance, without first paying a probe to find out. Table 5 compares five policies that must choose from the task text alone, under held-family historical replay. Trace H1 and Trace H2 apply one harness level to every instance without routing. The Text+action Router predicts the action directly from task text, whereas Text→root→action first predicts a root cause and then maps that label to an action. The Observed per-instance Oracle picks the best action using matched outcomes and is an unattainable reference. Every policy acts on Qwen3.5-9B, so ranking by mean full-chain tokens ranks by cost.

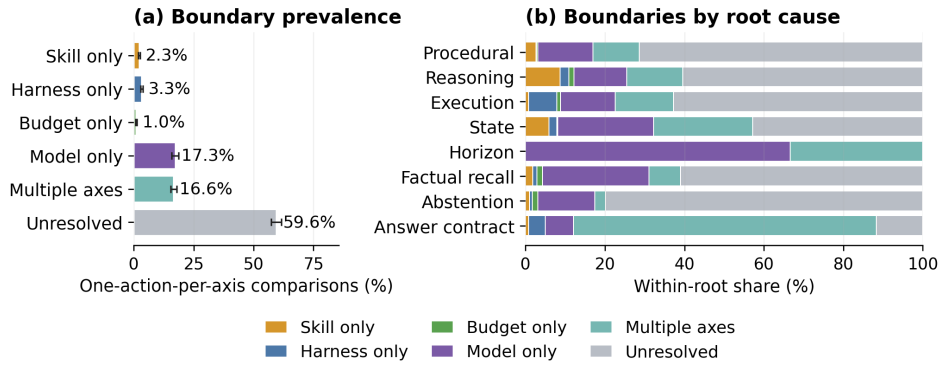


Figure 5: Equal-cardinality empirical capability boundaries on 2,225 labeled source failures.

Table 5: Five routing policies.

Policy	Performance	Mean full-chain tokens
Trace H1	38.99%	2,853.51
Trace H2 reference	45.29%	4,070.32
Text+action Router	39.42%	3,620.89
Text→root→action	40.10%	4,264.12
Observed per-instance Oracle	58.12%	2,084.54

Predicting the root cause first helps slightly, 40.10% against 39.42%, but that difference is not statistically resolved. Both routers stay below the Trace H2 reference of 45.29%, so no deployable router is yet established, while the Oracle reaches 58.12%. That gap makes intervention selection a value-of-information problem: acquiring the signal can consume the benefit of acting on it (Appendix C.9).

Take-away 4: Most failures are out of reach, and the rest cannot yet be routed

Even with the aligned intervention, most failures stay out of reach: model changes and multi-axis combinations recover most of what is recoverable, while harness, skill, and budget help only when their precondition holds. The cases with several valid actions leave real headroom, yet a lightweight router keyed on the task or failure type still cannot beat always running H2 on quality or cost, so capturing it needs a stronger, costlier signal.

5 Discussion

Our four results share one structure. The *source configuration* matters: no configuration is best everywhere, and inside a single benchmark the same intervention helps some instances and hurts others. The *residual failure* selects the axis, missing capability answering to a stronger model and execution errors to a richer harness. The *matched response* is bounded: model changes and multi-axis combinations recover most of what is recoverable, while skill, harness, and budget help only where their precondition holds. Agent evaluation should therefore report *source configuration* + *residual failure* + *matched intervention response* rather than an average component gain.

Three questions stay open. The missing procedure did not stably map to a skill, so procedural failures lack a demonstrated repair. Diagnosis did not guarantee repair either: an exposed failure became a durable one on some benchmarks and almost never on others, leaving what makes a failure recoverable unexplained. Most failures were beyond every single tested action, and no lightweight router beat always running H2, so the remaining headroom needs a stronger and costlier decision signal. These boundaries hold only for the benchmarks, models, skills, harnesses, and budgets tested here, and small effects should be read against the measured noise floor.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance, for literature retrieval and discovery, for code used in analysis and figure generation, and as an experimental instrument. Specifically, large language models are the object of study throughout: every evaluated configuration is a model execution, the DIRECT, SKILLOPT, and TRACE2SKILL skill artifacts are model-generated, and a fixed Qwen3.5-9B judge assigns root-cause hypotheses where deterministic trajectory evidence is insufficient (Appendix C.6). Separately, the authors used a general-purpose assistant for prose editing, for locating candidate related work, and for checking numerical consistency between tables. We have not used generative AI tools for research ideation, for experimental design, for selecting which results to report, or for authorship-level contributions; peer review is not applicable to this work.

We have reviewed all AI-assisted work. Every number reported in this paper was verified by the authors against the saved run artifacts, and the internal consistency of the root-cause, opportunity, and transition tables was checked by hand as well as by script. Model-assigned root-cause labels are marked separately from evidence-backed labels in every table and are treated as hypotheses rather than measurements, and Appendix C.6 states the reliability study that would be required to promote them. AI-assisted analysis code was read and re-run by the authors, and every assistant-suggested reference was read in the original before being cited. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work evaluates existing language models on existing public benchmarks and released benchmark suites. It collects no human-subject data and no personally identifying information, and the SOP suites describe industrial and operational procedures rather than personal records. We report negative and null results, including the fact that the H1-conditioned router is dominated by a fixed policy and the failure of the procedural root-to-skill mapping, because selective reporting of intervention gains is the specific evaluation pathology this paper argues against. Two risks deserve naming. First, the frozen skill artifacts derive from one model's trajectories, so reusing them elsewhere may transfer that model's idiosyncrasies. Second, our capability-boundary estimates are configuration-specific and should not be cited as evidence that any class of agent failure is intrinsically unfixable. Provider API prices used for the cost ledger are public rates recorded at experiment time and will change.

REPRODUCIBILITY STATEMENT

Appendix A specifies the design matrix, the nominal budget ceilings, the native success thresholds, and the execution ledger. Appendix A.5 defines the evaluation metrics and statistical abbreviations used throughout the paper. Appendix A.6 gives the improvement and success criteria and states exactly which tolerances remain uncalibrated. Appendix A.3 records benchmark sources, sample counts, split identifiers, and the train/method-validation/harness-validation/test partition. Appendix B records the two artifact-provenance exceptions we could not fully certify. Appendix C.6 gives the operational root-cause definitions, the evidence grades, and the label coverage and missingness status. We will release, in an anonymized archive at submission time and publicly on acceptance, the matched outcome ledger at the task-instance level for all executions summarized here, the frozen skill artifacts with their hashes, the split unique-identifier (UID) lists, the root-cause labels with their evidence grades, and the analysis scripts that regenerate every table and figure. The matched outcome is the reusable artifact of this paper, and we consider its release part of the contribution rather than supplementary material.

REFERENCES

Krishna Chaitanya Balusu. Agenttelemetry: A fault detection benchmark and toolkit for llm agent observability. In *Proceedings of the 3rd ACM International Conference on AI-Powered Software*, pp. 380–387, 2026.

Shraddha Barke, Arnav Goyal, Alind Khare, Avaljot Singh, Suman Nath, and Chetan Bansal. Agentrx: Diagnosing ai agent failures from execution trajectories. *arXiv preprint arXiv:2602.02475*, 2026.

George EP Box, J Stuart Hunter, William G Hunter, et al. Statistics for experimenters. *Design, innovation, and discovery*, 2:235–274, 2005.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. Why do multi-agent llm systems fail? *Advances in Neural Information Processing Systems*, 38, 2026.

Jixuan Chen, Jianzhi Shen, Haoqiang Kang, Zhi Hong, Qingyi Jiang, Soham Bose, Yiming Zhang, Leon Leng, Amit Vyas, Lingjun Mao, et al. Agentspec: Understanding embodied agent scaffolds through controlled composition. *arXiv preprint arXiv:2606.14674*, 2026a.

Shiqi Chen, Jingze Gai, Ruochen Zhou, Jinghan Zhang, Tongyao Zhu, Junlong Li, Kangrui Wang, Zihan Wang, Zhengyu Chen, Klara Kaleb, et al. Skillcraft: Can llm agents learn to use tools skillfully? *arXiv preprint arXiv:2603.00718*, 2026b.

Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. Trail: Trace reasoning and agentic issue localization. *arXiv preprint arXiv:2505.08638*, 2025.

Ulrich Faigle and Walter Kern. The shapley value for cooperative games under precedence constraints. *International Journal of Game Theory*, 21(3):249–266, 1992.

Shengda Fan, Xuyan Ye, Yupeng Huo, Zhi-Yuan Chen, Yiju Guo, Shenzhi Yang, Wenkai Yang, Shuqi Ye, Jingwen Chen, Haotian Chen, et al. Agentprocessbench: Diagnosing step-level process quality in tool-using agents. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 8823–8834, 2026.

Linyang He, Qiyao Yu, Hanze Dong, Baohao Liao, Xinxing Xu, Micah Goldblum, Jiang Bian, and Nima Mesgarani. Livemathematicianbench: A live benchmark for mathematician-level reasoning with proof sketches. *arXiv preprint arXiv:2604.01754*, 2026.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Sayash Kapoor, Benedikt Stroebl, Zachary S Siegel, Nitya Nadgir, and Arvind Narayanan. Ai agents that matter. *arXiv preprint arXiv:2407.01502*, 2024.

Zekun Li, Shinda Huang, Jiangtian Wang, Nathan Zhang, Antonis Antoniades, Wenyue Hua, Kaijie Zhu, Sirui Zeng, Chi Wang, William Yang Wang, et al. Sopbench: Evaluating language agents at following standard operating procedures and constraints. *arXiv preprint arXiv:2503.08669*, 2025.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Jiale Liu, Huajun Xi, Shaokun Zhang, Yifan Zeng, Tianwei Yue, Chi Wang, Jian Kang, Qingyun Wu, and Huazheng Wang. Who&when pro: Can llms really attribute failures in ai agents? *arXiv preprint arXiv:2607.09996*, 2026.

Ming Liu. More is not always better: Cross-component interference in llm agent scaffolding. *arXiv preprint arXiv:2605.05716*, 2026.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of multi-turn llm agents. *Advances in neural information processing systems*, 37:74325–74362, 2024a.

Zeyao Ma, Bohan Zhang, Jing Zhang, Jifan Yu, Xiaokang Zhang, Xiaohan Zhang, Sijia Luo, Xi Wang, and Jie Tang. Spreadsheetbench: Towards challenging real world spreadsheet manipulation. *Advances in Neural Information Processing Systems*, 37:94871–94908, 2024b.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pp. 681–699. Springer, 2020.

Subhrangshu Nandi, Arghya Datta, Rohith Nama, Udita Patel, Nikhil Vichare, Indranil Bhattacharya, Shivam Asija, Arushi Gupta, Giuseppe Carenini, Jing Xu, et al. Sop-bench: Complex industrial sops for evaluating llm agents. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 9604–9615, 2026.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*, 2026.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37: 126544–126565, 2024.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Lloyd S. Shapley. *A Value for N-Person Games*. RAND Corporation, Santa Monica, CA, 1952. doi: 10.7249/P0295.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 10131–10165, 2025.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. Openhands: An open platform for ai software developers as generalist agents. In *International Conference on Learning Representations*, volume 2025, pp. 65882–65919, 2025.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, et al. Skillopt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Luan Zhang, Ruochen Zhou, Dandan Song, Zhengyu Chen, Yuhang Tian, Jun Yang, Huipeng Ma, Chenhao Li, Guangyuan Feng, Xudong Li, et al. Harnesscompass: Guiding automatic harness evolution toward generalizable and effective agent harnesses. *arXiv preprint arXiv:2608.01918*, 2026.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, et al. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. *arXiv preprint arXiv:2505.00212*, 2025.

Yue Zhao. Catchbench: When can an agent failure be caught? *arXiv preprint arXiv:2608.22808*, 2026.

Xiaolin Zhou, Jinbo Liu, Li Li, Ryan A Rossi, and Xiyang Hu. Counterfactual trace auditing of llm agent skills. *arXiv preprint arXiv:2605.11946*, 2026.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, et al. Where llm agents fail and how they can learn from failures. *arXiv preprint arXiv:2509.25370*, 2025.

Kunlun Zhu, Xuyan Ye, Zhiguang Han, Yuchen Zhao, Bingxuan Li, Weijia Zhang, Muxin Tian, Xiangru Tang, Pan Lu, James Zou, et al. Agentdebugx: An open-source toolkit for failure observability, attribution, and recovery in llm agents. *arXiv preprint arXiv:2607.18754*, 2026a.

Pengyu Zhu, Lijun Li, Yaxing Lyu, Qianxin Luo, Jingyi Yang, Yi Liu, Tingfeng Hui, Xinyu Yuan, Li Sun, Sen Su, et al. A unified framework for the evaluation of llm agentic capabilities. *arXiv preprint arXiv:2605.27898*, 2026b.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, volume 2025, pp. 66602–66656, 2025.

A Experimental Details and Measurement Definitions

Design Matrix. We organize the matched study as $T \times M \times S \times H \times B$, use three repeats per available cell, and hold task instances, frozen Skill artifacts, and evaluators fixed. Figure 6 summarizes the matrix; the remainder of this section reports the intervention definitions, task-property annotations, execution ledger, protocol details, and uncertainty calibration.

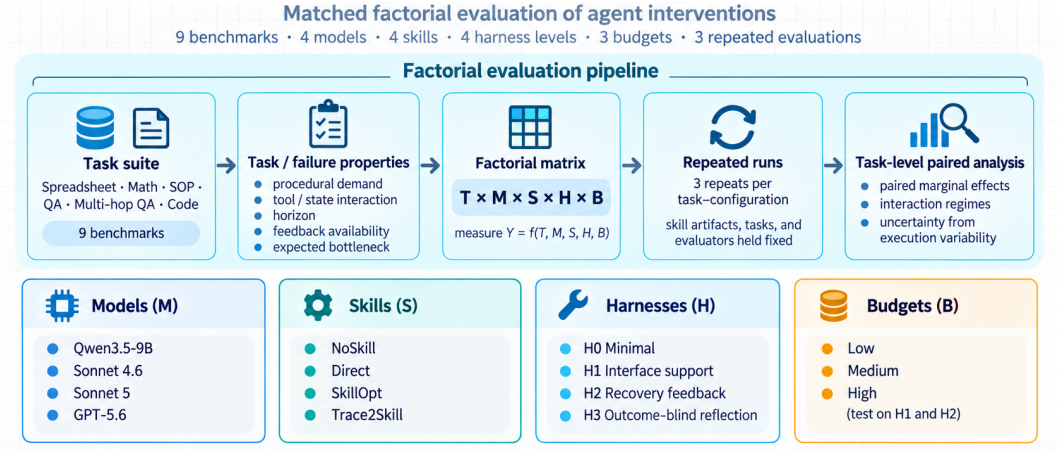


Figure 6: Matched factorial study. Nine benchmarks are crossed with four Models, four Skills, and H0–H3 Harness conditions; Low/Medium/High Budget sweeps are run on H1 and H2. Matched outcomes at the task-instance level support intervention, failure-mechanism, capability-boundary, and routing analyses.

Table 6: Skill construction conditions. The Outcome feedback column indicates whether scored outcomes directly drive optimization; Trace2Skill distills execution experience rather than primarily optimizing a procedure against validation scores. All constructed Skill artifacts are frozen before downstream evaluation. Locally constructed SKILLOPT and TRACE2SKILL artifacts use Qwen3.5-9B executions and their recorded method-validation procedures. Spreadsheet imports the released combined Trace2Skill package, whose complete upstream trajectory and validation provenance cannot be certified from the local archive. All frozen artifacts are transferred unchanged across target Models.

Condition	Rollout evidence	Outcome feedback	Construction	What it isolates
NOSKILL	No	No	None	Base model + harness capability
DIRECT	No	No	One-shot synthesis	Value of prior-only procedural guidance
SKILLOPT	Yes	Yes	Iterative optimization	Value of outcome-guided skill learning
TRACE2SKILL	Yes	Not primarily optimization-driven	Trajectory distillation	Value of experience-derived reusable procedures

A.1 INTERVENTION AXES

Model (M). We evaluate Qwen3.5-9B, Sonnet 4.6, Sonnet 5, and GPT-5.6. The cross-model axis tests capability-limited cases such as unsupported factual recall and whether the value of an external Skill or Harness transfers across model backbones.

Skill conditions. We compare four Skill conditions that differ in how procedural guidance is constructed and in the execution evidence available during Skill construction. NOSKILL provides no additional reusable procedure and serves as the reference condition. DIRECT uses a one-shot procedure synthesized from the task specification without access to rollout trajectories or outcome feedback, testing whether generic procedural guidance alone is useful. SKILLOPT (Yang et al., 2026) instead constructs the Skill through iterative outcome-guided optimization: candidate procedures are evaluated through task executions, and scored outcomes are used to revise the Skill toward higher validation performance. TRACE2SKILL (Ni et al., 2026) takes a trajectory-distillation approach, extracting transferable experience from prior execution traces and consolidating it into a reusable procedure. Thus, SKILLOPT is optimization-centric, whereas TRACE2SKILL is distillation-centric. All resulting Skill artifacts are frozen before downstream factorial evaluation, so measured differences reflect the supplied Skill rather than online adaptation during test execution.

The comparison therefore separates no supplied procedure (NOSKILL), one-shot procedure synthesis without execution evidence (DIRECT), validation-gated optimization from scored rollouts (SKILLOPT), and trajectory-level experience distillation into reusable procedures (TRACE2SKILL). Figure 7 summarizes the three construction pathways.

Harness (H). A Harness changes the information and actions available during execution rather than supplying a reusable task policy before execution. H0 uses the minimal task wrapper; H1 standardizes the answer or tool interface (Patil et al., 2024); H2 additionally exposes executable observations or errors and permits bounded recovery (Yao et al., 2022); and H3 adds one outcome-blind revision of the response or trajectory (Madaan et al., 2023). Figure 8 illustrates this progression, and Table 7 makes the differences explicit.

H1 therefore targets interface and answer-contract compatibility, whereas H2 jointly changes both failure observability and the opportunity to act on that evidence. H2 is deliberately composite, so an H2 gain cannot be attributed separately to error exposure or retry. H3 tests whether a completed response or trajectory can be improved through one additional outcome-blind revision.

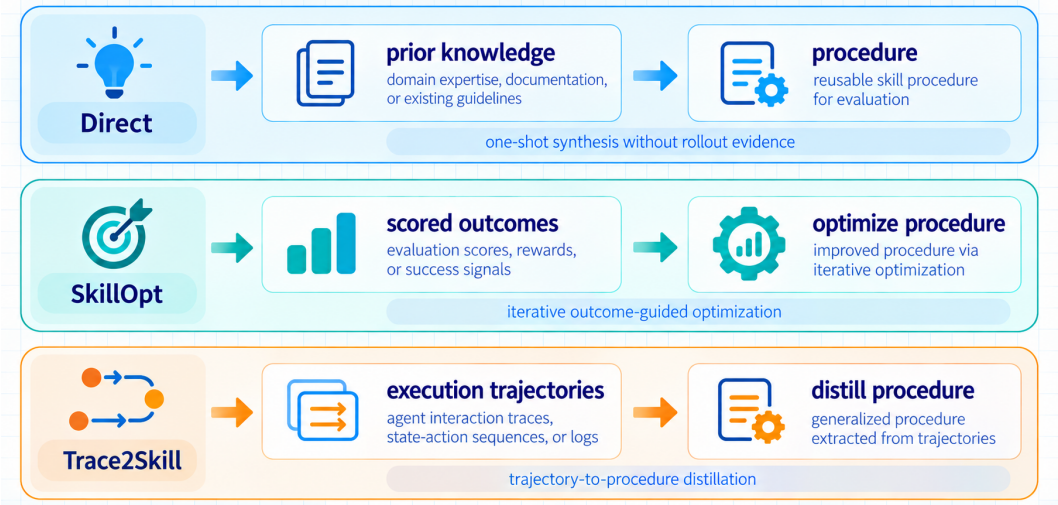


Figure 7: Skill construction pathways. DIRECT synthesizes a procedure from prior knowledge without rollout evidence; SKILLOPT optimizes a procedure using scored outcomes; and TRACE2SKILL distills a reusable procedure from execution trajectories.

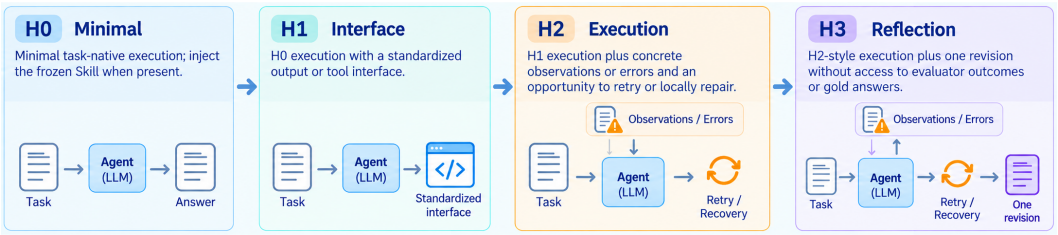


Figure 8: Harness intervention design. H0 provides minimal task-native execution; H1 adds a standardized output or tool interface; H2 adds concrete observations or errors and bounded retry or local repair; and H3 adds one outcome-blind revision without access to evaluator outcomes or gold answers.

Table 7: Harness interventions. "Added observations/errors" refers to evidence introduced by the Harness beyond the task's native input. H3 starts from H2 and adds a distinct revision operation; it is not simply a larger H2 recovery budget. A privileged diagnostic branch that exposes evaluator-side information is excluded from every analysis and from the Default matrix.

Harness	Execution mode	Standardized interface	Added observations/errors	Bounded recovery	Outcome-blind revision
H0 Minimal	Minimal task-native execution; inject the frozen Skill when present.	No	No	No	No
H1 Interface	H0 execution with a standardized output or tool interface.	Yes	No	No	No
H2 Execution	H1 execution plus concrete observations or errors and an opportunity to retry or locally repair.	Yes	Yes	Yes	No
H3 Reflection	H2-style execution plus one revision without access to evaluator outcomes or gold answers.	Yes	Yes	Yes	One pass

Budget (B). Low, Medium, and High resource ceilings are crossed only with H1 and H2; H0–H3 are also evaluated under their historical Default configurations. The sweep jointly raises completion-token, turn, and timeout ceilings. It therefore tests whether a trajectory is resource-feasible and where performance saturates, rather than isolating token-only inference scaling; excessive ceilings can also increase deployment cost without improving quality. Table 8 reports the parameter settings. These

are resource allowances, not observed consumption or model reasoning-effort settings. Default is a historical benchmark-specific configuration and is not relabeled Medium.

Table 8: Nominal resource ceilings in the Low/Medium/High sweep.

Budget	Completion tokens	Turns	Timeout (s)
Low	4,096	3	120
Medium	16,384	30	360
High	32,768	60	720

The sweep jointly changes token, turn, and timeout ceilings, so its effects cannot be attributed to tokens alone. Historical UCSB Low restricts workflow feasibility; the separately recorded protocol-preserving corrected Low reduces only the token cap and retains its reference turns, tools, and timeout. Historical and corrected endpoints must be identified separately and are not interchangeable.

Matched intervention and root-cause protocol. For each intervention, we compare matched task-instance–model configurations while changing one axis and holding the rest of the execution condition fixed. For the failure analysis, we assign 2,225 failed source executions to eight root-cause categories using the earliest decisive cause supported by the trajectory; Table 22 defines those categories. The source label is frozen before matched target outcomes are inspected. Benchmark-specific deterministic checks support the label when possible; cases requiring additional interpretation use an AgentRx-inspired constraint-grounded judge.

Two properties of this labeling limit what it can support. The judge is Qwen3.5-9B, which is also one of the four evaluated models and the model whose training executions produced every SKILLOPT and TRACE2SKILL artifact, so on one quarter of the matrix it attributes causes to its own failures. And we report *no* inter-annotator agreement of any kind. We therefore treat model-assisted labels as hypotheses throughout and mark them as such in every table; Section C.6 gives the definitions, the evidence boundary, and the reliability study that would be needed to promote them to measurements.

Measurement validity. An unchanged-configuration repeat comparison (A/A) compares repeated executions of the same task instance to measure variation without an intervention. Noise-qualified improvement must exceed the corresponding A/A tolerance, whereas success requires reaching the benchmark's native success threshold. Section A.6 distinguishes these criteria and documents label coverage, budget limits, and the calibration status of the reported descriptive results.

A.2 PER-BENCHMARK CAPABILITIES

Table 9: Human-authored descriptive task properties and task-structure intervention hypotheses used to diversify benchmark selection. These annotations are not trajectory-derived root labels and are not used as evidence in the root-cause analysis. Proc. = procedural; Fb. = feedback; Med. = medium. "Expected intervention" records a design hypothesis, not an observed winner.

Benchmark	Proc. demand	Tool/ state	Horizon	Fb. demand	Fb. avail.	Expected bottleneck	Expected intervention
Spreadsheet	Med.	High	Med.	High	High	Execution/tool	H2; H3
LiveMath	Low	Low	Med.	Low	Low	Reasoning/decision rule	Skill; Model
UCSB SOP	High	High	Med.	High	High	Procedural	Skill; H2
Amazon SOP	High	High	High	High	High	Execution/tool; horizon	H2; H3; Budget
SimpleQA	Low	Low	Low	Low	Low	Factual knowledge/recall	Model
SQuAD	Low	Low	Low	Low	High	Evaluation/answer contract	H1
Hotpot Context	Low	Low	Med.	Low	Med.	Evaluation/answer contract	H1
Hotpot FullWiki	Med.	High	High	High	Med.	Horizon/termination	Skill; H2; Budget

Table 9 reproduces the compact benchmark-property summary here for local reference. These columns are human-authored design annotations used to diversify benchmark selection before any execution; they are not trajectory-derived root labels and enter no root-cause analysis, so the *Expected bottleneck* and *Expected intervention* entries can be read against the measured results without circularity. The primary capability each benchmark exercises is the following.

- **SpreadsheetBench** (Ma et al., 2024b): multi-step, state-changing spreadsheet edits issued through a tool; stresses execution and tool-use reliability under an evolving workbook state.
- **LiveMathematicianBench** (He et al., 2026): recent multiple-choice mathematical reasoning; stresses reasoning together with a reusable answer-selection regularity, namely a recurring meta-option that is systematically correct when present.
- **UCSB SOPBench** (Li et al., 2025): long, constraint-sensitive standard operating procedures executed against state; stresses procedural compliance, step ordering, and constraint verification.
- **Amazon SOP** (Nandi et al., 2026): a long, state-changing inspection workflow with many tool calls; stresses execution reliability and path feasibility over a long horizon.
- **SimpleQA** (Wei et al., 2024): short-form factual questions with no supplied evidence; stresses parametric factual recall and calibrated abstention.
- **SQuAD** (Rajpurkar et al., 2016): extractive question answering with the evidence passage supplied; stresses answer-contract and interface compliance rather than retrieval.
- **HotpotQA, provided-context** (Yang et al., 2018): multi-hop questions with gold and distractor paragraphs supplied; stresses multi-hop evidence integration under a fixed answer contract.
- **HotpotQA, full-wiki** (Yang et al., 2018): the same questions with only a frozen Wikipedia index; adds retrieval-policy, grounding, and horizon demands on top of evidence integration.
- **BigCodeBench-Hard** (Zhuo et al., 2025): program synthesis executed against held-out tests; stresses code generation with executable feedback. It is retained as appendix-only evidence because its evaluator environment is not fully comparable across Models.

A.3 BENCHMARK SAMPLES AND FROZEN SPLITS

Table 10 reports every sample used for Skill construction, method selection, Harness qualification, and final evaluation. Method validation selects or accepts Skill artifacts; Harness validation qualifies execution code and interfaces without selecting Skill content; the locked test split is used only for final evaluation. We provide machine-readable `split_manifest.json` files and ordered `ids.json` files for every role, including the split seed, source revision, and SHA-256 hashes. Where redistribution is permitted, the archive also provides the corresponding `items.json`. These files reproduce the exact task-instance membership and ordering used in the experiments and permit direct overlap checks across roles.

SpreadsheetBench. We use the verified 400-instance release at the frozen source revision. Its 80/40/280 source train/validation/test partition is preserved, with the 40 validation instances divided

Table 10: Benchmark sample and split accounting. `Used` is the number of distinct task instances assigned to the four experimental roles, before multiplying by Models, Skills, Harnesses, Budgets, or repeats. Hotpot Context and FullWiki are two execution settings over the same UID split and are therefore listed separately but do not represent disjoint questions. All project-defined sampling uses seed 42. For SQuAD and HotpotQA, the two source-pool numbers denote the official training and development/validation pools. Amazon has 12 additional audit-reserve examples that are excluded from the four experimental roles. BigCodeBench is appendix-only because its evaluator environment is not fully comparable across Models.

Benchmark	Source pool	Train	Method val.	Harness val.	Test	Used
SpreadsheetBench	400	80	24	16	280	400
LiveMathematicianBench	177	35	10	8	124	177
UCSB SOP	830	70	20	15	175	280
Amazon SOP	112	30	10	10	50	100
SimpleQA	4,326	70	20	15	300	405
SQuAD	87,599+10,570	70	20	15	300	405
Hotpot Context	90,447+7,405	70	20	15	300	405
Hotpot FullWiki	90,447+7,405	70	20	15	300	405
BigCodeBench-Hard	148	70	20	15	43	148

into 24 method-validation and 16 Harness-validation instances. The 280-instance test set remains locked. Spreadsheet has an artifact-provenance exception described in Section B.

LiveMathematicianBench. We deterministically partition the 177 questions from four monthly releases with a 2:1:7 train/validation/test ratio and seed 42, yielding 35/18/124 task instances. The 18 validation questions are divided in source order into ten method-validation and eight Harness-validation instances.

UCSB SOP. From the 830 released SOPBench instances, we freeze 280 examples using deterministic balancing over domain, path length, constraint complexity, tool-call demand, outcome type, and SOP branch. The split contains 70 training, 35 validation, and 175 locked test examples, with 25 test examples from each of seven domains. The validation set is partitioned into 20 method-validation and 15 Harness-validation examples.

Amazon SOP. The aircraft-inspection configuration contains 112 released examples. Because the official release is test-only, we define project roles with seed 42: 30 training, ten method-validation, ten Harness-validation, and 50 test examples. The remaining 12 form an audit reserve and are not included in the factorial evaluation.

SimpleQA. The official release contains 4,326 test-only questions. We use seed 42 and stratify by topic and answer type to select 405 disjoint examples: 70 training, 20 method-validation, 15 Harness-validation, and 300 locked test questions. The remaining 3,921 questions are not used.

SQuAD. Training and validation roles are sampled from the 87,599-question official training set, using one question per development article and stratification by question type and answer-length bucket. The 70/20/15 training, method-validation, and Harness-validation examples are article-disjoint. The 300 locked test questions come from the 10,570-question official development set and cover all 48 development articles.

Hotpot Context. We sample 405 question UIDs with seed 42 from the official distractor configuration: 70 training, 20 method-validation, 15 Harness-validation, and 300 test questions. This setting supplies the official gold and distractor paragraphs to the agent.

Hotpot FullWiki. FullWiki reuses exactly the same 405 UIDs and role assignments as Hotpot Context, but supplies only the question and permits retrieval from a frozen local 2017 Wikipedia index; gold titles are not used to construct that index. Exact UID pairing makes differences between the two settings attributable to evidence access rather than question sampling.

BigCodeBench. We use all 148 task instances in BigCodeBench-Hard v0.1.4 and define disjoint roles of 70/20/15/43 with seed 42. Agent-visible prompts and evaluator-private solutions/tests are

stored separately. BigCodeBench is retained as appendix evidence because its evaluator environment is not fully comparable across Models.

Evaluation and aggregation. Every benchmark retains its native evaluator and success semantics. Native scores are never pooled directly across benchmarks; cross-benchmark summaries use equal-weight macro averaging so that large test sets do not dominate. Exact evaluator versions and instance UIDs are tied to the released split manifests and execution ledger.

A.4 EXECUTION LEDGER

Table 11 states the nominal size of the full nine-benchmark study. The eight primary comparable benchmarks contribute 1,829 test instances, and appendix-only BigCode contributes 43, for 1,872 test instances in total. A small number of cells are unavailable rather than executed, most notably the protocol-preserving Low endpoints for the three API models on UCSB SOP, so the realized total is slightly below the nominal one.

Table 11: Nominal execution ledger over all nine benchmarks (1,872 test instances). BigCode is retained as appendix-only evidence; derived main-analysis cohorts use the eight comparable benchmarks. [a]Four Models × four Skills × H0–H3 × three repeats. [b]Four Models × four Skills × H1/H2 × three Budgets × three repeats. The same formulas include the 43 BigCode test instances in the nominal nine-benchmark count; BigCode remains excluded from primary comparative analyses because its evaluator environment is not fully comparable across models. Qwen3.5-9B was served locally, so its executions have no per-token API price (Table 35) and no comparable latency or dollar ledger. Wall-clock and accelerator hours were not logged uniformly across the campaign and we do not report them.

Quantity	Count
Primary comparable test instances	1,829
BigCode appendix test instances	43
All test instances	1,872
Default matrix executions[a]	359,424
Budget sweep executions[b]	539,136
Total nominal agent executions	898,560
Matched H1→H2 pairs at Default	87,792
Matched Skill×Harness interaction units	21,948
Repeat-1 common-support source failures	3,939
Root-labeled source failures	2,225

A.5 EVALUATION METRICS AND STATISTICAL ABBREVIATIONS

We distinguish benchmark performance, matched intervention response, success, interaction, predictive discrimination, and routing cost. Let $Y_i(c) \in [0,1]$ be the native evaluator outcome for task instance i under configuration c; tables and figures report $100Y$ unless stated otherwise. A cell mean averages $Y_i(c)$ over its task instances and three repeated executions. A matched intervention change is

$$\Delta_{i,a} = Y_i(c+a) - Y_i(c), \tag{1}$$

where the task instance and all axes other than intervention a are held fixed. Positive observed change, change exceeding execution noise, and crossing the benchmark's success threshold are separate events; Section A.6 defines the latter two and states their calibration status.

For matched marginal summaries, let $\bar{Y}_s$, $\bar{Y}_h$, $\bar{Y}_b$, and $\bar{Y}_m$ denote balanced marginal means. We report four *within-axis observed envelopes*, each clipped at zero and each defined against its own reference cell:

$$\Delta_S^+ = \max\left(0, \max_{s \neq \text{NoSkill}} \bar{Y}_s - \bar{Y}_{\text{NoSkill}}\right), \qquad \Delta_H^+ = \max\left(0, \max_{h \in \{H1,H2,H3\}} \bar{Y}_h - \bar{Y}_{H0}\right),$$
$$\Delta_B^+ = \max\left(0, \max_{b \in \{\text{Low,Medium,High}\}} \bar{Y}_b - \bar{Y}_{\text{Low}}\right), \qquad \Delta_M = \max_m \bar{Y}_m - \min_m \bar{Y}_m. \tag{2}$$

Table 12: Evaluation metrics used throughout the paper. Native outcomes are displayed as percentages unless noted otherwise.

Metric	Definition	Interpretation boundary
Native performance	Benchmark-native evaluator outcome $Y_i(c)$; a cell mean averages task instances and three repeats.	Native scores are combined across benchmarks only by equal-weight macro averaging.
Matched change	$\Delta_{i,a} = Y_i(c+a) - Y_i(c)$ with the task instance and non-target axes fixed.	An observed positive change is not automatically larger than execution noise.
Within-axis envelope	Δ_S^+, Δ_H^+, Δ_B^+, or Δ_M in Eq. 2.	A selected maximum or span, not an unbiased effect or an action-count-matched ranking.
Skill–Harness interaction	Difference-in-differences $I_{S,H}$ in Eq. 3.	The four named patterns use a prespecified five-point descriptive tolerance; they are not A/A-calibrated significance tests.
Intervention opportunity	Fraction of failed source task instances with a strictly positive matched target change; standardized summaries give equal weight to the candidate actions being compared.	Measures observed rescue opportunity, not noise-qualified improvement or deployable predictability.
Success crossing	Target outcome reaches the benchmark-native threshold τ_b after a failed source outcome.	Partial score improvement need not constitute success.
Rescue, preservation, regression, persistent failure	Matched binary transitions $0 \to 1$, $1 \to 1$, $1 \to 0$, and $0 \to 0$, respectively. Net gain is rescues minus regressions over all matched units.	Rescue must be interpreted jointly with preservation and regression.
Predictive fit and discrimination	R^2 is the coefficient of determination under LOO held-cell reconstruction; AUPRC and AUROC measure ranking performance, and log loss evaluates probabilistic predictions.	Grouped or leave-one-benchmark-out splits are used where stated; none is a causal estimand.
Routing quality and cost	Equal-weight macro performance and mean full-chain tokens, including required probes and restarts. One policy strictly dominates another if it has higher quality and lower cost.	The Observed Oracle uses matched outcomes and measures headroom rather than deployable performance.
Uncertainty	Task-instance/UID cluster bootstrap confidence intervals preserve all configurations for a resampled instance; paired sign-flip or stratified permutation tests are used where specified.	Sampling uncertainty does not substitute for A/A execution-noise calibration.

Each envelope is a maximum over a different candidate set and is therefore selection-biased; the four values are descriptive within-axis summaries rather than directly comparable treatment effects. Δ_M is a best-to-worst span because no Model is a natural baseline. For Skill $\times$ Harness, the matched interaction contrast is

$$I_{S,H} = [Y(S, H2) - Y(0, H2)] - [Y(S, H1) - Y(0, H1)]\,. \tag{3}$$

A/A denotes an unchanged-configuration repeat comparison; API, application programming interface; AUPRC, area under the precision–recall curve; AUROC, area under the receiver-operating-characteristic curve; BH, Benjamini–Hochberg; CI, confidence interval; LOO, leave-one-cell-out; LOBO, leave-one-benchmark-out; OOF, out-of-fold; PRESS, prediction sum of squares; SD, standard deviation; SOP, standard operating procedure; and UID, unique identifier. The abbreviation pp denotes percentage points, and NA or N/A denotes unavailable or not applicable as specified by the table.

Table 12 collects the remaining metrics used in the main text and detailed analyses. These definitions fix the denominator and interpretation of each quantity; they do not turn descriptive matched comparisons into randomized causal effects.

Abbreviations.

A.6 MEASUREMENT VALIDITY AND OPERATIONAL THRESHOLDS

A.6.1 MEASUREMENT FLOORS AND ANSWER-CONTRACT EFFECTS

Table 13: Benchmark-level within-axis observed envelopes in percentage points using Equation 2. Blue marks the numerically largest envelope in each row. Because the four columns use different reference cells and different candidate-set sizes ($|S| = 3$, $|H| = 3$, $|B| = 3$, $|M| = 4$) and each is a maximum over its set, this mark is a reading aid and *not* an action-count-matched ranking of intervention axes. Rows marked ‡ contain native-score floor cells, so their Δ_H^+ mixes answer-contract specification with execution support (Appendix A.6.1). †UCSB is unavailable because matching protocol-preserving Low endpoints do not exist for the three API models.

Benchmark	Δ_S^+	Δ_H^+	Δ_B^+	Δ_M
Spreadsheet	0.00	5.54	4.18	24.33
LiveMath	8.15	0.79	2.67	22.26
UCSB SOP	5.55	0.00	NA†	10.42
Amazon SOP‡	0.00	48.03	81.13	53.10
SimpleQA	0.13	0.00	0.10	60.85
SQuAD‡	17.13	44.34	0.24	11.88
Hotpot Context‡	1.71	48.18	0.00	9.57
Hotpot FullWiki‡	10.58	3.92	7.26	6.99

Table 14: Root-cause composition of the 2,225 labeled NoSkill+H1+Medium source failures. Column order follows the taxonomy: Procedural, Reasoning, Execution/tool, State/grounding, Horizon/termination, Factual knowledge/recall, Abstention/calibration, Evaluation/answer contract. Labeled n is the benchmark denominator and sums to the same 2,225 source failures reported in Table 23. *Model-assisted or provisional answer-level hypotheses, produced by the Qwen3.5-9B judge rather than by deterministic evidence, and carrying no reliability estimate; Appendix C.6 describes the judge and its evidence boundary. Hotpot Context and Hotpot FullWiki have one and ten labeled failures here, so their composition explains nothing about their benchmark-wide results.

Benchmark	n	Proc.	Reas.	Exec.	State	Horiz.	Fact.	Abst.	Contr.
Spreadsheet	588	0.00	0.00	99.83	0.00	0.17	0.00	0.00	0.00
LiveMath	305	0.33	73.44*	0.00	26.23*	0.00	0.00	0.00	0.00
UCSB SOP	266	78.57	0.00	7.52	13.91	0.00	0.00	0.00	0.00
Amazon	113	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
SimpleQA	783	0.00	0.00	0.00	0.00	0.00	74.84*	24.78*	0.38
SQuAD	159	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
Hotpot Context	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
Hotpot FullWiki	10	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00

Before interpreting any axis mechanistically we isolate a confound that inflates the largest contrasts in the matrix. Of the 512 Default cells, 16 sit at a native-score floor below 2.0 on the 0–100 scale, and their distribution is not uniform (Table 15). Twelve occur at H0, the one level supplying no standardized output or tool interface; the other four are all on Amazon SOP, where DIRECT on Qwen3.5-9B returns exactly 1.7 at each of H0–H3. A score identical to one decimal across four harness levels is one reproducible failure signature, not four graded outcomes.

Table 15: Native-score floor cells in the 512-cell Default matrix, defined as a three-repeat mean below 2.0 on the 0–100 scale. Δ_H^+ is that benchmark's harness envelope from Table 13. Spreadsheet, LiveMath, UCSB SOP, and SimpleQA contain no floor cells and are omitted. The three benchmarks with the largest harness envelopes are exactly the three with the most floor cells. We therefore do not read those envelopes as evidence that execution support repairs residual reasoning or planning failures.

Benchmark	Floor cells	At H0	Configurations at floor	Δ_H^+
Amazon SOP	6	2	Qwen+DIRECT at H0–H3; Sonnet 5+DIRECT at H1; Sonnet 5+SKILLOPT at H0	48.03
SQuAD	3	3	Sonnet 4.6+NOSKILL; Sonnet 4.6+SKILLOPT; Sonnet 5+NOSKILL	44.34
Hotpot Context	6	6	Qwen+SKILLOPT; Sonnet 4.6+NOSKILL/SKILLOPT/TRACE2SKILL; Sonnet 5+SKILLOPT; GPT-5.6+SKILLOPT	48.18
Hotpot FullWiki	1	1	Qwen+DIRECT	3.92

Two consequences follow. First, the three largest harness envelopes in Table 13, Amazon SOP at 48.03, Hotpot Context at 48.18, and SQuAD at 44.34, belong to exactly the three benchmarks carrying the most floor cells. For SQuAD and Hotpot Context every floor cell is at H0, and SQuAD's labeled failures are 100% Evaluation/answer contract (Table 14). The parsimonious reading is that H0 leaves the answer contract unspecified, so the native scorer credits little and H0→H1 largely purchases a satisfiable answer format. That is a real effect but a claim about *contract specification* rather than about executable feedback repairing a residual bottleneck, so we label these contrasts contract-level and exclude them from mechanism-level interpretation. Second, a Skill can collapse a working configuration to the floor: on Hotpot Context, GPT-5.6 with SKILLOPT scores 0.0/13.6/12.2/12.2 across H0–H3 against 22.6/62.0/61.9/61.3 for NOSKILL, a persistent deficit near 49 points more consistent with the skill overriding the answer contract than with graded interference.

We have quarantined the affected claims without settling the question, since separating a format or schema violation from a capability failure requires re-scoring saved trajectories under a contract-relaxed evaluator that we have not run. Until then those three harness envelopes are upper bounds of mixed composition, and the two Amazon SOP interaction examples in Table 20, marked § there, are floor-adjacent illustrations rather than calibrated magnitudes. This is the largest threat to the corresponding quantitative claims, and every affected result is flagged where it appears rather than only here.

A/A and improvement. All scores below are on the native $[0, 1]$ scale. An A/A comparison changes neither task instance nor configuration: for two executions $r \neq r'$, define $D_{i,c}^{AA} = Y_i^{(r)}(c) - Y_i^{(r')}(c)$. For a calibrated analysis at the task-instance level, define the benchmark tolerance as $\delta_b = Q_{0.95}(|D_{i,c}^{AA}|)$ using matched repeated executions from a prespecified, configuration-matched calibration set. A repeat-averaged contrast requires an A/A distribution of repeat-averaged differences with the same repeat count; a benchmark-average noise floor must not be substituted for a task-instance tolerance. Calibration should precede inspecting intervention contrasts. The noise-qualified improvement event and success-crossing event are distinct:

$$I_{i,a} = \mathbf{1}[Y_i(a) - Y_i(c) > \delta_b]\,, \qquad K_{i,a} = \mathbf{1}[Y_i(a) \geq \tau_b]\,. \tag{4}$$

For binary single-execution scores, observed improvement is exactly $0 \to 1$. This is not automatically evidence beyond execution noise: if the task-instance A/A quantile equals 1, a single binary contrast cannot clear it and repeated evidence is required. For fractional Amazon scores, any positive difference is only an observed gain; a noise-qualified gain must clear the calibrated fractional-score tolerance.

Calibration status. The current opportunity tables report observed positive changes, not A/A-qualified gains. The saved historical A/A analyses concern behavioral metrics and do not provide a matched performance tolerance for this common-support cohort. Accordingly, numerical δ_b values remain uncalibrated for Spreadsheet, LiveMath, UCSB SOP, Amazon SOP, SimpleQA, SQuAD, Hotpot Context, and Hotpot FullWiki; no value is imputed as zero. Equation 4 specifies the calibration criterion and does not retroactively certify the reported counts. Bootstrap confidence intervals quantify sampling uncertainty but do not replace an execution-noise calibration.

Repeat design. We use three repeated executions per cell to capture material execution-level nondeterminism. In a controlled Spreadsheet H2 diagnostic comprising 3,360 trajectories (two Skill arms $\times$ three request seeds $\times$ three repeats $\times$ 280 task instances), the paired Skill effect had a pooled within-seed repeat SD of 2.26 points and an estimated request-seed SD of 1.56 points; fixed-seed repeat variation accounted for 68% of the estimated variance. This diagnostic is limited to one benchmark and two Skill arms, so it motivates repeated execution but does not establish a universal variance decomposition.

Interaction tolerance. The reported four-pattern prevalence uses a fixed descriptive tolerance $\epsilon = 0.05$ (five native-score points) applied to repeat-averaged four-cell contrasts. This threshold was not estimated from A/A. A noise-qualified interaction analysis would additionally require a matched null for the four-cell interaction contrast, not merely reuse a two-cell improvement tolerance. The existing prevalence therefore characterizes thresholded response patterns rather than effects certified to exceed A/A noise.

Label coverage and success. The repeat-1 common-support cohort contains 3,939 source failures; 2,225 have assigned labels (56.5%) and 1,714 remain unassigned (43.5%). Root-conditioned percentages describe the assigned subset. Evidence-backed labels and model-assisted hypotheses retain their distinct status. Capability crossings use $\tau_b = 1$ for all eight benchmarks: binary task instances must pass their native scorer, and Amazon must attain full field accuracy. Implementations allow 10^{-9} numerical slack at 1. Partial Amazon improvements need not cross this strict full-success threshold. BigCode is excluded from this common-support analysis.

B SKILL PROVENANCE, SPLIT ISOLATION, AND TRANSFER SENSITIVITY

Construction and selection. NoSkill supplies no learned artifact. Direct synthesizes a procedure once from a frozen catalog, without rollout collection or a validation gate. SkillOpt uses training rollouts for iterative edits, and locally generated Trace2Skill distills training-trajectory lessons under its recorded method-validation procedure. The protocol reserves method validation for skill selection and separates harness qualification, which qualifies execution code and interfaces rather than selecting skill content. Frozen packages are reused unchanged across target models instead of regenerated for each target. Holding the artifact fixed does not remove its potential alignment to the construction model.

UID and artifact evidence. The anonymized supplementary archive contains a provenance inventory recording exact archived UIDs, split-file hashes, within-directory overlap checks, and saved freeze declarations, together with the cross-model artifact bundle from which every frozen skill package is imported. UCSB training and test UIDs come from the archived 280-item SOP split, whose original validation set is partitioned into method and harness validation. LiveMath uses the archived ten-item method-validation split; Hotpot stores its Context and FullWiki views beneath a common split root. The protocol excludes test instances from generation, selection, patching, and acceptance. Disjoint UID lists and saved freeze declarations support this rule but are not a complete audit of every historical generation call. Hash-freezing specifies unchanged cross-model import; the inventory does not independently compare artifact bytes in every provider execution.

Spreadsheet exception. The archived validation manifest records Direct's 104-item catalog as 80 training plus 24 method-validation items, so Direct is not train-only here. SkillOpt uses the 80/24 method split. Trace2Skill imports the authors' released combined spreadsheet package rather than regenerating it from the current Qwen3.5-9B training split. The archived provenance records freeze entrypoint and package hashes before evaluation. The upstream release's complete trajectory UID provenance is not established by these manifests; test isolation of its original construction cannot be certified from the local record alone.

Qwen-excluded sensitivity. We retain Sonnet 4.6, Sonnet 5, and GPT-5.6 on the same task instances and frozen artifacts. Skill gains average matched NoSkill contrasts over H0–H3 at Default and over retained models. Interaction prevalence retains the original H1/H2 classifier, three-repeat means, $\epsilon = 0.05$, and equal benchmark weights. BigCode is excluded. Intervals use 2,000 instance-UID cluster bootstrap draws with seed 42, preserving all configurations per sampled task instance. Removing Qwen reduces interaction units from 21,948 to 16,461. No new model executions are used.

Table 17: Interaction prevalence with and without Qwen. Original H1/H2 rules, $\epsilon = 0.05$, complete three-repeat cells, and benchmark-equal weighting are retained.

Regime	Four models (%)	API only [95% CI] (%)
Absorption	3.77	2.79 [2.28, 3.31]
Activation	2.35	2.02 [1.65, 2.46]
Complementarity	1.48	1.27 [0.90, 1.70]
Interference	6.46	5.13 [4.59, 5.67]

FullWiki, SQuAD, LiveMath, and UCSB retain positive gains for their responsive skills. LiveMath's Trace2Skill gain contracts from +8.15 to +3.99 points. Amazon also changes: its marginal SkillOpt gain moves from -0.09 to +3.01 points, illustrating dependence on model composition. All four named patterns persist with lower combined macro prevalence. These results establish transfer to the tested API targets, not construction-model neutrality or a causal explanation of alignment. Intervals reflect task sampling, not sampling of model families.

Table 16: Skill transfer sensitivity at Default, averaged over H0–H3. Gains are paired against NoSkill; API-only excludes Qwen. Intervals resample task UIDs within benchmark (2,000 draws), retaining all models and harnesses for each task.

Benchmark	Skill	Four-model gain (pp)	API-only gain [95% CI] (pp)
Amazon	Direct	-13.29	+2.20 [+0.33, +4.09]
Amazon	SkillOpt	-0.09	+3.01 [+1.71, +4.25]
Amazon	Trace2Skill	-2.92	-0.25 [-1.70, +1.15]
Hotpot Context	Direct	+1.63	+2.75 [+1.92, +3.65]
Hotpot Context	SkillOpt	-13.48	-14.26 [-15.64, -12.81]
Hotpot Context	Trace2Skill	+1.71	+1.75 [+1.06, +2.44]
Hotpot FullWiki	Direct	-3.02	-0.68 [-1.99, +0.61]
Hotpot FullWiki	SkillOpt	-4.67	-7.21 [-8.75, -5.75]
Hotpot FullWiki	Trace2Skill	+10.58	+10.93 [+9.17, +12.61]
LiveMath	Direct	-1.50	-1.16 [-3.07, +0.83]
LiveMath	SkillOpt	+2.20	+1.14 [-1.52, +3.79]
LiveMath	Trace2Skill	+8.15	+3.99 [+1.07, +6.83]
SQuAD	Direct	+12.26	+10.50 [+9.02, +11.88]
SQuAD	SkillOpt	+6.86	+4.49 [+3.30, +5.65]
SQuAD	Trace2Skill	+17.13	+15.61 [+14.19, +17.10]
SimpleQA	Direct	-1.83	-2.56 [-3.81, -1.31]
SimpleQA	SkillOpt	+0.12	-0.20 [-1.07, +0.65]
SimpleQA	Trace2Skill	-0.12	-0.28 [-0.96, +0.45]
Spreadsheet	Direct	-1.64	-2.85 [-3.81, -1.86]
Spreadsheet	SkillOpt	-4.87	-5.58 [-6.84, -4.37]
Spreadsheet	Trace2Skill	-6.23	-7.49 [-9.00, -5.91]
UCSB SOP	Direct	-0.04	-1.02 [-2.38, +0.25]
UCSB SOP	SkillOpt	+5.55	+3.59 [+2.16, +5.05]
UCSB SOP	Trace2Skill	+1.89	-0.17 [-1.16, +0.86]

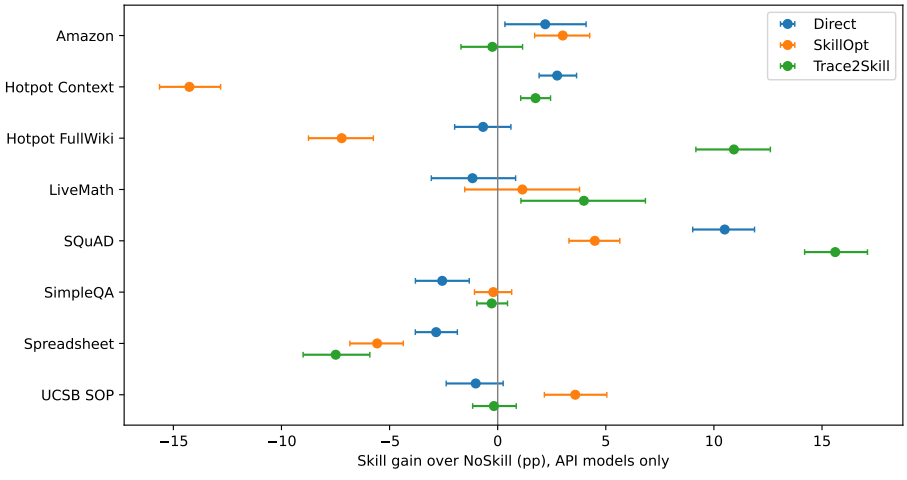


Figure 9: Skill gains in three API models after excluding Qwen. Points and 95% task-cluster bootstrap intervals use the Default H0–H3 support in Table 16. Excluding Qwen preserves several positive effects but does not remove artifact-construction bias.

C Detailed Analyses for the Results

This section mirrors the order of Section 4. It first documents the full factorial response surface and the two observations derived from it, then provides the detailed estimands, empirical evidence, uncertainty analyses, and interpretation boundaries for RQ1–RQ5.

C.1 No Universally Optimal Agent Configuration

C.1.1 Complete Default-Matrix Reference

Table 18: Combined Default-budget Skill × Harness performance matrix. Rows follow the Benchmark × Skill layout; columns group each Model into H0–H3 Harness conditions. Each entry is the mean benchmark-native performance on a 0–100 scale over three complete repeats; the privileged H4 branch is excluded, and BigCode is omitted because its evaluator environment is not fully comparable across Models. **Bold blue** marks the best cell in each Benchmark × Skill row, a red-shaded cell the best in each Benchmark × Model block, and a box the best in the whole benchmark; ties at one-decimal precision are all marked and the styles combine.

Benchmark	Skill / contrast	Qwen3.5-9B				Sonnet 4.6				Sonnet 5				GPT-5.6			
		H0	H1	H2	H3	H0	H1	H2	H3	H0	H1	H2	H3	H0	H1	H2	H3
Spreadsheet	NoSkill	12.5	9.3	24.9	30.4	48.6	49.2	51.1	**53.1**	33.5	31.9	34.5	34.2	35.1	35.0	36.0	37.5
	Direct	14.3	13.3	28.6	28.8	45.8	43.3	45.8	**48.9**	30.1	31.4	34.0	28.2	33.7	33.0	35.0	36.0
	SkillOpt	8.0	9.3	22.5	26.2	34.2	39.6	**40.4**	36.1	30.2	30.5	33.2	29.0	33.3	34.0	34.8	37.3
	Trace2Skill	9.3	10.1	22.7	25.1	34.5	36.7	38.0	**39.3**	26.1	25.7	26.9	24.5	33.2	33.7	34.5	36.5
LiveMath	NoSkill	34.1	34.9	34.4	36.3	31.7	32.0	30.9	31.7	40.1	38.7	37.9	35.2	52.2	**53.5**	51.9	49.2
	Direct	33.3	32.3	33.3	30.9	28.5	29.8	31.5	29.8	36.3	36.6	38.4	37.4	50.0	**51.9**	49.7	51.1
	SkillOpt	40.3	40.6	39.8	40.6	31.5	33.6	33.1	35.5	34.7	37.4	37.1	32.5	55.4	55.1	**58.6**	54.3
	Trace2Skill	53.2	58.1	56.5	54.6	33.9	34.4	33.6	37.1	40.6	39.0	36.8	35.5	61.6	**62.1**	57.8	60.5
UCSB SOP	NoSkill	52.8	51.0	52.6	59.2	61.7	60.2	60.2	60.2	66.7	65.9	66.7	64.2	69.9	69.3	**70.3**	68.4
	Direct	54.1	57.5	55.2	60.4	59.0	58.7	59.4	63.0	66.5	65.5	66.1	61.3	68.2	**68.6**	68.2	66.9
	SkillOpt	66.3	66.3	65.9	62.9	67.4	67.2	65.9	65.1	68.6	67.8	69.3	67.6	72.6	**73.1**	71.6	70.3
	Trace2Skill	62.1	61.9	61.9	62.1	58.9	57.5	60.6	62.9	65.1	66.1	65.0	64.0	**71.8**	70.3	71.2	68.2
Amazon SOP	NoSkill	28.3	33.9	88.6	95.1	82.7	82.7	**100.0**	99.6	2.6	4.2	58.9	87.8	58.5	57.1	**100.0**	99.0
	Direct	1.7	1.7	1.7	1.7	90.8	80.2	**99.4**	99.2	2.6	1.7	85.7	89.5	57.8	57.8	95.7	99.0
	SkillOpt	8.8	6.7	94.8	98.2	**100.0**	**100.0**	**100.0**	**100.0**	1.7	9.0	61.5	80.0	59.1	58.5	**100.0**	99.3
	Trace2Skill	19.6	8.6	80.6	93.3	77.1	82.9	**100.0**	99.9	9.4	6.8	58.5	85.1	57.1	55.9	98.0	99.3
SimpleQA	NoSkill	10.7	5.1	4.7	5.0	33.8	30.1	29.9	29.7	31.6	29.2	30.4	29.6	**69.4**	67.8	68.0	68.8
	Direct	8.0	6.3	6.1	6.4	28.2	29.7	30.1	29.3	27.6	25.0	24.6	25.7	65.0	66.1	67.0	**69.2**
	SkillOpt	11.3	6.3	6.3	5.9	33.4	30.1	30.1	32.9	27.7	30.9	29.3	30.8	67.6	68.0	66.6	**68.4**
	Trace2Skill	9.9	6.0	5.4	5.6	33.3	29.8	29.0	31.1	30.3	31.2	31.2	28.0	67.8	67.4	66.9	**68.8**
SQuAD	NoSkill	10.7	36.6	78.6	79.3	0.0	85.4	85.6	86.3	1.8	**90.0**	89.0	89.4	23.9	63.4	63.1	68.3
	Direct	46.7	76.4	75.9	76.3	42.4	80.4	80.3	82.0	37.6	**89.2**	88.3	87.4	59.3	73.6	74.4	77.2
	SkillOpt	29.0	77.0	77.6	77.4	0.0	82.4	82.1	82.6	50.3	**87.1**	86.9	84.7	49.7	63.8	64.1	66.6
	Trace2Skill	76.9	55.0	80.8	79.2	16.1	86.7	86.9	87.2	69.3	**91.0**	90.6	90.7	72.8	80.6	80.8	81.1
Hotpot Context	NoSkill	18.6	59.3	59.7	58.6	0.0	66.2	66.2	**67.8**	3.4	64.3	64.6	65.8	22.6	62.0	61.9	61.3
	Direct	21.3	55.2	55.3	57.3	16.6	66.8	66.9	**69.2**	9.9	62.9	62.7	64.8	29.7	64.6	64.3	60.9
	SkillOpt	0.0	44.0	47.9	59.7	0.0	68.2	68.4	**68.6**	0.0	63.9	63.0	64.9	0.0	13.6	12.2	12.2
	Trace2Skill	26.2	58.4	58.6	59.2	1.0	68.6	68.1	**68.7**	17.6	62.8	62.1	65.4	28.0	61.6	62.0	61.3
Hotpot FullWiki	NoSkill	24.2	25.9	26.8	26.9	29.1	30.4	29.3	31.1	31.1	30.9	31.3	**33.2**	21.0	22.6	23.1	19.9
	Direct	1.4	3.9	28.4	**29.8**	26.7	26.1	26.7	28.0	27.2	26.7	27.8	27.4	27.2	27.2	27.2	26.8
	SkillOpt	21.1	24.0	34.9	**35.6**	19.8	20.7	21.0	23.9	20.1	24.0	23.6	24.4	15.2	20.0	17.1	16.8
	Trace2Skill	32.7	34.1	36.0	39.1	44.0	41.6	41.7	42.7	41.1	**45.0**	43.2	41.9	31.6	30.1	32.6	28.9

Table 18 reproduces the complete 512-cell Default matrix here for local reference. It reports every Benchmark×Model×Skill×Harness cell. Cells are three-repeat means without standard errors, and the floor cells listed in Table 15 should be read as instrument floors rather than graded scores.

C.1.2 Detailed Evidence for Context-Dependent Rankings

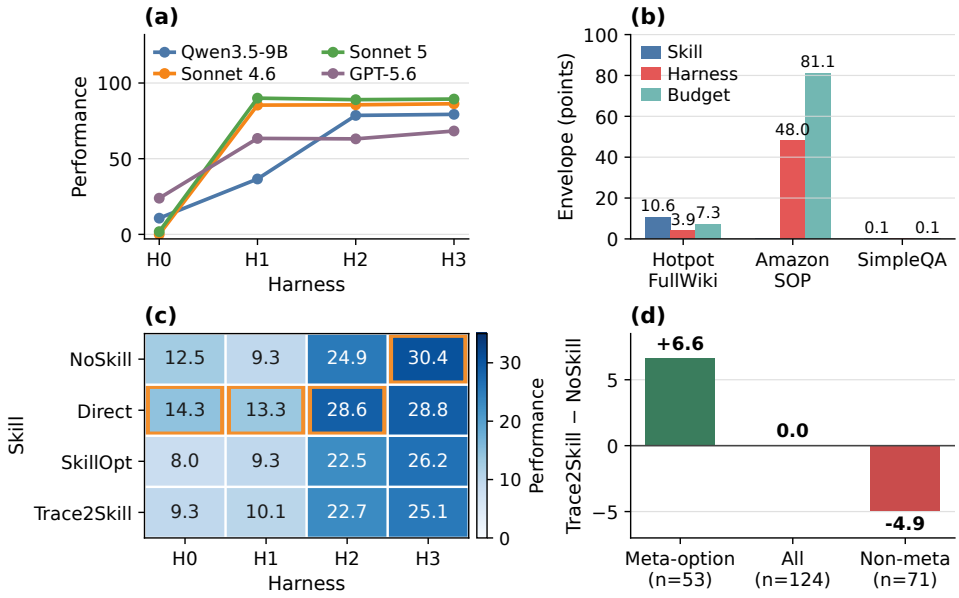


Figure 10: Two forms of intervention-response heterogeneity. (a) Model ordering reverses: SQuAD × NoSkill. (b) Intervention effects vary by task. (c) Benchmark × Model block: Spreadsheet × Qwen3.5-9B. (d) Same Skill, opposite subgroup effects: LiveMath × Sonnet 5.

The absence of a context-free winner is first visible along the Model axis. Among the 32 Benchmark×Skill rows of the Default matrix, 30 have a unique best Model, distributed 12/10/6/2 across GPT-5.6, Sonnet 4.6, Sonnet 5, and Qwen3.5-9B, respectively. These are not merely near-tie reversals: Sonnet 4.6 reaches the benchmark-wide maximum on Spreadsheet at 53.1 but ranks last on LiveMath at 37.1, whereas GPT-5.6 reaches 69.4 on SimpleQA but ranks last on Hotpot FullWiki at 32.6, behind Sonnet 5 at 45.0. Appendix Figure 17 further shows that Model ordering itself changes with the surrounding intervention: Harness produces especially prominent crossings on SQuAD and Hotpot Context, whereas Skill changes the ordering on Hotpot FullWiki. Figure 10a visualizes the SQuAD+NoSkill reversal under changing Harness support.

The other intervention axes exhibit the same conditionality. After balanced marginalization, Trace2Skill contributes to positive Skill envelopes on Hotpot FullWiki ($\Delta_S^+ = 10.58$ points) and LiveMath (8.15), whereas no fixed alternative Skill outperforms NoSkill on Spreadsheet or Amazon SOP ($\Delta_S^+ = 0$ for both). Harness effects are large on Amazon SOP and on the answer-interface-sensitive SQuAD and Hotpot Context, but are nearly flat on LiveMath and SimpleQA; Appendix A.6.1 shows that the largest of these contrasts also contain floor and answer-contract effects. Budget exhibits an even sharper task boundary: Amazon SOP improves dramatically when Low is replaced by a feasible Medium or High resource ceiling because Low does not provide enough turns or wall-clock time to complete its long tool workflow ($\Delta_B^+ = 81.13$), whereas SimpleQA is essentially saturated ($\Delta_B^+ = 0.10$). Thus, stronger Skills, richer Harnesses, and larger Budgets can each improve performance, have little effect, or reduce performance depending on the task and the components already present. Figure 10b contrasts three representative task profiles.

The same contrast can appear within a single Benchmark×Model block. A floor-free example is Qwen3.5-9B on Spreadsheet. Moving from H1 to H2 raises NoSkill from 9.3 to 24.9, Direct from 13.3 to 28.6, SkillOpt from 9.3 to 22.5, and Trace2Skill from 10.1 to 22.7, showing a consistent execution-support gain. Skill effects remain conditional: Direct exceeds NoSkill at H0 and H1, but NoSkill is strongest at H3 (30.4 versus 28.8, 26.2, and 25.1 for Direct, SkillOpt, and Trace2Skill). Figure 10c visualizes the complete 4×4 block.

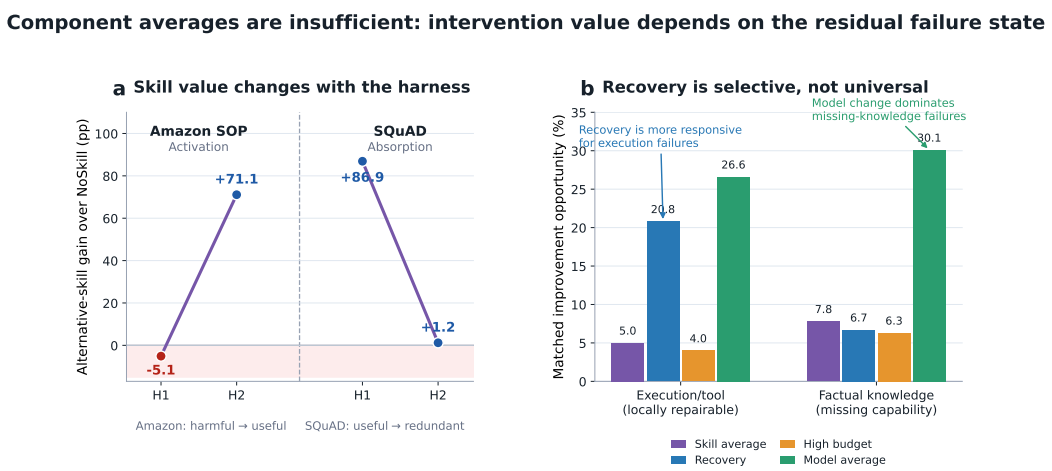


Figure 11: Two empirical examples of why component averages are insufficient. (a) Harness-conditioned skill value on outcome-selected task-instance–model subsets: Direct on Amazon SOP changes from -5.1 points at H1 to $+71.1$ at H2, while the SQuAD skill gain contracts from $+86.9$ to $+1.2$ points; the dashed line separates the benchmark examples. (b) Root-conditioned opportunity: Recovery is more responsive for Execution/tool than Factual knowledge/recall failures, whereas Model changes have the largest opportunity for the latter. These are descriptive matched opportunities, not causal treatment effects; panel (a) uses selected subsets and should not be read as the Table 18 mean over all task instances, and the Amazon panel is floor-affected (Appendix A.6.1).

C.2 Intervention-Matching Heterogeneity

For statistical support, we fit categorical response surfaces separately within each benchmark after min–max normalization. The additive specification contains Model, Skill, and Harness main effects; the interaction specification additionally contains all Model×Skill, Model×Harness, and Skill×Harness terms. We evaluate both with PRESS leave-one-cell-out prediction over the 64 Default-budget cells per benchmark. The interaction specification increases mean LOO R^2 from 0.729 to 0.812, with a mean paired increase of 0.083 and improvement on seven of eight benchmarks. An exact one-sided sign-flip test over the eight benchmark-level R^2 differences gives $p = 0.0078$. This test supports non-additive context dependence within the observed benchmark surfaces; it does not establish generalization to unseen benchmarks or identify residual failure as its cause.

Table 19: Direct empirical support for within-benchmark response heterogeneity. Skill×Harness percentages are benchmark-equal prevalence estimates under a prespecified five-point descriptive threshold. H1→H2 entries are pooled matched transition counts for the seven binary-outcome benchmarks; conditional rates divide rescues by H1 failures and regressions by H1 successes. Full benchmark distributions and uncertainty are reported in Appendix Tables 21 and 30. These are descriptive matched outcomes. The LiveMath partition is frozen but post hoc; the interaction labels depend on an uncalibrated five-point tolerance; and the H1/H2 executions are matched replays rather than randomized prospective continuations.

Matched analysis	Population	Empirical composition
Skill×Harness	21,948 task-instance–model–alternative-Skill units	Weak/neutral 85.94%; Absorption 3.77%; Activation 2.35%; Complementarity 1.48%; Interference 6.46%.
H1→H2	85,392 binary-outcome executions	Rescue 3,886 (8.54% of H1 failures); Preserve 37,568; Regression 2,340 (5.86% of H1 successes); Persistent failure 41,598.
LiveMath Skill subgroup	124 Sonnet 5 task instances, averaged over Default H0–H3 and three repeats	TRACE2SKILL minus NOSKILL: meta-option $+6.60$ points (53 instances); non-meta -4.93 points (71 instances); all instances, averaged over H0–H3, 0.00 points.
Amazon H1→H2	2,400 fractional-score executions	Improve 1,070; unchanged 1,322; degrade 8; mean change $+42.24$ points.

C.3 WITHIN-BENCHMARK RESPONSE HETEROGENEITY

A benchmark mean compresses distinct transitions at the task-instance level into one number:

$$\mathbb{E}[\Delta] = p_{\text{rescue}}\,\mathbb{E}[\Delta \mid \text{rescue}] + p_{\text{neutral}}\,\mathbb{E}[\Delta \mid \text{neutral}] + p_{\text{regression}}\,\mathbb{E}[\Delta \mid \text{regression}]. \quad (5)$$

Figure 10d illustrates this compression directly. Under the frozen post-hoc LiveMath partition, TRACE2SKILL versus NOSKILL changes Sonnet 5 performance by $+6.60$ points on 53 meta-option instances and -4.93 points on 71 non-meta instances; the weighted whole-benchmark contrast averaged over H0–H3 is approximately zero. Because the partition was defined after Skill selection and evaluation, this is descriptive subgroup evidence rather than a preregistered causal contrast. Table 19 provides four additional matched-unit views. Across 21,948 Skill×Harness units, 85.94% are weak or neutral under the prespecified five-point rule, while the remainder separates into absorption, activation, complementarity, and interference. Across 85,392 binary-outcome H1→H2 executions, H2 rescues 3,886 H1 failures but regresses 2,340 previously correct H1 executions. On Amazon SOP, 1,070 of 2,400 fractional-score executions improve, 1,322 remain unchanged, and eight degrade. Thus, a positive or near-zero benchmark mean can conceal substantial and opposing changes at the task-instance level.

C.3.1 LIVEMATH META-OPTION DIAGNOSTIC

The frozen post-hoc partition marks an item as meta when it contains the recurring meta option; 53 of the 124 test items meet this rule. The frozen SKILLOPT and TRACE2SKILL documents explicitly call attention to this choice, but differ in when they recommend selecting it. For Qwen3.5-9B, averaging three repeats over Default H0–H3, TRACE2SKILL scores 71.5% versus 11.2% for NOSKILL on meta items, but 43.7% versus 52.7% on the 71 non-meta items. The subgroup was defined after skill selection and evaluation, so this is a descriptive interaction; no ablation isolates the meta-option instruction from the rest of either Skill.

C.4 WHICH INTERVENTIONS HELP UNDER WHICH TASK

C.4.1 DETAILED TASK-CONDITIONED BOTTLENECK PROFILES

Factual knowledge/recall can favor Model changes. SimpleQA is the clearest model-led pattern: it asks for factual knowledge that is generally absent from the task context, and neither a procedural Skill nor execution feedback supplies facts missing from the base model. Its Model span is 60.85 points, compared with Skill, Harness, and Budget envelopes of 0.13, 0.00, and 0.10. More generally, current performance provides a coherent descriptive boundary for model escalation. Switching a non-GPT model to GPT-5.6 improves approximately 65% of 384 matched Benchmark$\times$Skill$\times$Harness comparisons, and a leave-one-benchmark-out linear rule predicts the direction of that gain in 79.4% of cases (Appendix Figure 20). This does not imply that a low score mandates escalation: the predictor uses within-benchmark normalized performance, its slope is partly exposed to baseline–gain mathematical coupling, and LOBO contains only eight benchmark families.

Procedural can favor Skills. Skill value concentrates where a reusable decision or search procedure can be encoded. On LiveMath's 53 meta-option items, TRACE2SKILL reaches 71.5% against 11.2% for NOSKILL, but falls to 43.7% against 52.7% on the 71 remaining items, indicating a narrow reusable answer-selection rule rather than general mathematical capability (Appendix C.3.1). Hotpot FullWiki supplies a second pattern: multihop retrieval can reuse search and evidence-integration procedures, and its Skill envelope reaches 10.58 points. In contrast, no fixed alternative Skill beats NOSKILL after balanced marginalization on Spreadsheet or Amazon SOP, where runtime state and execution dominate. Skill value is therefore primarily policy-dependent rather than universally capability-enhancing. However, Procedural$\rightsquigarrow$Skill remains a task-level hypothesis: the frozen root-conditioned analysis does not establish a Skill-specific rule.

Execution/tool and Evaluation/answer contract can favor Harness support. The Harness changes what the model can observe and repair: H1 adds a standardized answer or tool interface, H2 adds execution observations, error feedback, and bounded recovery, and H3 adds one outcome-blind revision. Spreadsheet and Amazon illustrate why this matters for state-changing tool workflows, where a reasonable plan can still fail locally during execution. The interpretation has an important measurement boundary. Some of the largest apparent Harness gains occur on SQuAD and Hotpot Context because H0 can fail the output or answer contract; their H0$\rightarrow$H1 gains therefore mix interface specification with capability improvement. They cannot be read directly as recovery of reasoning, and the floor audit in Appendix A.6.1 treats them as contract-level effects.

Horizon/termination can favor additional Budget. Budget is not a smooth test-time scaling axis in this matrix because Low, Medium, and High jointly change token, turn, and timeout ceilings. Its clearest pattern is

$$\text{insufficient} \longrightarrow \text{feasible} \longrightarrow \text{saturated}.$$

Amazon's long tool workflow cannot usually finish under Low and produces an 81.13-point Low-to-best envelope once Medium or High makes the path feasible. By contrast, SimpleQA, SQuAD, and Hotpot Context change little after feasibility is reached, and High frequently ties or falls slightly below Medium (Appendix Figure 18). Budget primarily changes whether a valid path can be completed; beyond that threshold, additional resources often saturate. Horizon/termination$\rightsquigarrow$Budget nevertheless remains a task-level hypothesis in the root analysis because the frozen Medium-budget source cohort contains only one labeled Horizon/termination case.

C.4.2 MATCHED CONFIGURATION EXAMPLES

Figure 12 illustrates the outcome-selected comparisons discussed. Each target changes one axis from its panel's initial configuration, with three complete repeats per cell; these examples do not estimate population prevalence or define a deployable selection policy.

Figure 13 visualizes the benchmark-level marginal summary using Equation 2. For Δ_B^+, H2 is fixed, Models and Skills are weighted equally, and Low, Medium, and High all enter the maximum; Low is the reference endpoint. UCSB remains unavailable on the four-model support because its three API-model Low runs alter normal workflow semantics.

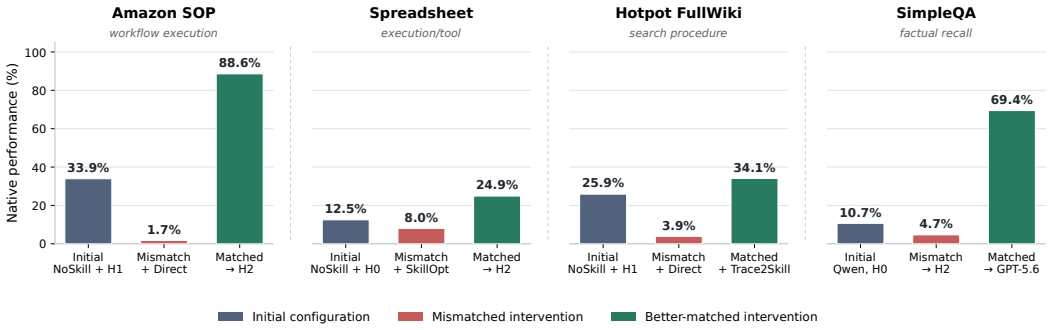


Figure 12: Representative matched configurations in which one intervention lowers performance while another raises it. Within each panel, the red and green bars each change exactly one axis from the blue initial configuration while holding the other axes fixed. Values are benchmark-native performance averaged over three complete repeats and should be compared only within a panel. "Mismatched" and "better-matched" are retrospective descriptive labels, not prospective routing decisions.

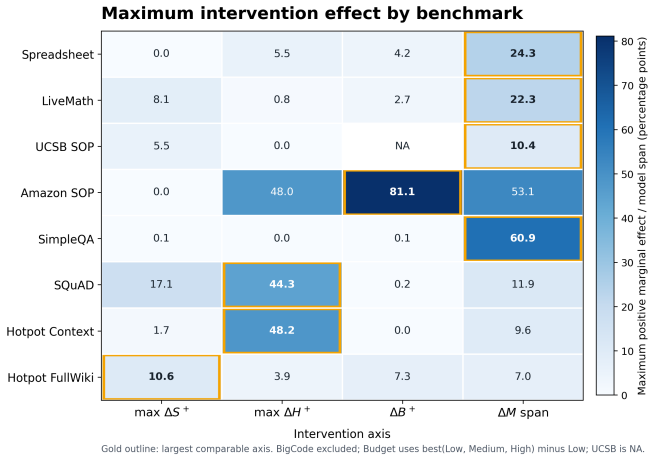


Figure 13: Maximum observed intervention effects by benchmark. The axes report positive Skill, Harness, Budget, and Model-span effects (definitions in Eq. 2). These are within-axis observed envelopes with different candidate sets, not action-count-matched rankings.

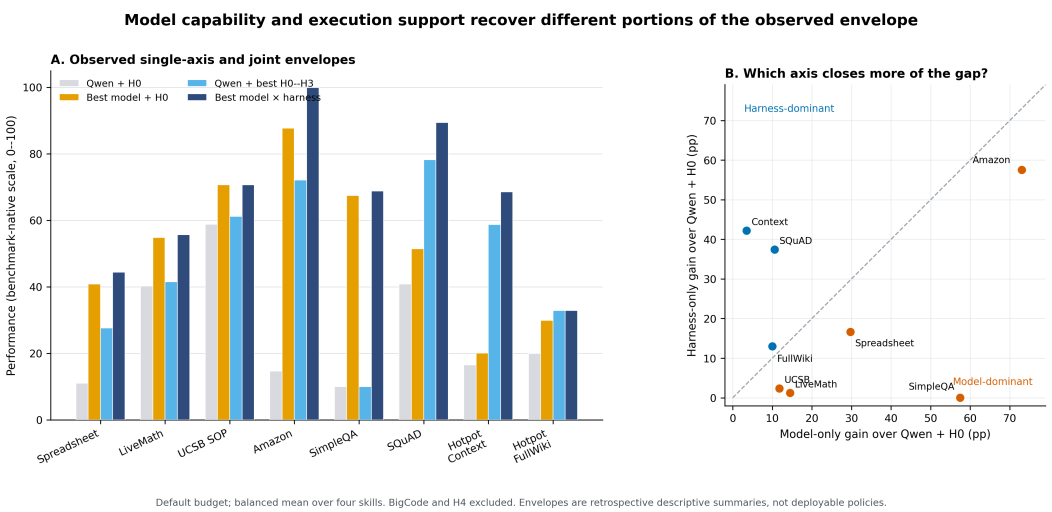


Figure 14: Model capability and execution support recover different portions of the observed performance envelope. Panel A compares a fixed Qwen3.5-9B+H0 reference, best model at H0, best harness for Qwen3.5-9B, and best Model×Harness cell; Panel B contrasts model-only and harness-only gains. These retrospectively selected envelopes are descriptive rather than an identified additive causal equation.

C.4.3 Model- and Budget-Conditioned Intervention Profiles

The response profiles support several recurring observations. On Spreadsheet, skills act as a debuff on average: NoSkill performs best across models, while recovery and reflection appear important for execution-heavy workflows. On LiveMath, Qwen3.5-9B outperforms both Sonnet models under Trace2Skill, plausibly because those skills were trained from Qwen3.5-9B trajectories, but GPT-5.6 remains strongest overall. On UCSB SOP, SkillOpt consistently outperforms other skill methods, which is consistent with a procedure- and constraint-sensitive task requiring specific action sequences and verification rules. On Amazon SOP, recovery and reflection help, with weaker models often benefiting more from the stronger harness. SimpleQA remains primarily model-sensitive. Trace2Skill is comparatively useful on SQuAD and Hotpot FullWiki, where reusable context-reading, evidence-selection, retrieval, and reasoning procedures can be applied to the provided task state; its gain is smaller on Hotpot Context, where the relevant evidence is already supplied.

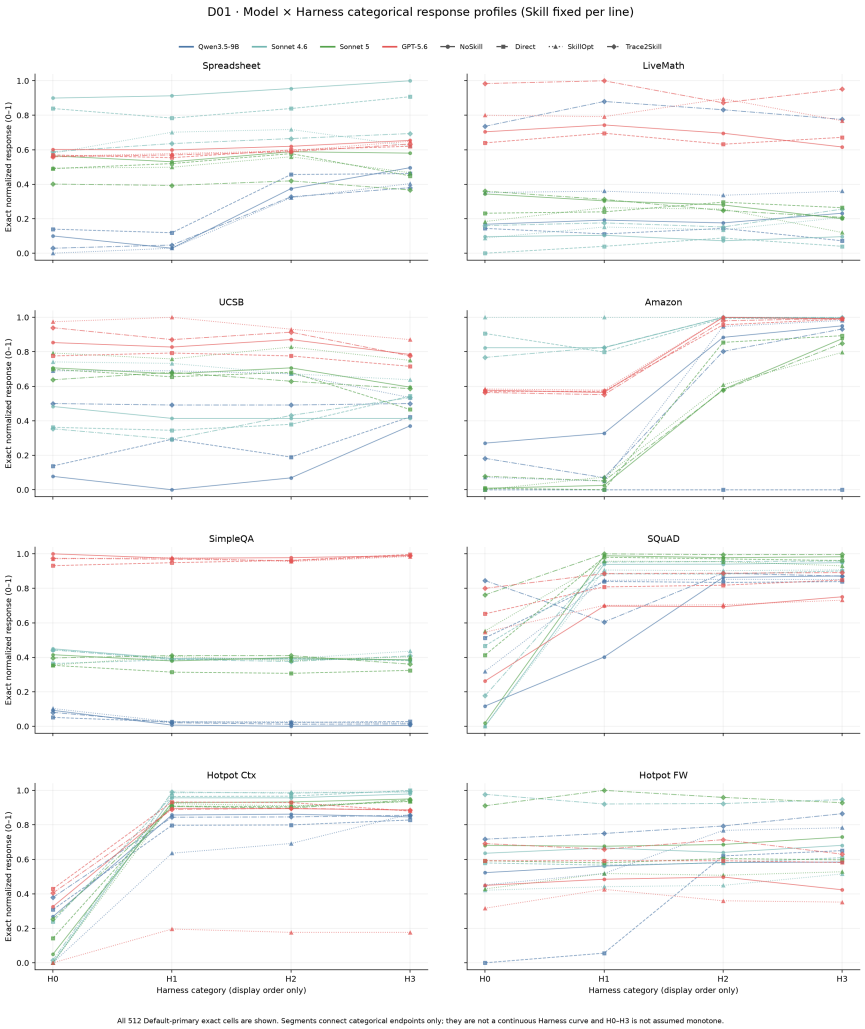


Figure 15: Model × Harness exact response profiles. Harness gains vary sharply across benchmarks and models; execution support can sometimes compensate for a weaker base model, but only on tasks with exposed and repairable failures.

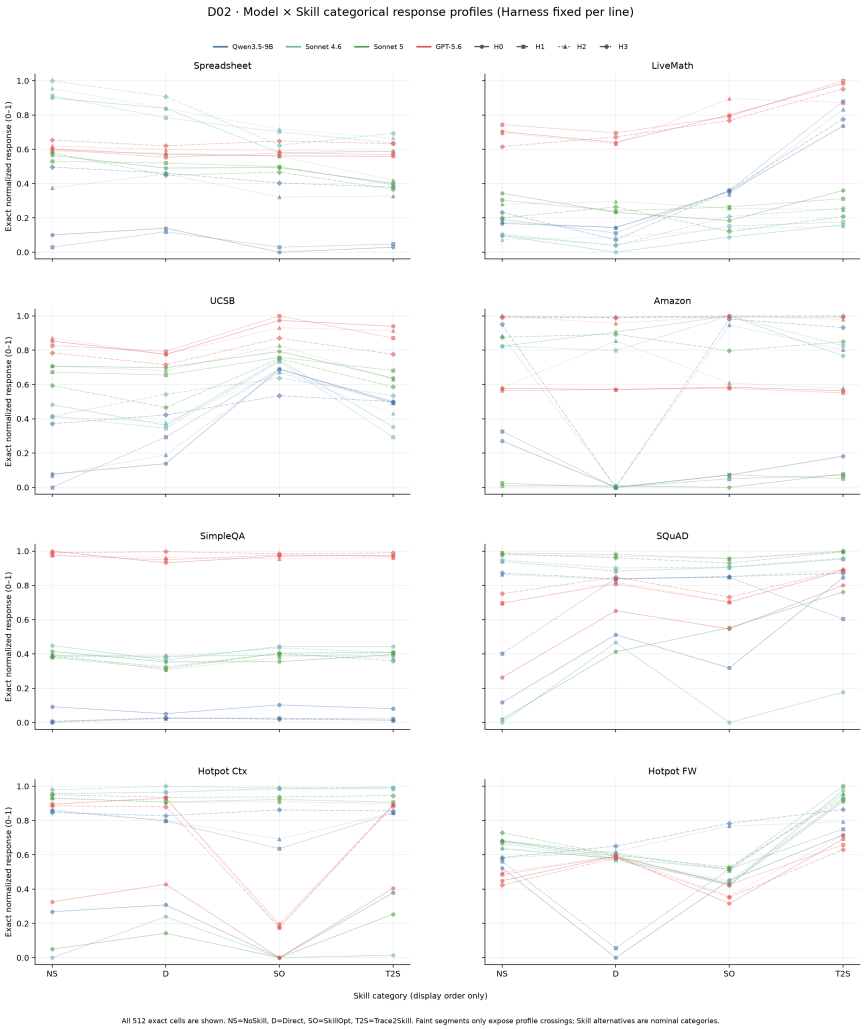


Figure 16: Model × Skill exact response profiles. Skill value is model- and task-conditional rather than a uniform offset to model capability.

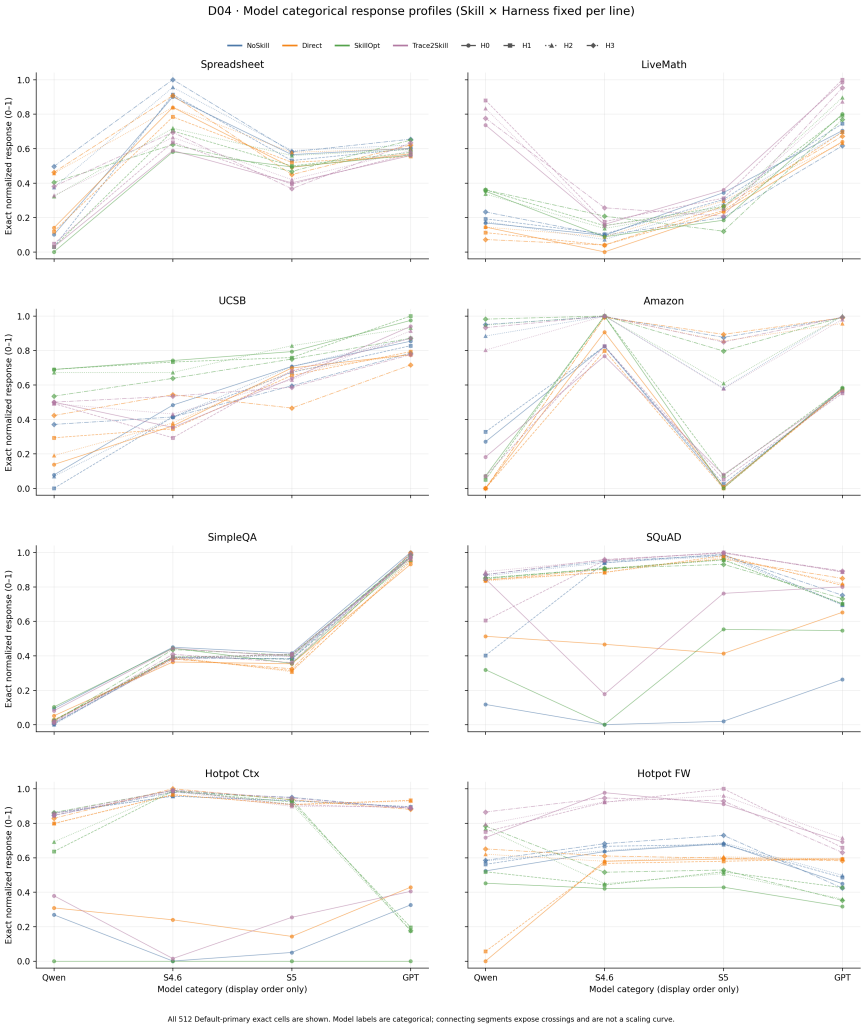


Figure 17: Exact Model response profiles at Default budget. Each panel shows all 64 Model × Skill × H0–H3 cells for one benchmark, min–max normalized within that benchmark for display. Each line fixes one Skill × Harness configuration while the categorical horizontal axis changes Model. The persistent separation on SimpleQA contrasts with frequent profile crossings elsewhere, showing that model changes do not act as a uniform offset to external interventions. Connected segments are visual guides between categorical endpoints, not a controlled model-scaling curve. BigCode is excluded.

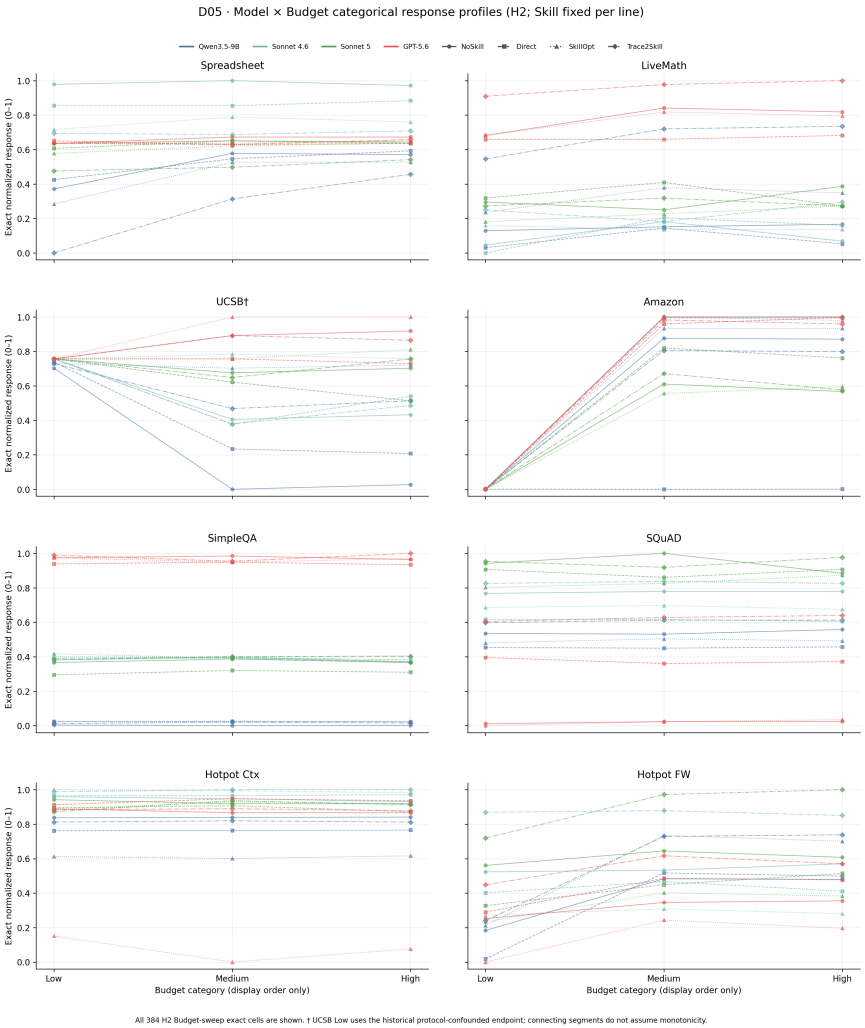


Figure 18: Exact Budget response profiles at H2. Each panel shows all 48 Model × Skill × Low/Medium/High cells for one benchmark, min–max normalized within that benchmark for display. Each line fixes one Model × Skill configuration. Amazon exhibits a broad Low-to-Medium feasibility threshold followed by saturation, whereas SimpleQA, SQuAD, and Hotpot Context are comparatively flat. The dagger marks UCSB because the frozen cross-model matrix uses its historical protocol-confounded Low endpoint. Connected segments do not assume monotonicity, and BigCode is excluded.

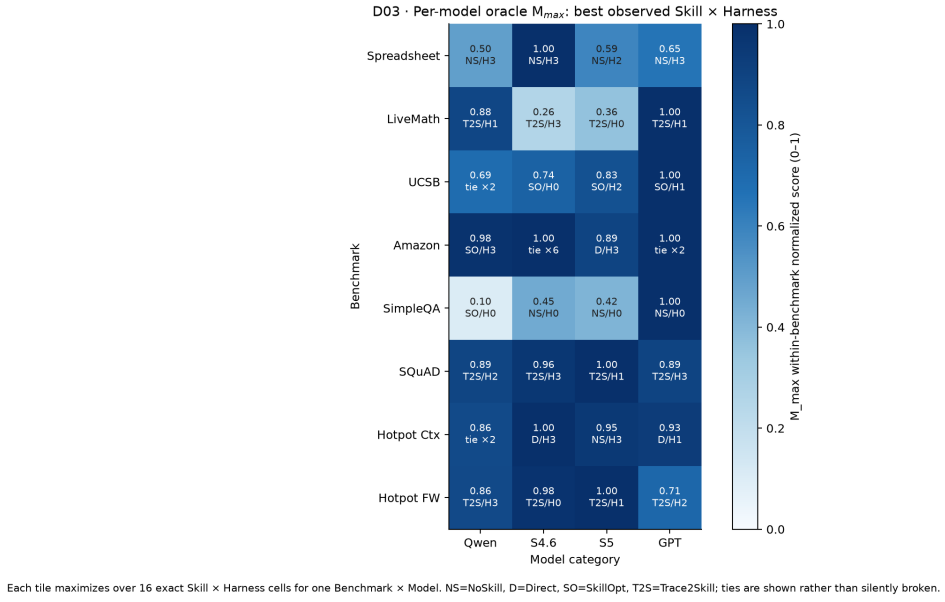


Figure 19: Model-conditioned maximum observed performance within the tested deployable configurations.

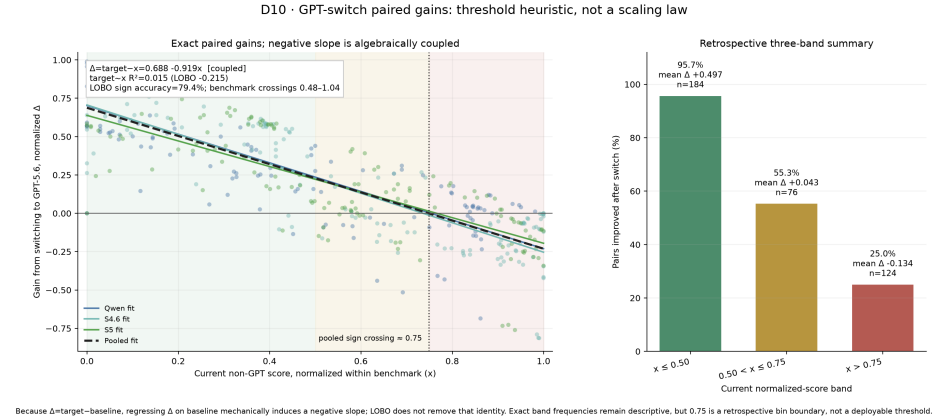


Figure 20: Descriptive Model-escalation boundary. Each of 384 points pairs a non-GPT source with GPT-5.6 under the same Benchmark×Skill×Harness context. The horizontal axis is the source's within-benchmark min–max normalized score and the vertical axis is the matched GPT-5.6 gain. Switching improves approximately 65% of pairs, and a leave-one-benchmark-out linear rule predicts gain direction in 79.4%. The relationship is descriptive: baseline appears with a negative sign in gain, normalization uses the complete benchmark surface, and eight held-out benchmark families do not establish a deployment threshold.

Table 20: Recurring Skill–Harness patterns using all task instances at Default budget in Table 18. Every raw H0–H2 score is copied from the corresponding Benchmark–Model–Skill cell of Table 18; Skill gains and $I(H2 \mid H1)$ are computed from the displayed one-decimal means. These four patterns are illustrated at the panel-mean level, whereas population prevalence is classified per task-instance–model unit. Model superscripts: † Qwen3.5-9B; ‡ Sonnet 5. § One endpoint of the contrast is a native-score floor cell (Appendix A.6.1); the pattern label is qualitative and the magnitude is not a calibrated interaction estimate. Blue/red bold highlights selected favorable/unfavorable outcomes and contrasts; it does not indicate statistical significance, and no cell in this table carries a standard error. H0 provides context; the pattern rule uses H1/H2 with a five-point descriptive threshold that was chosen a priori and not derived from A/A variation. Outcome-selected subset illustrations use different denominators and appear separately in Appendix Figure 21.

Pattern	Benchmark	Skill/contrast	H0	H1	H2	$I(H2 \mid H1)$
Absorption†	SQuAD	NoSkill	10.7	36.6	78.6	
		SkillOpt	29.0	77.0	77.6	
		Skill gain	+18.3	**+40.4**	**−1.0**	**−41.4**
Activation†	Hotpot FullWiki	NoSkill	24.2	**25.9**	**26.8**	
		SkillOpt	21.1	**24.0**	**34.9**	
		Skill gain	−3.1	−1.9	+8.1	**+10.0**
Complementarity‡§	Amazon SOP	NoSkill	2.6	**4.2**	**58.9**	
		Direct	2.6	**1.7**	**85.7**	
		Skill gain	0.0	−2.5	+26.8	**+29.3**
Interference†§	Amazon SOP	NoSkill	28.3	33.9	88.6	
		Direct	1.7	1.7	1.7	
		Skill gain	−26.6	**−32.2**	**−86.9**	**−54.7**

C.5 How Interventions Interact

For matched four-cell Skill×Harness comparisons, let $Y_{00} = Y(\text{NoSkill}, H1)$, $Y_{10} = Y(s, H1)$, $Y_{01} = Y(\text{NoSkill}, H2)$, and $Y_{11} = Y(s, H2)$, with $I_{S\times H} = Y_{11} - Y_{10} - Y_{01} + Y_{00}$. Using the fixed descriptive native-score tolerance $\epsilon = 0.05$ specified in Appendix A.6, Absorption occurs when both interventions help but their joint result stays within tolerance of the stronger one; Activation when neither alone clears tolerance but the combination does; Complementarity when the joint result exceeds the stronger individual intervention with positive interaction residual; and Interference when the combination underperforms a useful individual intervention.

Interaction patterns are recurrent but sparse. Across 21,948 matched task-instance–model–alternative-Skill units, Absorption accounts for 3.77%, Activation 2.35%, Complementarity 1.48%, and Interference 6.46% under benchmark-equal weighting. The four named patterns cover 14.06% in total; the remaining 85.94% are weak, neutral, or outside the prespecified definitions. An interaction therefore exists only in a minority of matched units, but its sign changes systematically enough to reject a uniformly additive view of external support.

The empirical taxonomy uses the same four H1/H2 cells for each task instance and Model. *Absorption* means both interventions help but their combination stays near the stronger one; *activation* means neither alone clears the fixed improvement tolerance while their combination does; *complementarity* means the combination exceeds the stronger individual intervention with a positive interaction residual; and *interference* means stacking reduces the benefit of a useful intervention. The exact five-point rules use the matched four-cell contrast defined above. Table 20 reports representative panels averaged over all task instances from the frozen Default matrix.

Absorption arises when both components address the same bottleneck. On Qwen3.5-9B SQuAD, SkillOpt gains 40.4 points over NoSkill at H1, but its incremental gain contracts to −1.0 point at H2. A consistent explanation is that the Skill and stronger Harness both help satisfy the answer procedure or contract; once H2 has removed that bottleneck, the Skill has little residual opportunity. This is descriptive overlap rather than an identified mediator, especially because SQuAD contains answer-contract floor effects.

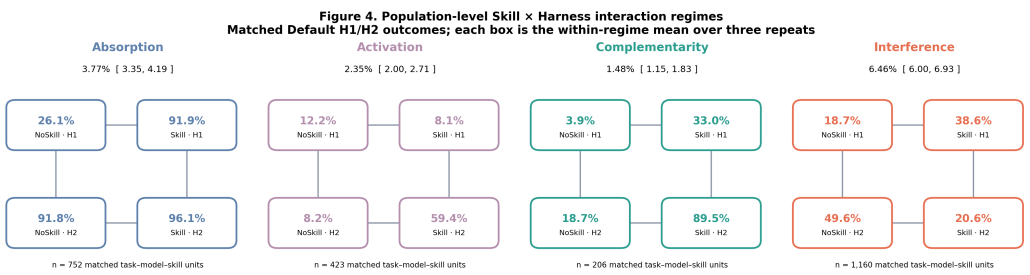


Figure 21: Outcome-selected task-instance–model subset examples of the four matched Skill × Harness interaction patterns. Each panel compares the same task instances under NoSkill and an alternative skill across H1 and H2. These conditional subset means differ from the Table 20 panel means over all task instances.

Activation and complementarity arise when components address different bottlenecks. On Qwen3.5-9B Hotpot FullWiki, neither SkillOpt alone nor H2 alone improves the H1 reference by five points, but the combination gains 9.0 points. The pattern is consistent with a division of labor: the Skill supplies a reusable search policy, while the Harness supplies the execution support needed to carry it out. The Sonnet 5 Amazon example is an even larger complementarity pattern, but one endpoint lies at the native-score floor; it is retained as a qualitative illustration rather than a calibrated magnitude.

Interference arises when stronger execution preserves a mismatched policy. On Qwen3.5-9B Amazon SOP, H2 raises NoSkill from 33.9 to 88.6, whereas Direct remains at 1.7 under H0–H3. The result is consistent with a mismatched Skill steering the system toward a bad policy that additional execution support does not undo. Because the 1.7 endpoint is an instrument floor, this row establishes catastrophic incompatibility within the tested configuration but does not quantify a general causal interaction.

The interaction sign follows the residual bottleneck. *The sign of an interaction depends on what failure remains after the first intervention is applied.* If the first component removes the bottleneck targeted by the second, incremental value is absorbed; if it exposes a different bottleneck that the second can repair, the pair activates or becomes complementary; if it imposes a mismatched policy that the second component executes or preserves, the pair can interfere. The full intervals and API-model subset appear in Tables 21 and 17.

C.5.1 Skill–Harness Configuration View

For benchmark d, skill s, and harness h, define performance relative to the benchmark-specific NoSkill+H0 reference as

$$R_d(s,h) = \bar{Y}_d(s,h) - \bar{Y}_d(\text{NoSkill}, H0), \tag{6}$$

where $\bar{Y}_d(s,h)$ is the equal-weight mean over the four models at Default budget using the benchmark's native metric.

C.5.2 Population Prevalence of Interaction Patterns

Table 21: Population prevalence of the four Skill–Harness patterns. Prevalence is the equal-weight macro average across eight comparable benchmarks; intervals use instance-UID cluster bootstrap. The representative benchmark is selected mechanically by the highest within-benchmark prevalence.

Pattern	Prevalence	95% CI	Representative benchmark
Absorption	3.77%	[3.35%, 4.19%]	SQuAD
Activation	2.35%	[2.00%, 2.71%]	Amazon
Complementarity	1.48%	[1.15%, 1.83%]	Amazon
Interference	6.46%	[6.00%, 6.93%]	LiveMath

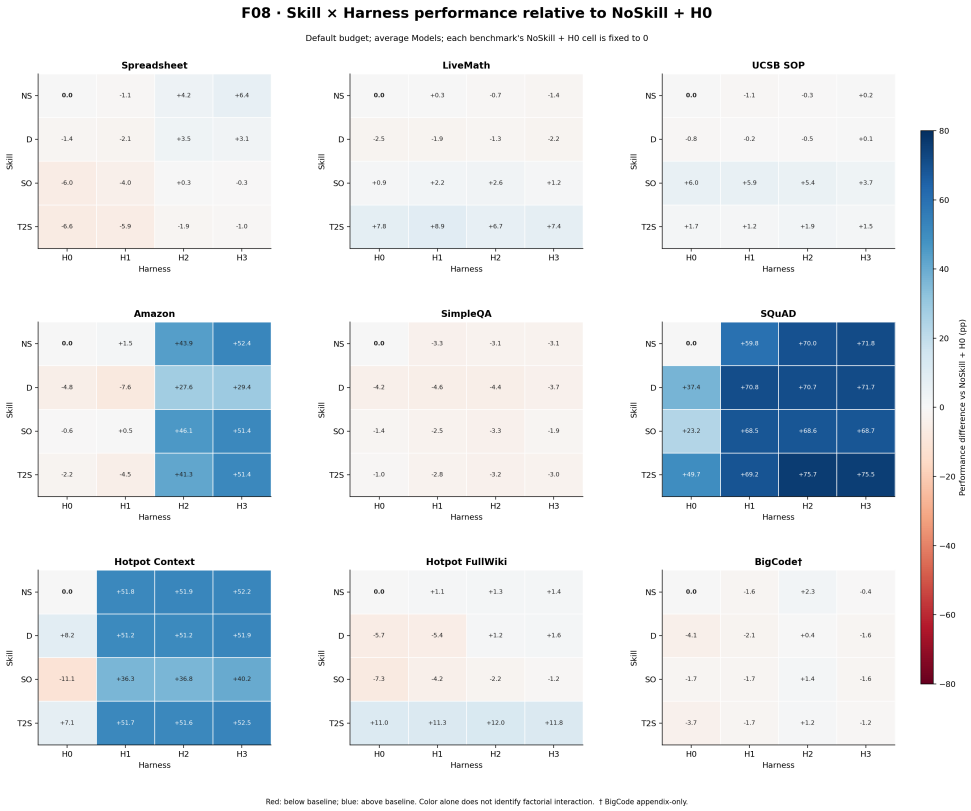


Figure 22: Skill × Harness performance relative to the benchmark-specific NoSkill+H0 reference. Each panel sets that benchmark's reference cell to zero; blue cells are above and red cells below the reference, in percentage points. This is a descriptive configuration-level marginal view. Interaction labels in Section C.5 instead use matched four-cell task-instance–model contrasts. BigCode is shown only as appendix evidence.

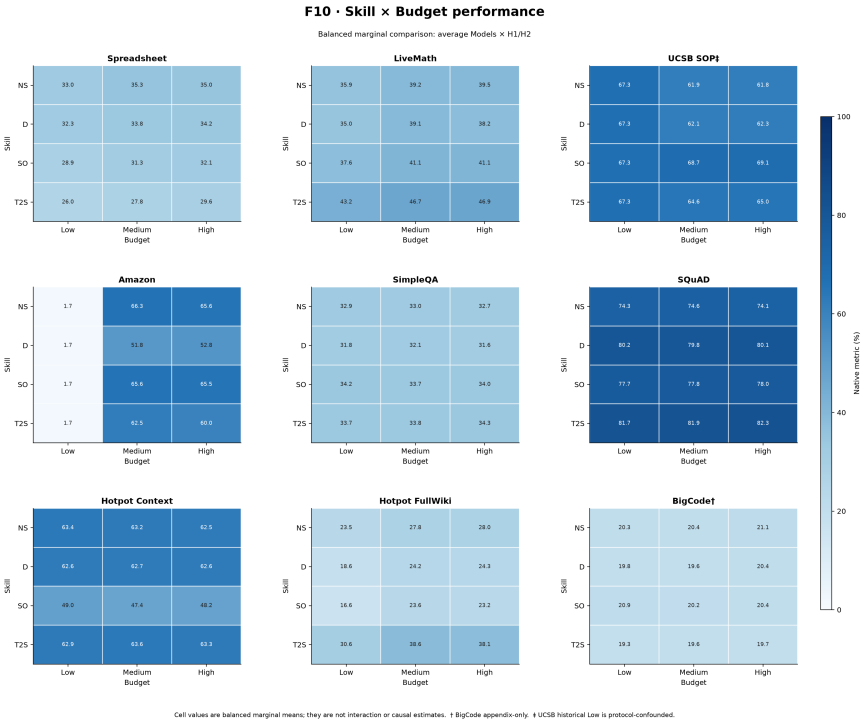


Figure 23: Skill × Budget balanced marginal performance, averaging over Models and H1/H2. Budget gains depend on both task and skill: Amazon exhibits a Low-to-Medium feasibility threshold, FullWiki shows a larger gain for TRACE2SKILL, and several benchmarks remain nearly flat. The historical UCSB Low endpoint is protocol-confounded, and BigCode is appendix-only.

Across 21,948 matched task-instance–model–alternative-skill units, the four named patterns cover about 14% of outcomes. Interference is the most frequent named pattern, but it still accounts for only 6.46%; the result does not imply that combining Skill and Harness is usually harmful. Instead, the evidence supports sparse, task-dependent interaction pockets: external interventions are neither uniformly additive nor safely stackable, and their joint value depends on the active bottleneck and surrounding execution configuration.

C.5.3 ADDITIONAL PAIRWISE INTERACTIONS

The additional pairwise views use balanced marginal means and therefore expose recurring conditional patterns rather than causal interaction estimates. BigCode remains appendix-only, and the historical UCSB Low endpoint is protocol-confounded.

Skill × Budget. Additional budget is useful mainly when the task contains a resource-dependent search or execution path. On Amazon, every skill remains at 1.7 under Low but rises to 51.8–66.3 under Medium, revealing a workflow-feasibility threshold rather than a smooth compute law. On FullWiki, TRACE2SKILL rises from 30.6 under Low to 38.6 under Medium, while NOSKILL rises from 23.5 to 27.8; the widening skill advantage is consistent with Skill and Budget being complementary for multi-step retrieval. LiveMath and Spreadsheet show smaller Low-to-Medium gains, whereas SimpleQA, SQuAD, and Hotpot Context are nearly flat. High rarely improves materially over Medium, indicating saturation once the required trajectory becomes feasible.

Model × Skill/Harness. The same external intervention can be valuable for one model and redundant or harmful for another. On LiveMath, TRACE2SKILL improves Qwen from 34.9 to 55.6, GPT-5.6 from 51.7 to 60.5, and Sonnet 4.6 only from 31.6 to 34.7, while producing no gain for Sonnet 5 (38.0 to 38.0). UCSB shows a similar contraction in SKILLOPT gain from +11.4 points for Qwen to approximately +2.4 for Sonnet 5 and GPT-5.6, whereas FullWiki retains a roughly +9–13 point TRACE2SKILL gain across models. Harness effects follow a different pattern: Hotpot Context

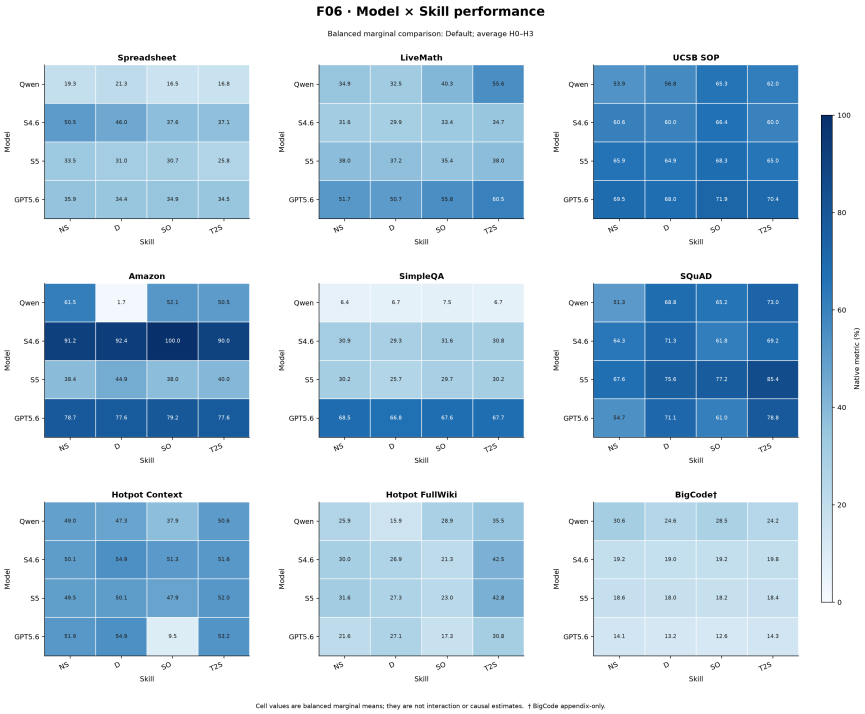


Figure 24: Model × Skill balanced marginal performance at Default budget, averaging over H0–H3. Skill value changes non-uniformly across models and benchmarks; the cells are descriptive marginal means rather than causal interaction estimates. BigCode is appendix-only.

and SQuAD benefit strongly across models, Amazon shows large H1-to-H2 recovery, and SimpleQA and LiveMath remain largely insensitive. FullWiki's harness gain is concentrated in Qwen. Model capability can therefore absorb a skill without eliminating the value of execution support, and it can determine whether feedback is converted into recovery.

Budget × Model. More budget does not induce a uniform model ranking or monotonic improvement. Amazon's Low endpoint collapses all four models to 1.7, while Medium exposes large model differences; FullWiki gains from Low to Medium range from +1.7 points for Sonnet 4.6 to +12.3 for Qwen, and LiveMath similarly benefits Qwen more than the API models. In contrast, SimpleQA, SQuAD, and Hotpot Context remain almost unchanged across budget. High is frequently tied with or slightly below Medium, supporting model-conditional thresholds and saturation rather than a general "more budget is better" rule.

Together, these views show that the value of external interventions changes non-uniformly with both task structure and model capability. Skills help when they supply a reusable protocol that the current model can exploit; harnesses help when execution failures are observable and repairable; budget helps when the task requires a longer search or tool trajectory. None has a model-independent or task-independent ordering.

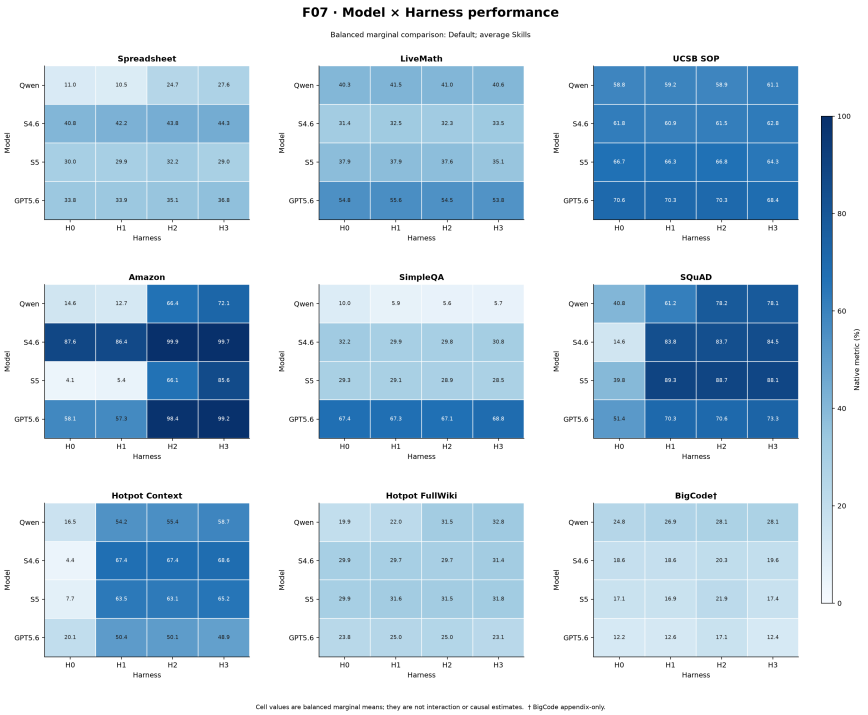


Figure 25: Model × Harness balanced marginal performance at Default budget, averaging over Skills. Harness gains are concentrated on tasks with exposed execution or answer-interface failures and vary across models; the cells are descriptive marginal means. BigCode is appendix-only.

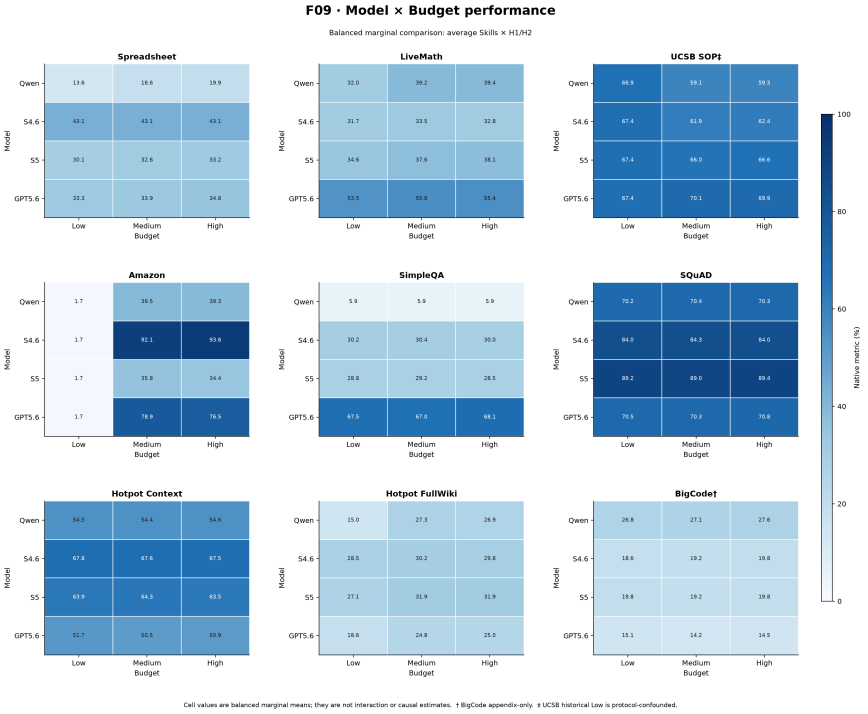


Figure 26: Model × Budget balanced marginal performance, averaging over Skills and H1/H2. Budget changes expose task- and model-specific feasibility thresholds, rank differences, and saturation rather than a uniform monotonic gain. The historical UCSB Low endpoint is protocol-confounded, and BigCode is appendix-only.

C.6 Which Interventions Help Under Which Failure

We assign the earliest decisive error supported by the saved trajectory to one of the eight categories in Table 22. Every label comes from that trajectory alone. The human-authored task properties of Table 9, including their expected bottlenecks, are never used as evidence here, so the agreement between a benchmark's expected bottleneck and its measured root composition is a result rather than an assumption. A label is evidence-backed when a logged event or evaluator record localizes the error. When deterministic evidence does not determine the category, a fixed Qwen3.5-9B judge supplies an AgentRx-inspired, constraint-based causal hypothesis. Model-assisted and answer-level hypotheses remain marked separately from deterministic labels. Composition analyses normalize the 2,225 labeled source failures within each benchmark.

Reliability status and the study that would establish it. We report no reliability estimate for these labels, and we do not want that absence to be inferred only from the hypothesis markers. Three specific gaps exist. First, there is no human–human agreement on the location of the earliest decisive step, so the taxonomy's central operational choice is unvalidated. Second, there is no human–judge agreement, so the Qwen3.5-9B hypotheses have no measured error rate against expert labels; because that judge is also an evaluated model and the skill-construction model, its errors may correlate with the configurations being compared rather than being independent noise. Third, there is no sensitivity analysis over alternate decisive steps, so we cannot state how many labels would change under a defensible competing reading of the same trajectory.

The study we consider necessary is a double-blind annotation of a stratified sample of 400 to 600 source trajectories, sampled to cover every benchmark and every candidate root, by two annotators working independently from the saved trajectory alone. It would report agreement on the decisive step and on the primary root, an adjudication record for disagreements, the disagreement rate between the judge and the deterministic checks on the subset where both apply, and the fraction of labels that change under the alternate-step reading. Until those numbers exist, the Reasoning, State/grounding, Factual knowledge/recall, and Abstention/calibration rows should be read as structured hypotheses, and any conclusion that depends on distinguishing them is provisional.

Coverage and missingness. The repeat-1 common-support cohort holds 3,939 source failures and 2,225 receive labels, so 1,714 are unassigned. The missingness is concentrated rather than random, and the two Hotpot views account for the large majority of it, which is why their one and ten labeled failures cannot explain their benchmark-wide behavior. We therefore report root-conditioned quantities on the assigned subset only and provide no weighted bound for the unassigned remainder. A stratified audit of the unassigned cases, followed by inverse-probability weighting or a worst-case missingness bound, is required before any root-conditioned percentage can be read as a property of the full failure population.

C.6.1 Root-Conditioned Intervention Opportunity

Failure type provides partial, but not universal, information about intervention opportunity. The factorial surface and interaction patterns establish that intervention value changes with the failure left by the surrounding configuration, but they do not identify that failure directly. If the residual-failure hypothesis is correct, a trajectory-level root label should provide information about intervention opportunity beyond benchmark identity. We therefore freeze each source trajectory before inspecting matched target outcomes, assign its earliest supported failure label, and test that incremental association.

Freeze the failure before testing its response. We fix the NoSkill+H1+Medium source condition and assign the earliest decisive error supported by each failed trajectory before inspecting any matched target outcome. The eight labels are Procedural, Reasoning, Execution/tool, State/grounding, Horizon/termination, Factual knowledge/recall, Abstention/calibration, and Evaluation/answer contract, defined in Appendix Table 22 and distributed over the 2,225 labeled failures in Table 14. That distribution is the central obstacle to interpreting roots: six of eight benchmarks are effectively dominated by one root, with UCSB SOP 78.57% Procedural, Spreadsheet and Amazon predominantly Execution/tool, SQuAD entirely Evaluation/answer contract, and SimpleQA 74.84% Factual with 24.78% Abstention. Only LiveMath, SimpleQA, and UCSB SOP provide the within-benchmark

Table 22: Domain-general root-cause taxonomy. Each failed trajectory receives one primary label corresponding to its earliest decisive error; later symptoms may be retained as secondary evidence.

Root cause	Operational definition
Procedural	The required information is available, but the trajectory violates a mandatory step, ordering constraint, schema, or task rule.
Reasoning	The relevant facts or state are available, but the plan, calculation, inference, or final conclusion is incorrect.
Execution/tool	The intended action is plausible, but a tool call, API invocation, program, syntax, or runtime operation fails.
State/grounding	The trajectory acts on the wrong entity, document, cell, evidence item, or environment state.
Horizon/termination	No earlier decisive error is found, but the trajectory stops too early, loops, or exhausts its turn, token, or time allowance.
Factual knowledge/recall	The required fact is neither supplied by the task nor correctly recalled, and no earlier procedural, reasoning, execution, or grounding error explains the failure.
Abstention/calibration	The model explicitly declines or withholds an answer despite the task requiring one.
Evaluation/answer contract	The substantive answer may be available, but the submitted answer violates the evaluator-visible answer contract or cannot be credited under it.

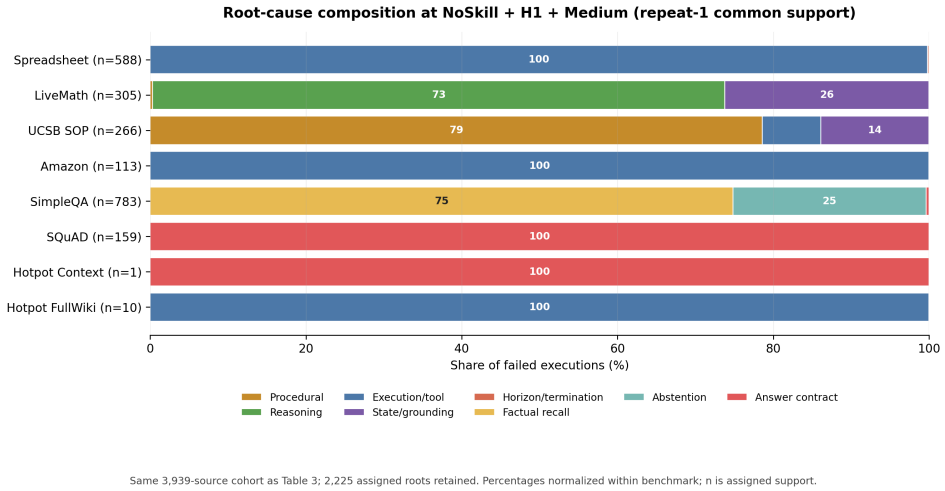


Figure 27: Root-cause composition of the 2,225 labeled NoSkill+H1+Medium repeat-1 source failures used by Table 23.

overlap used below. Root and benchmark are therefore strongly associated by construction, and the Reasoning, State, Factual, and Abstention assignments are model-assisted hypotheses rather than deterministic labels.

Link each source root to matched intervention opportunity. For each labeled failure we ask whether changing one axis improves that same task instance. Table 23 averages candidate-specific positive-change indicators over the three Skill and three Model alternatives, with H2 and High Budget each testing one action. Contract-labeled SQuAD failures respond to H2 in 77.3% of cases, and SimpleQA's factual-recall failures respond to a Model change in 30.1% against 6.7% for H2. Notably, the Model column is the largest entry in seven of the eight rows, including Procedural, where the Skill average is 12.5% against 23.3% for Model. A root label can thus suggest an axis without identifying the best action, and no row supports a Skill-specific rule.

Because root and benchmark are strongly associated, the pooled rows alone cannot show that the label adds information beyond benchmark membership. We therefore prespecify the within-benchmark overlap of LiveMath, SimpleQA, and UCSB SOP (1,350 failures). A benchmark- and source-model-stratified permutation test finds a root $\times$ intervention association ($p = 0.0002$). In grouped out-of-fold prediction, adding Root to Benchmark+Model improves AUPRC by 3.69 points (95%

Table 23: Pooled descriptive root cause and action-count-standardized matched intervention opportunity. Skill and Model average candidate-specific improvement indicators over three actions; H2 and High budget each test one action. Columns are overlapping improvement opportunities rather than mutually exclusive outcomes, so rows need not sum to 100%. Improvement is a strictly positive matched native-score change ($\delta_b = 0$), preserving the original estimand. Each axis has total weight one despite different candidate counts. Bold marks the largest standardized opportunity in each row; underlining marks the second largest. †The single Horizon case is descriptive.

H1 root cause	Failures	Skill avg.	H2	High budget	Model avg.
Procedural	210	12.5%	7.1%	3.8%	**23.3%**
Reasoning	224	18.0%	10.7%	9.4%	**23.2%**
Execution/tool	730	5.0%	20.8%	4.0%	**26.6%**
State/grounding	117	22.8%	12.0%	10.3%	**45.6%**
Horizon/termination†	1	33.3%	0.0%	0.0%	**100.0%**
Factual knowledge/recall	586	7.8%	6.7%	6.3%	**30.1%**
Abstention/calibration	194	2.7%	2.1%	2.6%	**16.2%**
Evaluation/answer contract	163	63.0%	**77.3%**	1.8%	76.7%

CI [+1.69,+5.75]), but changes AUROC by only -0.04 points (95% CI [-1.31,+1.26]) and reduces log loss by 0.0040 with an interval that includes zero (95% CI [-0.0008,+0.0090]). This is a partial positive and partial null result: failure labels carry localized information beyond benchmark identity on the observed overlap support, but the signal is not metric-general or stable enough to define a transferable cross-benchmark routing rule (Appendix C.6.2).

A separate outcome-blind root-to-action mapping gives the same qualified picture. H2 is comparatively responsive for Execution/tool, State/grounding, and Evaluation/answer contract failures (+6.54 points over the non-aligned mean, 95% CI [+2.21,+10.77]), while a Model change is comparatively responsive for Reasoning, Factual knowledge/recall, and Abstention/calibration failures (+40.78 [+32.28,+49.11]). Procedural→Skill receives no universal support, and the mapped action does not always beat the strongest individual alternative (Appendix C.6.3).

C.6.2 Benchmark-Conditioned Root–Intervention Check

The root-composition, root-opportunity, and benchmark-marginal tables use different estimands and cannot be algebraically composed. Table 14 conditions on labeled failures from the fixed NoSkill+H1+Medium source; Table 23 measures matched positive changes on those failures; and Table 13 averages broader configuration support. Their correspondences therefore provide a mechanism-level interpretation rather than a reconstruction of the marginal effects.

The benchmark selection deliberately gave each suite a distinct dominant failure mode, and six of the eight are accordingly single-root. That success is what forces this check to be conditional: where a benchmark has one root, root and benchmark identity are collinear and a pooled test cannot tell which is doing the work. We therefore require each included benchmark to contain at least two root categories with at least 20 source failures each. This retains 1,350 failures: LiveMath (Reasoning and State/grounding; $n = 304$), SimpleQA (Factual knowledge/recall and Abstention/calibration; $n = 780$), and UCSB SOP (Procedural, Execution/tool, and State/grounding; $n = 266$). All other benchmarks are excluded from this identification check because their source failures are effectively single-root under the frozen protocol.

Within this support, we shuffle roots 5,000 times inside benchmark × source-model strata. The omnibus root × intervention profile differs from the stratified null ($p = 0.0002$); axis-wise associations are present for Skill ($p = 0.0092$), H2 ($p = 0.0302$), and Model ($p = 0.0002$), but not High budget ($p = 0.1056$). Figure 28 shows that most large within-benchmark differences arise in UCSB SOP, whereas LiveMath differences are smaller.

Together, the significant stratified association and AUPRC gain show that roots are not merely renamed benchmark IDs on the overlap support. However, the absent AUROC gain and inconclusive log-loss improvement fail the metric-general incremental criterion. We therefore treat root labels as a

Table 24: Task-grouped OOF incremental prediction on the 1,350-source overlap support. Both models control benchmark, source model, and intervention-action identity; the extended model adds Root and Root × Intervention. Evaluation gives equal mass to every benchmark and intervention axis. Intervals are paired task-cluster bootstrap intervals.

Model or contrast	Log loss ↓	AUPRC ↑	AUROC ↑
Benchmark + Model	0.3592	32.36%	72.13%
Benchmark + Model + Root	0.3552	36.05%	72.09%
Root increment	+0.0040 reduction	+3.69 pp	−0.04 pp
95% CI	[−0.0008, +0.0090]	[+1.69, +5.75]	[−1.31, +1.26]

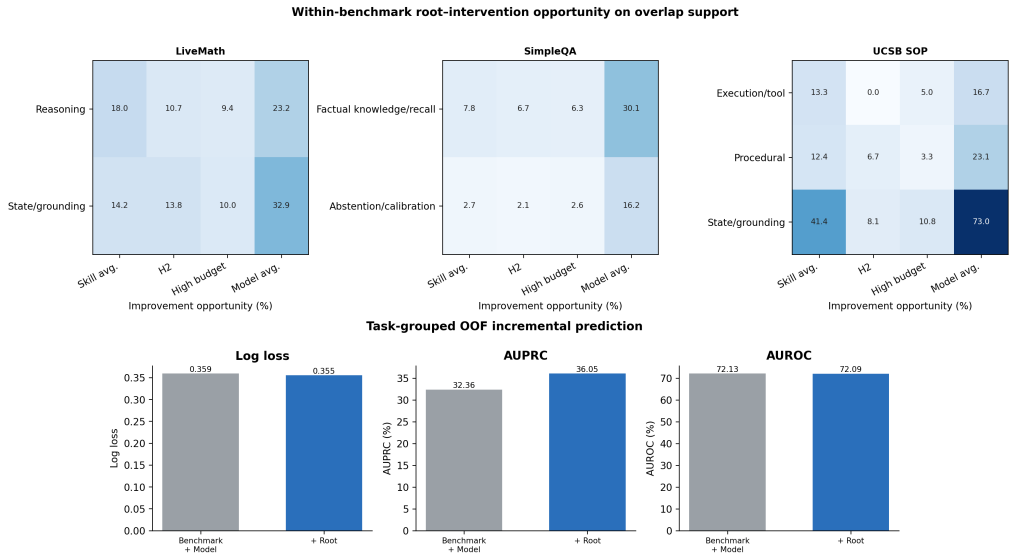


Figure 28: Benchmark-conditioned root analysis. Top: within-benchmark action-count-standardized opportunity for root categories satisfying the prespecified overlap threshold. Bottom: grouped OOF prediction with and without Root. Root improves AUPRC but not AUROC, and its log-loss interval crosses zero; the result supports localized association rather than stable benchmark-independent prediction.

localized explanatory signal beyond benchmark identity on the observed overlap support, but not as a validated universal or cross-benchmark predictor of intervention sensitivity.

C.6.3 Frozen Root-Aligned Intervention Contrast

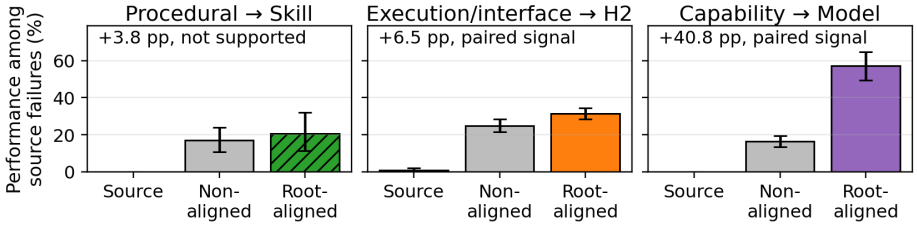


Figure 29: Root-aligned versus non-aligned intervention performance. For each frozen source-failure pattern, the bars show the failed-source score, the mean score of the three non-aligned axes, and the score of the semantically mapped action. Scores are paired within task, averaged within benchmark, and then benchmark-equal averaged. Error bars are 95% paired task-cluster bootstrap intervals for the aligned-minus-non-aligned contrast. Root-aligned means taxonomy-aligned, not empirically optimal.

Table 25: Compact root-aligned intervention contrast. n/B gives source failures and contributing benchmarks. The non-aligned column averages the other three active intervention axes. A paired signal requires a positive bootstrap lower bound, BH $q < .05$, and at least two benchmarks; it remains descriptive rather than causal. Horizon/termination is omitted because its single source case cannot support a pattern-level preset. Source: Qwen3.5-9B + NoSkill + H1 + Medium, repeat 1. Same-configuration rerun is kept separate from the three-axis non-aligned mean. High remains an action-level comparison but is not assigned a root pattern.

Failure pattern	n/B	Source	Non-aln.	Root-aligned action	Aligned−mean [95% CI]	Decision
Proc.→Skill	53/1	0.00	16.98	Skill: 20.75	+3.77 [−5.66,+14.47]	Not supported
Exec./interface→H2	465/7	0.99	24.88	H2: 31.42	+6.54 [+2.21,+10.77]	Paired signal
Capability→Model	314/2	0.00	16.46	GPT-5.6: 57.24	+40.78 [+32.28,+49.11]	Paired signal

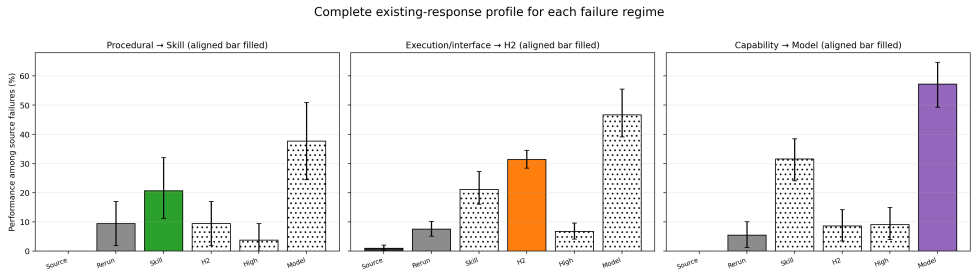


Figure 30: Complete action profile behind the compact root-aligned contrast. Every panel shows the failed source, same-configuration rerun, Trace2Skill, H2, High budget, and GPT-5.6 target performance. The outlined bar is the action fixed by the semantic mapping; a high non-aligned mean can therefore be traced to its individual action.

Table 26: Complete performance table for the frozen root-aligned intervention contrast. Scores are paired within task and benchmark-equal aggregated. "Strongest other" is reported to distinguish a taxonomy-aligned signal from empirical action optimality. Performance values are percentages; blue bold values mark the intervention fixed by the root-to-axis mapping. B is the number of contributing benchmarks. Bootstrap p values are .4560, .0032, and .0000 in row order; the corresponding BH q values are shown in the final column. Horizon/termination is excluded because $n = 1$ cannot support a pattern-level preset.

(a) Per-action performance

Failure pattern	n	B	Source	Rerun	Skill	H2	High	Model
Proc.→Skill	53	1	0.00	9.43	**20.75**	9.43	3.77	37.74
Exec./interface→H2	465	7	0.99	7.58	21.19	**31.42**	6.75	46.70
Capability→Model	314	2	0.00	5.51	31.60	8.70	9.07	**57.24**

(b) Aligned-versus-alternative contrasts

Failure pattern	Other mean	Aligned−mean [95% CI]	Strongest other	Aligned−strongest	BH q
Proc.→Skill	16.98	+3.77 [−5.66,+14.47]	Model 37.74	−16.98	.4560
Exec./interface→H2	24.88	+6.54 [+2.21,+10.77]	Model 46.70	−15.28	.0048
Capability→Model	16.46	+40.78 [+32.28,+49.11]	Skill 31.60	+25.64	.0000

The outcome-blind mapping fixes Procedural→Skill, Execution/tool/State/answer contract→H2, and Reasoning/Factual knowledge/Abstention→Model before loading matched target outcomes. Figure 29 and Table 25 show the compact contrast on 832 Qwen NoSkill+H1+Medium repeat-1 failures with complete targets. The H2 and Model mappings exceed the non-aligned three-axis mean; Procedural→Skill is unsupported. The H2 mapping does not exceed the strongest other individual action, and capability has support from only two benchmarks. Figure 30 exposes each action. The reconstruction uses existing target executions and is descriptive rather than prospective randomized intervention.

Table 27: Available cost fields for the frozen root-aligned contrast. Missing values mean that comparable full-chain telemetry was unavailable; they are not treated as zero. [a]A reproducible $1 per million accounted-token scenario, not an invoice. [b]Actual provider attribution is available only when GPT-5.6 is the aligned target; it excludes the local-Qwen source cost. Qwen latency and actual dollar cost were not logged comparably.

Failure pattern	Aligned tokens	Other-axis tokens	Aligned USD[a]	Provider USD / latency[b]
Proc.→Skill	NA	NA	NA	NA
Exec./interface→H2	21,966.28	19,107.73	$0.021966	NA
Capability→Model	4,380.55	6,903.82	$0.004381	$0.017542 / 15.80 s

C.7 When Feedback Becomes Recovery

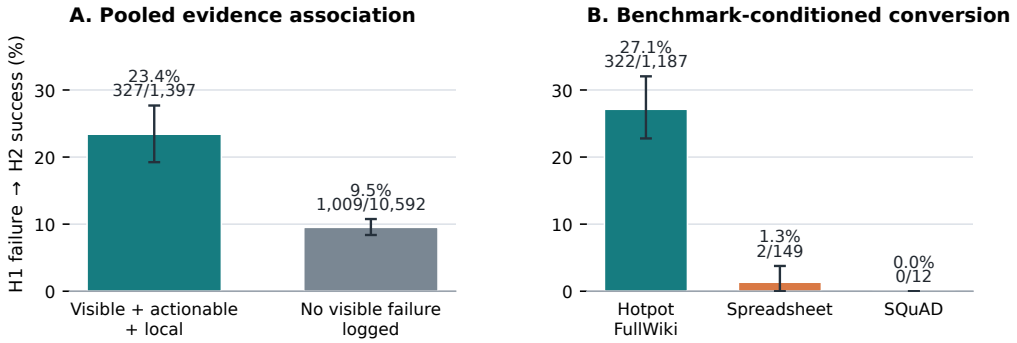


Figure 31: H1→H2 rescue conditional on evidence recorded in the failed H1 trajectory. (a) Pooled binary failures with visible, actionable, locally correctable evidence have a higher rescue rate than failures with no visible signal logged. (b) The same evidence pattern converts very differently across benchmarks, and Hotpot FullWiki supplies 85.0% of the pooled exposed-failure denominator. Whiskers are 95% instance-UID cluster-bootstrap intervals; the interval is omitted for SQuAD because none of its 12 cases is rescued. "No visible failure logged" describes available telemetry rather than proving that no usable signal existed.

Usable evidence is associated with recovery, but logged observability is not sufficient. H1→H2 provides a concrete test of when feedback becomes repair. Mechanistically, H2 can help only if the failure is exposed to the model, the resulting signal identifies a legal corrective action, and the error can be repaired locally without restarting the whole solution:

$$\boxed{\text{exposure} \longrightarrow \text{repair} \longrightarrow \text{preservation.}}$$

We operationalize the first two stages using whether an H1 failure is logged as model-visible, actionable, and locally correctable. These telemetry fields are imperfect proxies: absence of a logged visible failure does not establish that no usable signal existed.

Among H1 failures satisfying all three recorded conditions, H2 rescues 23.4% (327/1,397), compared with 9.5% (1,009/10,592) when no visible failure is logged. The pooled difference is consistent with the observability hypothesis, but it is dominated by one benchmark. Hotpot FullWiki supplies 1,187 of the 1,397 exposed failures (85.0% of the denominator) and rescues 27.1% of them, whereas Spreadsheet rescues only 2 of 149 (1.3%) and SQuAD 0 of 12 (Figure 31). The pooled 23.4% is therefore largely a weighted restatement of the retrieval case rather than evidence of a portable exposure-to-repair mechanism. Logged observability is associated with recovery, but it is not sufficient; its necessity also remains unproven because the logs may miss usable evidence.

Rescue alone is insufficient. A recovery intervention must repair failed executions without destroying executions that were already correct. We therefore report both

$$\text{RescueRate} = P(Y^{H2} = 1 \mid Y^{H1} = 0) \quad \text{and} \quad \text{RegressionRate} = P(Y^{H2} = 0 \mid Y^{H1} = 1),$$

with

$$\text{NetGain} = P(Y^{H1} = 0, Y^{H2} = 1) - P(Y^{H1} = 1, Y^{H2} = 0)$$

over the full matched population. A successful repair that is overwritten later in the trajectory fails the preservation stage. H2 produces both rescues and recovery-induced regressions, so its utility is determined by the complete exposure–repair–preservation pathway rather than rescue alone (Appendix Figure 32 and Tables 28–30).

A retrospective Qwen pre-H2 predictor attains AUROC 0.945 against 0.796 for benchmark identity, but this does not establish a deployable recoverability signal. Its features aggregate repeated H0 and H1 history for the same task instance while its label is whether H2 rescues that instance, so it has partial access to how the instance behaves under repetition and the gap to a single-trajectory predictor is unmeasured. It is an upper bound contaminated by task-instance identity information; the deployable numbers are the weaker ones in the router ablations (Appendix Figure 33).

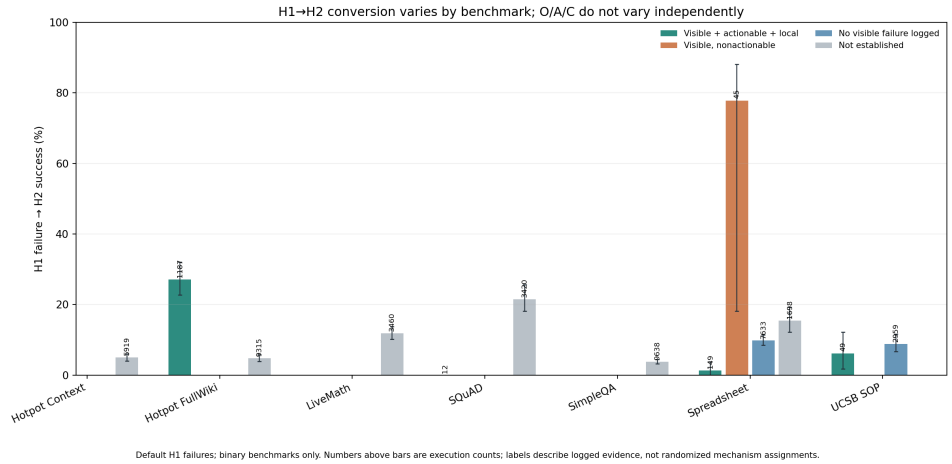


Figure 32: H1→H2 recovery conditional on audited model-visible evidence. A visible failure signal can be nonactionable, and even failures logged as visible, actionable, and locally correctable are not uniformly rescued. The pooled rescue rate is higher than when no visible failure is logged, but conversion varies sharply by benchmark. These attributes are observed jointly, so the figure does not identify their separate causal effects.

Table 29: Action-set Oracle opportunity on the same 2,225 repeat-1 source failures as Table 23. Skill and Model count a source as improved when any of three tested actions improves it; H2 and High budget each test one action. Candidate counts differ, so columns quantify action-set headroom and must not be used to rank intervention axes. A source counts as improved when any tested action on that axis improves it; Skill and Model each test three actions while H2 and High budget test one. These are overlapping retrospective action-set opportunities, not causal effects or router accuracy. †The single Horizon case is descriptive.

H1 root cause	Failures	Skill (any)	H2	High budget	Model (any)
Procedural	210	21.0% (44)	7.1% (15)	3.8% (8)	41.0% (86)
Reasoning	224	37.5% (84)	10.7% (24)	9.4% (21)	47.8% (107)
Execution/tool	730	11.2% (82)	20.8% (152)	4.0% (29)	47.0% (343)
State/grounding	117	42.7% (50)	12.0% (14)	10.3% (12)	66.7% (78)
Horizon/termination†	1	1/1	0/1	0/1	1/1
Factual knowledge/recall	586	14.0% (82)	6.7% (39)	6.3% (37)	58.7% (344)
Abstention/calibration	194	6.7% (13)	2.1% (4)	2.6% (5)	40.2% (78)
Evaluation/answer contract	163	81.6% (133)	77.3% (126)	1.8% (3)	91.4% (149)

For matched H1→H2 transitions, let $Z = 1$ indicate audited model-visible feedback/recovery exposure. For binary outcomes,

$$\mu(H2) - \mu(H1) = \Pr(Y_1 = 0, Y_2 = 1, Z = 1) \tag{7}$$

$$+ \Pr(Y_1 = 0, Y_2 = 1, Z = 0) - \Pr(Y_1 = 1, Y_2 = 0), \tag{8}$$

corresponding to exposed gain, unexposed gain, and recovery-induced regression.

Table 28: Exact values summarized by Figure 31. Rescue counts are exact matched H1 failure→H2 success transitions. Benchmark rows condition on visible, actionable, locally correctable evidence. "No visible failure logged" does not establish that no usable failure signal existed; it records only the available telemetry.

Scope and H1 evidence	Rescues	H1 failures	Rate
Pooled: visible + actionable + local	327	1,397	23.4%
Pooled: no visible failure logged	1,009	10,592	9.5%
Hotpot FullWiki: visible + actionable + local	322	1,187	27.1%
Spreadsheet: visible + actionable + local	2	149	1.3%
SQuAD: visible + actionable + local	0	12	0.0%

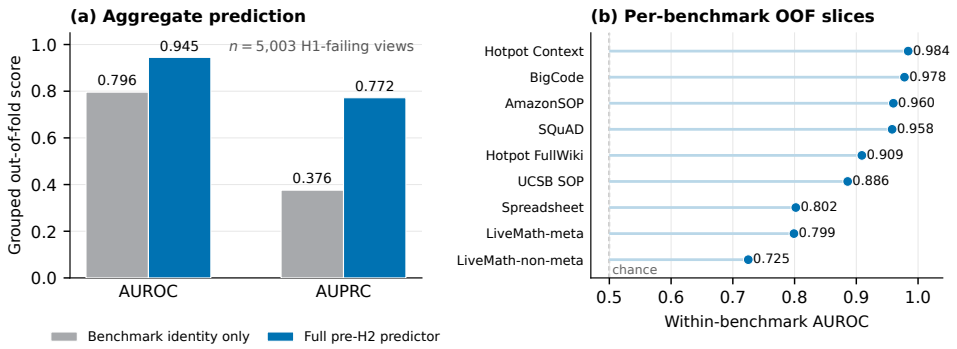


Figure 33: Retrospective H2-rescue prediction among 5,003 Qwen task–method views that fail under H1. Panel (a) compares benchmark identity with the full pre-H2 behavioral predictor under task-grouped five-fold cross-validation. Panel (b) reports within-benchmark grouped out-of-fold AUROC for benchmarks containing both classes; SimpleQA is omitted because it has no rescued views. The features aggregate repeated H0/H1 behavioral history, so this is diagnostic evidence of recoverability signal rather than a deployable single-trajectory router.

Table 30: Matched H1→H2 transitions at Default budget for binary-outcome benchmarks. Rescue and regression rates condition on H1 failure and H1 success, respectively; Net gain is $(N_{0\to1} - N_{1\to0})/N$. Amazon uses a fractional metric and is not dichotomized: among 2,400 matched executions, 1,070 improve, 1,322 are unchanged, eight degrade, and mean performance rises from 40.47 to 82.71 (+42.24 points).

Benchmark	N	Rescue	Preserve	Regr.	Persist.	Rescue %	Regr. %	Net
Spreadsheet	13,440	1,046	3,514	401	8,479	10.98%	10.24%	+4.80 pp
LiveMath	5,952	410	2,050	442	3,050	11.85%	17.74%	−0.54 pp
UCSB SOP	8,400	265	5,143	249	2,743	8.81%	4.62%	+0.19 pp
SimpleQA	14,400	365	4,366	396	9,273	3.79%	8.32%	−0.22 pp
SQuAD	14,400	736	10,828	140	2,696	21.45%	1.28%	+4.14 pp
Hotpot Context	14,400	294	8,201	280	5,625	4.97%	3.30%	+0.10 pp
Hotpot FullWiki	14,400	770	3,466	432	9,732	7.33%	11.08%	+2.35 pp

Interpretation boundary. The taxonomy localizes the earliest supported source error, whereas the matched target outcomes indicate whether a tested intervention subsequently improves that case. The association does not establish that the named root cause is the sole causal mediator, and labels produced from answer-only or model-assisted evidence remain hypotheses. The transition analysis likewise describes replayed matched executions; it does not constitute prospective continuation from one shared H1 state.

C.8 What Capability Boundaries Remain

For failed task instance i under source configuration c, let $\mathcal{A}_{i,J}$ be the available actions on axis $J \in \{S, H, B, M\}$ and τ_i the benchmark-native success criterion. To equalize candidate count, one action is selected uniformly per axis and the crossing set is

$$a_{i,J} \sim \mathrm{Unif}(\mathcal{A}_{i,J}), \qquad \mathcal{B}_i(\mathbf{a}_i) = \{J : Y_i(c, a_{i,J}) \geq \tau_i\}\,. \tag{9}$$

We enumerate all 3×3 Skill–Model pairs exactly; H2 and High budget are singleton actions. Reported boundary prevalence is the uniform expectation over these one-action-per-axis comparisons. If $\mathcal{B}_i$ is empty, no selected intervention reaches success; if several axes cross, deployment cost is required to identify the cheapest successful action in the full action set. The boundary is therefore empirical and configuration-dependent rather than an intrinsic hierarchy among intervention classes.

C.8.1 Uniform Boundary Decomposition

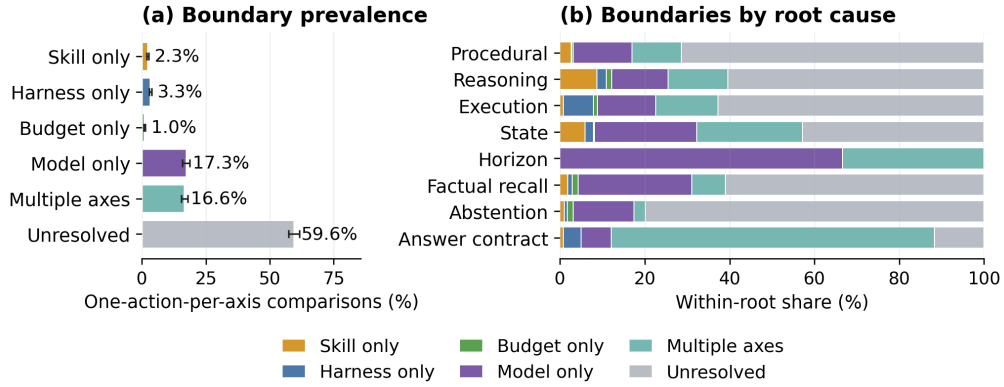


Figure 34: Equal-cardinality empirical capability boundaries on 2,225 labeled source failures. Each comparison selects one action per axis; all 3×3 Skill–Model pairs are enumerated uniformly, with H2 and High budget as singleton actions. An axis crosses when its selected action reaches native success. Bars in panel (a) include 95% task-cluster bootstrap intervals.

Figure 34 gives the uniform decomposition. In 59.6% of comparisons, none of the tested axes reaches success. Model is the only successful axis in 17.3%; multiple axes succeed in 16.6%; and Harness-only, Skill-only, and Budget-only crossings account for 3.3%, 2.3%, and 1.0%, respectively. Across this standardized distribution, most failed-source/action comparisons therefore remain unresolved; a failed source is not reliably locally recoverable merely because one action from each tested axis is available. At the same time, a substantial subset admits more than one valid intervention, so the capability boundary also identifies cases in which the same outcome may be reached at different costs.

These are empirical boundaries of the tested tasks, source configurations, action sets, and success thresholds, rather than an intrinsic hierarchy of Model, Skill, Harness, and Budget.

C.9 Can Intervention Choice Be Routed

Table 31: Direct routing without a paid H1 probe under held-family historical replay. The Observed Oracle is an unattainable reference using matched outcomes.

Policy	Performance	Mean full-chain tokens
Trace H1	38.99%	2,853.51
Trace H2 reference	45.29%	4,070.32
Text+action Router	39.42%	3,620.89
Text→root→action	40.10%	4,264.12
Observed per-instance Oracle	58.12%	2,084.54

Table 32: Recovery routing under held-family historical replay. Full-chain token means charge the H1 probe and selected H2 restarts.

Policy	Performance	Mean full-chain tokens
Always H1	37.17%	5,334.19
Always H2	51.01%	10,243.14
H1-conditioned router	49.25%	13,677.49
Observed H1/H2 Oracle	54.20%	11,374.29

Table 31 evaluates routing without a paid H1 probe under held-family historical replay. Text→root→action slightly outperforms the Text+action Router (40.10% versus 39.42%), suggesting that predicted root-cause information may help routing, although the corresponding Stage-A difference is not statistically resolved. Both methods remain below the Trace H2 reference (45.29%), so an effective deployable router has not yet been established. The Observed per-instance Oracle nevertheless reaches 58.12%, revealing substantial routing headroom.

The recovery setting purchases one H1 execution and decides whether to keep it or restart with H2; the direct setting avoids that probe and chooses among tested Qwen configurations from task text. The first exposes an information-acquisition cost, while the second exposes an identifiability limit. All comparisons use complete recorded call-chain token accounting, including probes and restarts.

Table 33: Recovery-routing Stage-A input ablation under nested leave-one-family-out evaluation. Intervals use hierarchical task-cluster bootstrap. Values are percentages. Text+mechanism has the highest point estimates, but its paired AUPRC improvement over Text only and shuffled H1 has a 95% interval containing zero.

Input	AUPRC [95% CI]	AUROC [95% CI]
Text only	25.23 [11.77, 47.78]	52.74 [40.15, 66.88]
H1 trajectory only	22.84 [10.09, 45.59]	48.00 [38.65, 58.56]
Text+semantic H1	24.03 [10.70, 47.61]	47.05 [37.04, 59.45]
Text+mechanism H1	**32.33 [15.18, 60.33]**	**67.84 [56.34, 80.32]**
Text+shuffled H1	23.44 [10.70, 46.24]	49.39 [42.80, 64.03]

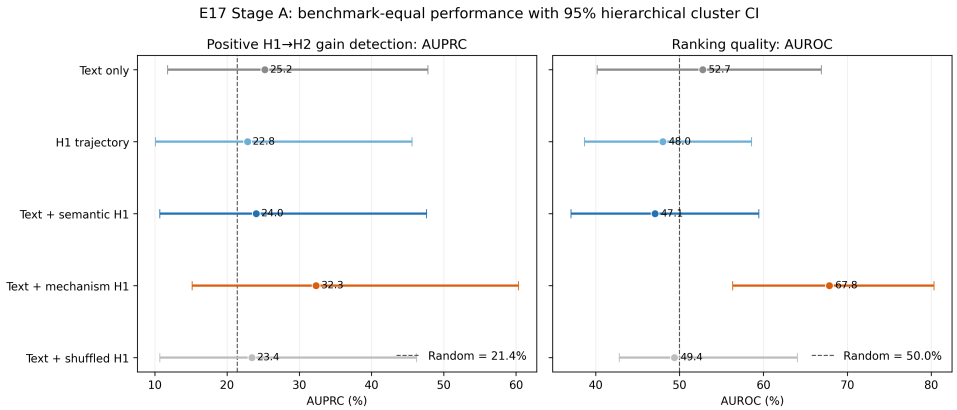
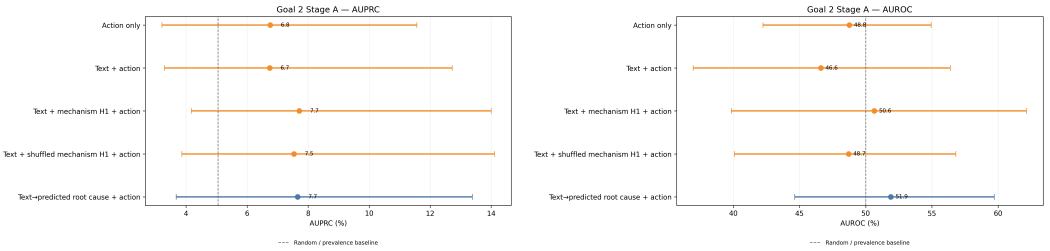


Figure 35: Recovery-routing Stage-A input ablation corresponding to Appendix Table 33. Points are benchmark-equal AUPRC and AUROC estimates; bars are 95% hierarchical task-cluster bootstrap intervals, and dashed lines mark the random baselines. Text+mechanism H1 has the strongest point estimates, but its paired AUPRC advantage over Text only and shuffled H1 is not statistically resolved.



(a) AUPRC **(b) AUROC**

Figure 36: Direct-routing Stage-A input ablation corresponding to Appendix Table 34. Bars are 95% hierarchical task-cluster bootstrap intervals and dashed lines mark the prevalence or chance baselines. H1-conditioned mechanism views are paid-probe diagnostics; the deployable text-predicted-root view does not show a statistically reliable advantage over direct Text+action prediction.

Table 34: Direct-routing Stage-A input ablation under nested leave-one-family-out evaluation. No current outcome, grader, gold answer, or future trajectory enters the direct inputs. Values are percentages. H1-conditioned rows are diagnostic paid-probe controls. The deployable text-predicted-root view does not significantly improve over Text+action.

Input	AUPRC [95% CI]	AUROC [95% CI]
Action only	6.76 [3.21, 11.57]	48.76 [42.24, 54.95]
Text+action	6.74 [3.29, 12.72]	46.61 [36.97, 56.39]
Text+mechanism H1+action	7.71 [4.18, 14.00]	50.65 [39.85, 62.15]
Text+shuffled mechanism H1+action	7.54 [3.86, 14.11]	48.71 [40.08, 56.81]
Text→predicted root+action	7.66 [3.67, 13.38]	51.89 [44.64, 59.72]

C.9.1 Routing Results and Interpretation

Routing headroom exists, but deployable signals do not yet realize it. RQ4 shows why routing could matter: some failures admit several successful axes with different costs, while others require a particular intervention. An Observed per-instance Oracle verifies that this heterogeneity contains substantial quality–cost headroom. On the direct action set, it reaches 58.12% with 2,085 mean full-chain tokens, compared with 45.29% and 4,070 tokens for the Trace H2 reference. Thus, a policy that knew the matched outcomes could be both more accurate and cheaper. The Oracle obtains that

advantage by reading the target outcomes it is supposed to predict, however, and is therefore an unattainable boundary rather than a deployable policy.

The tested deployment-valid information paths do not realize this headroom. The H1-conditioned router pays for an H1 probe before deciding whether to keep H1 or restart H2. It reaches 49.25% with 13,677 mean full-chain tokens, whereas Always H2 reaches 51.01% with 10,243 tokens. This router has both lower quality and higher cost and is therefore *strictly dominated*. Direct text-conditioned routing avoids the probe, but the Text+action Router reaches only 39.42%, and Text→root→action reaches 40.10%, both below the 45.29% Trace H2 reference. The latter also spends 4,264 tokens against 4,070 for Trace H2. Stage-A input ablations do not identify a reliable alternative signal: leave-one-family-out evaluation over eight benchmark families yields wide intervals, including AUPRC 25.23 [11.77, 47.78], so no input ranking is statistically supported (Appendix Tables 33 and 34).

The result is a negative deployment finding rather than evidence that routing has no value: routing headroom exists, but none of the tested lightweight deployable routing policies achieves a quality–cost Pareto improvement. In the terms of RQ2,

$$\text{failure information} \not\Rightarrow \text{a cheap deployable decision signal.}$$

These experiments are frozen historical replays rather than live prospective executions, and their money and latency ledgers remain incomplete. We therefore make no positive routing claim. Instead, the gap between the Observed Oracle and learned policies identifies the remaining problem: intervention selection is a value-of-information problem in which acquiring the signal can consume the benefit of acting on it.

C.9.2 TOKEN PRICE

Table 35: Model-specific API prices used by the provider-audited cost ledger, in USD per million tokens. Dollar cost is computed separately for each token class and summed over the complete call chain. Qwen was served locally and therefore has no observed per-token API price in these experiments. Prices are the public rates used to recompute the provider audit for the recorded US deployment profiles. †No Sonnet 5 cache-write tokens occurred in the audit. Prices should be versioned with the experiment because provider rates can change.

Model	Input	Output	Cache read	Cache write
Qwen3.5-9B	N/A	N/A	N/A	N/A
Claude Sonnet 4.6	$3.30	$16.50	$0.33	$4.125
Claude Sonnet 5	$2.20	$11.00	$0.22	N/A†
GPT-5.6 sol	$4.40	$22.00	$0.44	$5.50

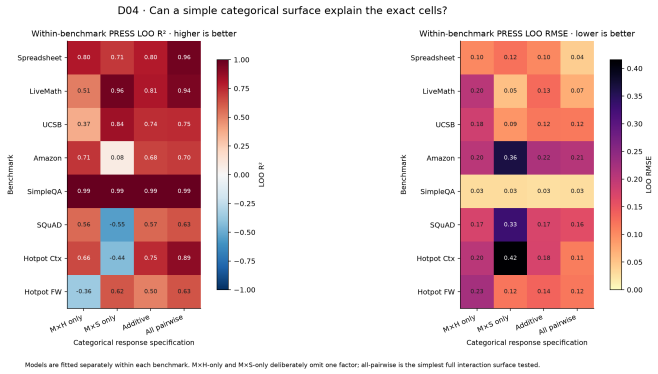


Figure 37: Categorical reconstruction of the Default response surfaces: leave-one-out R^2 for the additive and pairwise specifications. The diagnostic tests whether average component effects reconstruct held-out cells; it does not predict unseen benchmarks.

D BENCHMARK CASE STUDIES

This section reads the factorial results benchmark by benchmark. Task properties are the human-authored design annotations in Table 1; trajectory roots use the eight operational definitions in Table 22. Root frequencies refer only to the 2,225 labeled NOSKILL+H1+Medium source failures in Table 4; they are not benchmark labels, and the Hotpot estimates are too sparse for benchmark-wide interpretation. "Best" denotes the largest observed three-repeat mean in the Default-budget matrix (Table 2), with the per-model ceiling shown in Figure 19. It is descriptive selection over observed cells rather than a prospective policy.

Figures 37–43, including the held-cell check in Figure 38, collect the categorical reconstruction, discrete choice, paired-gain, and configuration-ranking analyses used below. The categorical models are evaluated by leave-one-cell-out (LOO) prediction within each benchmark. The rescue curves are descriptive regressions of paired gain on baseline score; because the baseline also enters the gain algebraically, their slopes are interpreted together with leave-one-benchmark-out (LOBO) performance and never as causal thresholds. Choice boundaries, regret, and configuration rankings use observed outcomes and therefore measure retrospective simplification or headroom rather than deployable routing performance.

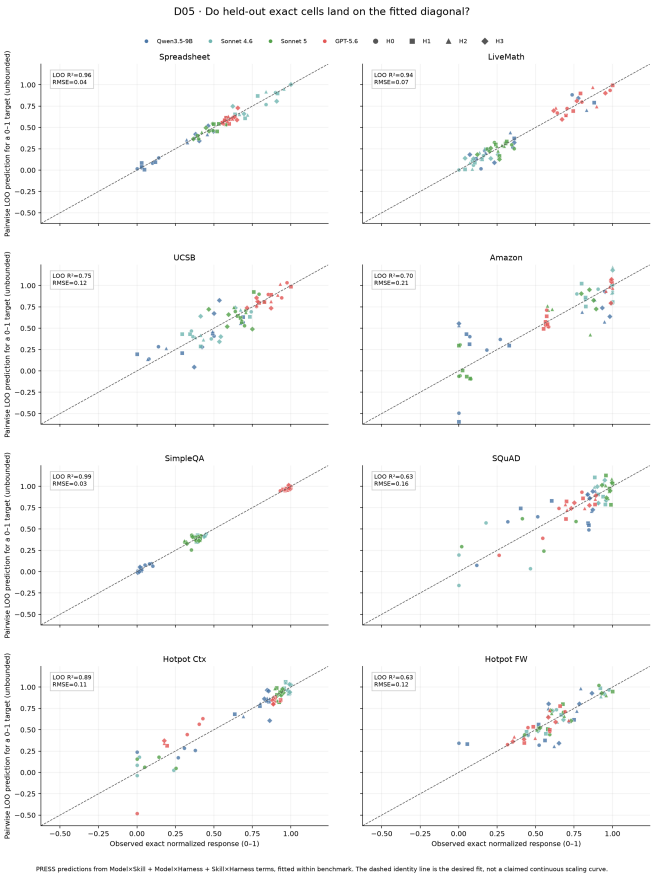
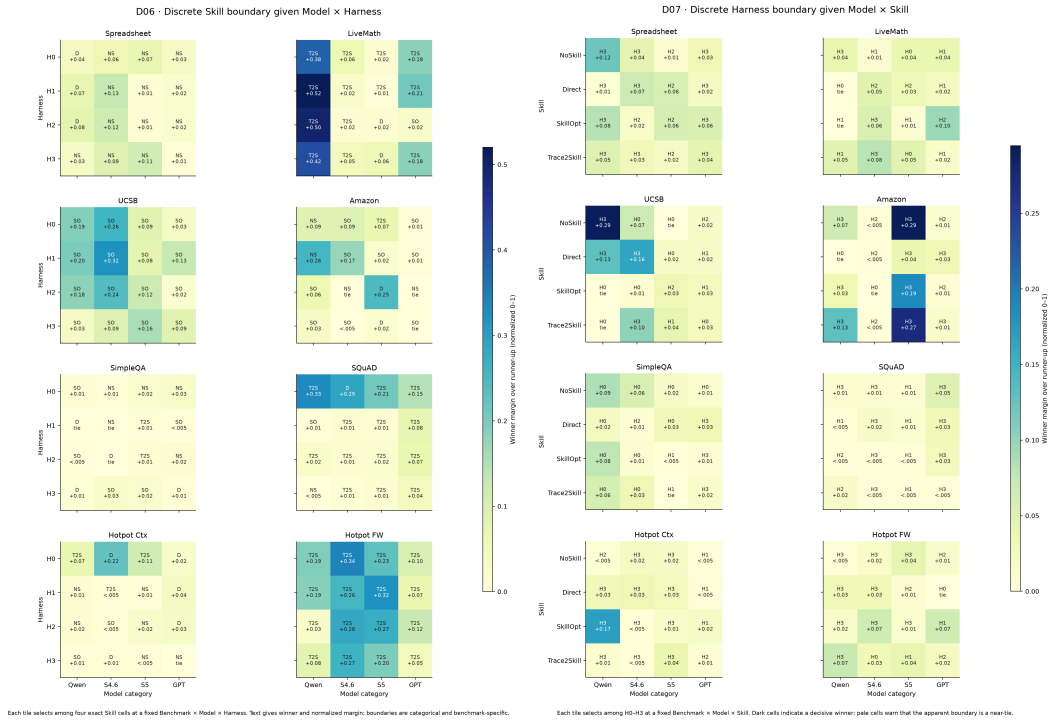
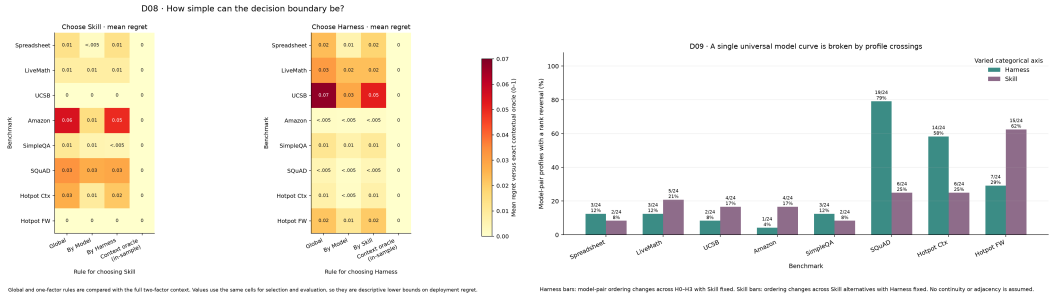


Figure 38: Observed Default cells against predictions from the all-pairwise model, on the same reconstruction as Figure 37.



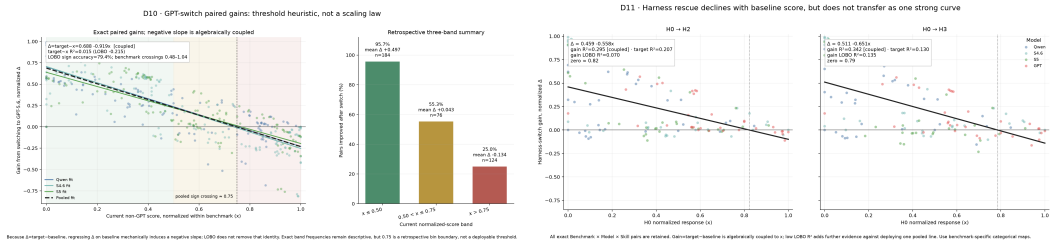
(a) Skill boundaries **(b) Harness boundaries**

Figure 39: Observed Default-budget choice boundaries. Each tile reports the best Skill at fixed Model×Harness or the best Harness at fixed Model×Skill, together with its margin. The maps summarize the measured matrix and are not deployment rules.



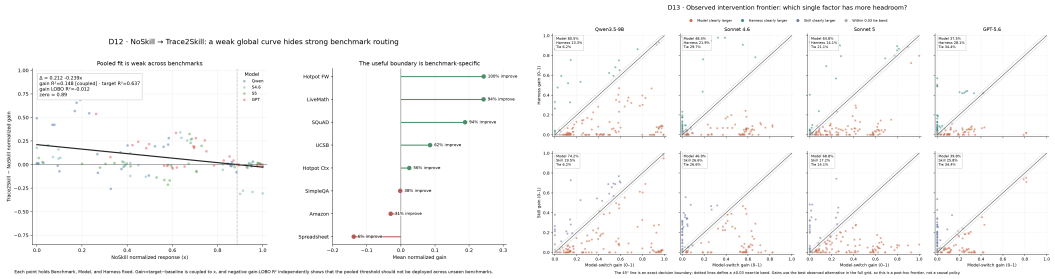
(a) Regret of simplified rules **(b) Model-rank reversals**

Figure 40: Discrete context dependence. (a) Retrospective normalized regret after simplifying the observed Skill or Harness boundary to a global or singly conditioned rule. (b) Fraction of model pairs whose ordering reverses as Skill or Harness changes.



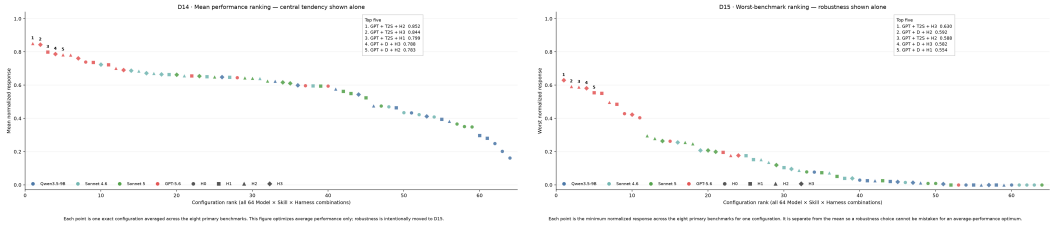
(a) Model-switch gain **(b) Harness-switch gain**

Figure 41: Paired rescue curves. (a) Switching a non-GPT Model to GPT-5.6. (b) Switching H0 to H2 or H3. Baseline–gain coupling and weak LOBO transfer limit both plots to descriptive boundary evidence.



(a) Trace2Skill gain **(b) Intervention-gain frontier**

Figure 42: Observed intervention gains. (a) NoSkill→Trace2Skill gains pooled and summarized by benchmark. (b) Retrospective gain available from changing Model, Skill, or Harness at each source Model.



(a) Mean ranking **(b) Worst-benchmark ranking**

Figure 43: Retrospective configuration rankings over the eight comparable benchmarks after within-benchmark normalization. The mean-optimal configuration is GPT-5.6+Trace2Skill+H2; the maximin configuration is GPT-5.6+Trace2Skill+H3. These answer different objectives and neither is a context-free winner.

D.1 SPREADSHEETBENCH

Context, properties, and failures. SpreadsheetBench requires stateful edits through a spreadsheet tool. We therefore annotate high tool/state interaction and feedback demand, with Execution/tool as the expected bottleneck. The observed source labels agree: 587 of 588 labeled failures are Execution/tool, with one Horizon/termination case. This unusually concentrated composition supports an execution interpretation for this source condition, although it does not imply that every configuration fails for the same reason.

Best observed configuration and mechanism. The best Default cell is Sonnet 4.6+NOSKILL+H3 at 53.1. The same Model rises from 48.6 at H0 to 49.2/51.1/53.1 under H1/H2/H3, while every supplied Skill lowers its H3 score. Qwen shows a larger execution-support response: NOSKILL rises from 9.3 at H1 to 24.9 at H2 and 30.4 at H3. The pattern is consistent with feedback, bounded repair, and revision addressing runtime failures, while a frozen procedure can distract from the current workbook state.

Categorical and predictive evidence. Figure 37 gives LOO $R^2 = 0.798$ for Model×Harness and 0.710 for Model×Skill; the all-pairwise model reaches 0.965, confirming substantial context dependence beyond main effects. Harness and Skill change model ordering relatively infrequently (0.125 and 0.083; Figure 40), and a global NOSKILL rule has only 0.013 normalized Skill regret, while global H3 has 0.024 Harness regret. Figure 42 shows the corresponding negative mean NOSKILL→TRACE2SKILL gain (-0.138 normalized; only 6.2% of contexts improve).

Takeaway. Stateful spreadsheet work is primarily execution-limited in this matrix. A capable Model with observable execution and revision is more useful than an additional frozen procedure, and Skill injection should be treated as a possible source of interference.

D.2 LIVEMATHEMATICIANBENCH

Context, properties, and failures. LiveMath is a multiple-choice mathematical reasoning task with low tool demand and a recurring meta-option. Its source labels are 73.44% Reasoning and 26.23% State/grounding, both model-assisted hypotheses, plus 0.33% Procedural. The post-hoc meta-option partition in Appendix C.3.1 shows that a reusable answer-selection regularity exists inside the benchmark but does not cover all instances.

Best observed configuration and mechanism. The best Default cell is GPT-5.6+TRACE2SKILL+H1 at 62.1. For GPT-5.6, TRACE2SKILL scores 61.6/62.1/57.8/60.5 across H0–H3; for Qwen it scores 53.2/58.1/56.5/54.6, far above Qwen+NOSKILL at 34.1/34.9/34.4/36.3. In contrast, additional Harness support is flat or non-monotone, and Sonnet 5+TRACE2SKILL declines from 40.6 at H0 to 35.5 at H3. The benefit is therefore better explained by a reusable decision procedure plus model competence than by execution recovery.

Categorical and predictive evidence. Model×Skill alone gives LOO $R^2 = 0.965$, compared with 0.512 for Model×Harness and 0.942 for the larger all-pairwise specification (Figure 37); this is the clearest Model–Skill surface. Skill changes model ordering in 20.8% of pairs, and a global TRACE2SKILL rule has only 0.008 normalized regret. Figure 42 reports a mean normalized TRACE2SKILL gain of 0.243 with improvement in 93.8% of Model×Harness contexts. Observation 2 remains essential: on Sonnet 5, that average masks a $+6.60$-point meta-option effect and a -4.93-point non-meta effect.

Takeaway. LiveMath rewards a reusable but narrow decision regularity. Trace-derived procedures help broadly across observed contexts, yet subgroup reversal shows why the Skill should not be interpreted as a general increase in mathematical capability.

D.3 UCSB SOPBENCH

Context, properties, and failures. UCSB SOPBench requires following long, constraint-sensitive standard operating procedures while interacting with state. We annotate high procedural and tool demand. Among labeled source failures, 78.57% are Procedural, 13.91% State/grounding, and 7.52% Execution/tool, giving the most direct root-level support for a procedure-oriented Skill hypothesis.

Best observed configuration and mechanism. The best Default cell is GPT-5.6+SKILLOPT+H1 at 73.1, narrowly above H0 at 72.6; H2 and H3 fall to 71.6 and 70.3. Qwen shows the same Skill preference: SKILLOPT reaches 66.3 at H0/H1 versus 52.8/51.0 for NOSKILL. Outcome-guided optimization can encode ordering and verification constraints, whereas additional recovery or reflection is not uniformly useful once that policy is supplied.

Categorical and predictive evidence. Model×Skill gives LOO $R^2 = 0.845$, exceeding Model×Harness at 0.369; the all-pairwise model reaches only 0.745, indicating that extra terms can hurt held-cell prediction (Figure 37). Skill changes model ordering in 16.7% of pairs. Global SKILLOPT has zero normalized Skill regret, but Harness selection is less compressible: its global regret is about 0.07 and conditioning on Model lowers it to 0.029 (Figure 40).

Takeaway. For procedure- and constraint-heavy SOP execution, an outcome-optimized reusable procedure is the stable first intervention. Harness choice remains model-dependent and stronger recovery is not automatically better after the procedure is supplied.

D.4 AMAZON SOP

Context, properties, and failures. Amazon SOP is a long, state-changing inspection workflow with high procedure, tool, horizon, and feedback demand. All 113 labeled source failures are Execution/tool. Horizon is nevertheless a task-level design hypothesis rather than an observed root conclusion, because the frozen Medium-budget source cohort contains almost no labeled Horizon/termination failures.

Best observed configurations and mechanism. The Default ceiling is 100.0 and is non-unique. It is reached by Sonnet 4.6+NOSKILL+H2, all four Harnesses with Sonnet 4.6+SKILLOPT, Sonnet 4.6+TRACE2SKILL+H2, GPT-5.6+NOSKILL+H2, and GPT-5.6+SKILLOPT+H2. The decisive transition is often execution feasibility: Qwen+NOSKILL rises 33.9→88.6 from H1 to H2, and GPT-5.6+NOSKILL rises 57.1→100.0. A mismatched Skill can instead collapse the path: Qwen+DIRECT remains at 1.7 across H0–H3. The Low-to-best Budget envelope is 81.13 points, consistent with a long workflow crossing a completion threshold rather than smooth scaling.

Categorical and predictive evidence. Model×Harness gives LOO $R^2 = 0.711$ while Model×Skill gives 0.081, reflecting the sharp, model-dependent execution transition (Figure 37). Harness changes model ordering only 4.2%, but Skill changes it 16.7%. A global H3 rule has 0.001 normalized Harness regret, whereas Skill requires Model conditioning to reduce regret to 0.015 (Figure 40). Figure 42 places the average TRACE2SKILL gain slightly below zero (-0.030; 31.2% positive), so the tied 100-point cells do not imply a universal Skill benefit.

Takeaway. Amazon SOP is governed by path feasibility and execution reliability. H2 or sufficient Budget can unlock completion, but a mismatched procedure can prevent that path entirely; after feasibility is reached, several configurations saturate at the ceiling.

D.5 SIMPLEQA

Context, properties, and failures. SimpleQA asks for short factual answers without supplied evidence or meaningful tool interaction. The labeled source failures are 74.84% Factual knowledge/recall, 24.78% Abstention/calibration, and 0.38% Evaluation/answer contract; the first two categories are model-assisted hypotheses. This composition aligns with a Model intervention more directly than with a procedural or execution intervention.

Best observed configuration and mechanism. The best Default cell is GPT-5.6+NoSkill+H0 at 69.4. GPT-5.6 remains near this level under H1–H3 (67.8/68.0/68.8), whereas Qwen+NoSkill is 10.7 at H0 and 4.7 at H2. Neither execution feedback nor a frozen procedure supplies a fact absent from the task and not recalled by the Model. The within-axis Model span is 60.85 points, compared with Skill, Harness, and Budget envelopes of 0.13, 0.00, and 0.10.

Categorical and predictive evidence. All categorical specifications reconstruct held cells almost perfectly (LOO $R^2 = 0.992$ for Model×Harness and Model×Skill, 0.994 additive, and 0.993 all-pairwise), so interactions add no useful predictive structure. Model-rank reversals are low (0.125 under Harness and 0.083 under Skill). Figure 41 supplies the broader model-switch result: GPT-5.6 improves about 65% of 384 matched non-GPT comparisons, with 79.4% LOBO direction accuracy, although baseline–gain coupling prevents a universal escalation threshold.

Takeaway. When the missing resource is factual recall, changing the Model dominates procedural and execution support. SimpleQA is the clearest case in which additional Harness, Skill, or Budget mostly adds cost without supplying the missing information.

D.6 SQuAD

Context, properties, and failures. SQuAD supplies the evidence passage, so the primary challenge in this setup is returning a creditable answer through the expected interface. All 159 labeled source failures are Evaluation/answer contract. This is strong evidence for the audited source condition, but the native-score floor analysis in Appendix A.6.1 means large H0→H1 gains should be read as contract specification rather than reasoning recovery.

Best observed configuration and mechanism. The best Default cell is Sonnet 5+Trace2Skill+H1 at 91.0, followed closely by H2/H3 at 90.6/90.7. The interface transition is large: Qwen+NoSkill rises 10.7→36.6 from H0 to H1 and then 78.6/79.3 under H2/H3, while Sonnet 4.6+NoSkill rises from the 0.0 floor to 85.4 at H1. A Skill can partially supply an extraction/answer procedure at H0, but once the Harness specifies the contract its incremental value is often absorbed.

Categorical and predictive evidence. Harness changes model ordering in 79.2% of pairs, the largest rate of any benchmark, while Skill changes it 25.0% (Figure 40). The all-pairwise model improves LOO R^2 from 0.574 additive to 0.629, but Model×Skill alone is negative (-0.553), reflecting the dominant interface discontinuity. Figure 42 still shows positive Trace2Skill gain in 93.8% of observed contexts (mean 0.188 normalized), illustrating that Skill benefit and Harness absorption can coexist at different locations in the surface.

Takeaway. SQuAD primarily tests whether the system exposes and satisfies the answer contract. H1 removes a measurement/interface bottleneck; after that transition, additional execution support or Skill often has little incremental value.

D.7 HotpotQA Context

Context, properties, and failures. Hotpot Context supplies gold and distractor paragraphs, reducing retrieval demand while retaining multi-hop evidence integration. We annotate Evaluation/answer contract as the expected bottleneck. Only one source failure is labeled, also as Evaluation/answer contract, so root composition cannot explain the benchmark; this is a coverage limitation rather than confirming evidence.

Best observed configuration and mechanism. The best Default cell is Sonnet 4.6+Direct+H3 at 69.2. The main transition is H0→H1: that configuration rises 16.6→66.8, and Sonnet 4.6+NoSkill rises 0.0→66.2. Later Harness levels add little. Skill compatibility remains important: GPT-5.6+SkillOpt stays at 0.0/13.6/12.2/12.2, compared with 29.7/64.6/64.3/60.9 for Direct. The former is consistent with a supplied policy overriding the answer contract, not with a graded loss of reasoning ability.

Categorical and predictive evidence. The all-pairwise model reaches LOO $R^2 = 0.893$, compared with 0.745 additive, 0.657 for Model×Harness, and -0.435 for Model×Skill (Figure 37). Harness changes model ordering in 58.3% of pairs, second only to SQuAD. Global H3 has 0.007 normalized Harness regret, while the nominal global TRACE2SKILL Skill rule retains 0.027 regret and matches the cellwise oracle in only one quarter of contexts; it should therefore not be read as a stable Skill prescription.

Takeaway. When evidence is already provided, interface compatibility dominates retrieval support. H1 captures most of the observed gain, and an incompatible Skill can erase it; the sparse root labels do not support a benchmark-wide mechanistic claim.

D.8 HOTPOTQA FULLWIKI

Context, properties, and failures. Hotpot FullWiki uses the same question UIDs as Context but removes the supplied passages and permits retrieval from a frozen Wikipedia index. This creates multi-hop search, grounding, execution, and horizon demands. All ten labeled source failures are Execution/tool, but ten labels are far too few to characterize the benchmark because missingness is concentrated in the Hotpot views.

Best observed configuration and mechanism. The best Default cell is Sonnet 5+TRACE2SKILL+H1 at 45.0, compared with 30.9 for Sonnet 5+NOSKILL at H1. Qwen reveals the interaction mechanism: DIRECT rises from 3.9 at H1 to 28.4 at H2, SKILLOPT from 24.0 to 34.9, and TRACE2SKILL from 34.1 to 36.0. A reusable retrieval policy supplies what to search and how to integrate evidence; executable observations and recovery determine whether weaker configurations can carry out that policy. The strongest cell nevertheless uses H1, showing that more recovery is not universally necessary once Model and Skill are well matched.

Categorical and predictive evidence. Skill changes model ordering in 62.5% of pairs, the largest Skill-induced reversal rate, while Harness changes it 29.2% (Figure 40). Model×Skill reaches LOO $R^2 = 0.620$, whereas Model×Harness is negative (-0.364); all pairwise terms reach 0.629. A global TRACE2SKILL rule has zero normalized Skill regret, and Figure 42 reports a 0.243 mean normalized gain with improvement in every observed Model×Harness context. Harness rescue is more configuration-specific and does not transfer as a single benchmark-independent threshold.

Takeaway. FullWiki couples a reusable multi-hop retrieval policy with the ability to execute it. Trace2Skill is the stable Skill choice in the observed matrix, while Harness support is most valuable when the current Model–Skill pair cannot reliably realize that policy.

D.9 BIGCODEBENCH

Context, properties, and evidence boundary. BigCodeBench-Hard requires program synthesis and execution against evaluator-private tests. We annotate medium procedural and horizon demand, high tool interaction and feedback demand, and Execution/tool as the expected bottleneck. We use all 148 available instances across frozen construction, validation, qualification, and test roles (Table 10).

Why no comparable winner is reported. BigCode is retained as supporting appendix evidence because its evaluator environment is not fully comparable across Models. It is therefore excluded from the 512-cell Table 2, Figure 19, the categorical D04–D09 analyses, the rescue curves, the root-conditioned common-support cohort, and the eight-benchmark configuration rankings. Reporting a cross-model "best intervention combination" from those analyses would silently mix evaluator differences with intervention effects.

Interpretation. The task structure motivates executable feedback and Model capability as hypotheses, but the present evidence cannot distinguish them on the same matched scale used for the other eight benchmarks. A valid case analysis requires rerunning every Model under one frozen execution

image, scorer, dependency set, timeout policy, and task split, followed by the same three-repeat factorial and root-label protocol.

Takeaway. Code execution is a plausible feedback-sensitive setting, but evaluator comparability is a prerequisite for attributing gains to Model, Skill, Harness, or Budget. BigCode remains supporting evidence rather than a ninth point in the comparative mechanism claims.

D.10 CROSS-CASE SYNTHESIS

The cases separate four recurring structures without turning them into universal rules: missing factual knowledge favors Model changes; reusable decision, SOP, or retrieval regularities can favor Skills; observable runtime or answer-contract failures can favor Harness support; and long workflows can cross a Budget feasibility threshold. The categorical and boundary analyses show why these relations remain conditional. The mean-optimal observed configuration across the eight normalized benchmarks is GPT-5.6+TRACE2SKILL+H2 (mean 0.8515), whereas the maximin configuration is GPT-5.6+TRACE2SKILL+H3 (worst-benchmark score 0.6301, mean 0.8439). Relative to H2, the latter sacrifices 0.0076 mean performance for a 0.0417 increase in the worst-benchmark floor (Figure 43). Objective choice therefore remains part of intervention choice, even after the benchmark context is known.

Takeaway. Across cases, intervention value follows the bottleneck that remains in the current configuration. Benchmark identity narrows the plausible mechanism, but task-instance heterogeneity, component interactions, and measurement boundaries prevent a context-free intervention rule.