SKILLS ARE EXECUTOR-RELATIVE: DIAGNOSING AND BRIDGING THE WRITER–EXECUTOR GAP

Anonymous authors
Paper under double-blind review

ABSTRACT

Language agents reuse skills written by other models, but these skills do not always transfer effectively. We identify the Writer–Executor Gap: different executors benefit from different skills, and skills produced by a stronger writer do not necessarily lead to better execution. Task success alone does not explain why this gap occurs or how to address it. We propose SKILLDIAG, a diagnostic method grounded in an information-theoretic view of skill transfer, to examine three sources of failure: insufficient evidence in the original experience, important information omitted during skill writing, and difficulty using information already present in the skill. We further introduce SKILLTUTOR, which uses the executor's failure feedback to clarify and reorganize skill instructions, helping the executor use them more effectively. We conduct experiments with multiple language models across ALFWorld, WebShop and DAPO-Math. The results reveal the Writer–Executor Gap across diverse settings. SKILLTUTOR improves action prediction in several settings and end-to-end success on WebShop.

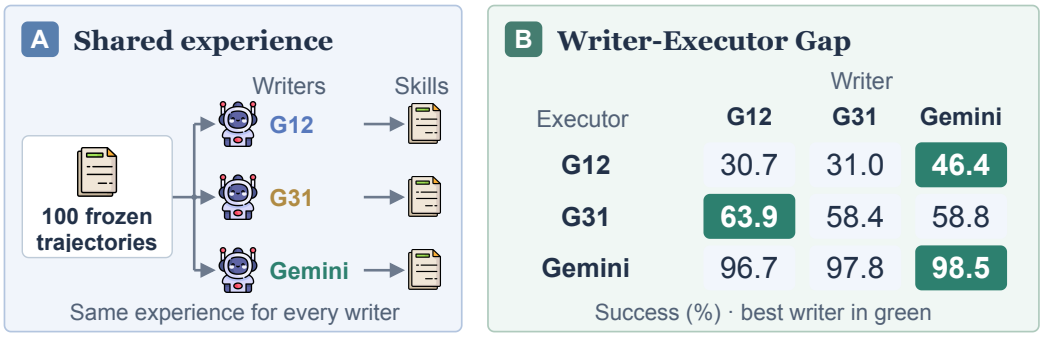


Figure 1: **The Writer–Executor Gap.** Three writers learn from the same 100 ALFWorld trajectories, yet their skills have different value for each executor. The best writer changes between Gemma-12B and Gemma-31B. Cells show success rates on 274 tasks; Table 4 includes model labels and no-skill baselines. Icons by Freepik/Magnific via Flaticon.

1 INTRODUCTION

Reusable skills allow language agents to retain experience and apply it to later tasks. A model can turn a successful trajectory into instructions for searching a website, manipulating an object or solving a mathematical problem, then share those instructions with another model. Reflection, skill libraries and workflow memory demonstrate the value of retaining experience (Shinn et al., 2023; Wang et al., 2023; 2025b), while skill benchmarks evaluate its effect without updating model parameters (Li et al., 2026; Zhong et al., 2026). Cross-model reuse raises a basic question: will the model reading a skill use it as effectively as its writer intended?

We identify the *Writer–Executor Gap*: the effectiveness of a written skill depends on the model executing it. Different executors can prefer different writers, and increasing writer size need not improve execution. Figure 1 gives a concrete example. After reading exactly the same 100 train-

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.

ing trajectories, Gemini produces the best-performing skill for Gemma-12B, whereas Gemma-12B produces the best-performing skill for Gemma-31B. Our Qwen experiments also show task-level writer-ranking reversals. These observations motivate studying skill utility as a relation $Q(S_w, E_e)$ between a skill and its executor.

Existing approaches encode experience as workflows, organize memory and revise behavior using execution feedback (Wang et al., 2025b; Hu et al., 2025; Wang et al., 2025a; 2026). Their task outcomes show whether these changes help, but a failure alone does not reveal what went wrong. The original experience may lack a necessary fact; the writer may omit an important step; or the executor may overlook a rule that is already present. These cases suggest different responses: obtain more evidence, restore missing content, or help the executor use the existing instructions.

We propose SKILLDIAG to examine these sources of failure. Its information-theoretic foundation views the writer as compressing experience into a skill and the executor as decoding that skill using its own knowledge and observations. A loss decomposition separates missing evidence, lost information and decoding difficulty. SKILLDIAG turns this distinction into paired measurements and controlled tests of evidence, content and presentation. We then introduce SKILLTUTOR to improve skill use. It studies the target executor's failures and constructs short teaching capsules that clarify or emphasize existing instructions. Candidate repairs are selected on separate data before evaluation on final tasks. Our contributions are:

- **Writer–Executor Gap.** We identify executor-dependent skill utility through full writer–executor matrices, including a cross-model experiment with a shared source of experience.
- **SKILLDIAG.** We propose an information-theoretic diagnostic method that examines evidence insufficiency, writing loss and decoding mismatch through paired scores and controlled interventions.
- **SKILLTUTOR.** We propose a repair method that learns from executor failures to improve skill use, and evaluate its effects on both individual decisions and task success.

2 RELATED WORK

Skill acquisition and reuse. Voyager and Agent Workflow Memory retain reusable routines and workflows (Wang et al., 2023; 2025b). AgentBank and AgentTrek instead curate or synthesize trajectories for agent training (Song et al., 2024; Xu et al., 2025). Reflection, hierarchical memory and step-level calibration improve long-horizon behavior (Shinn et al., 2023; Hu et al., 2025; Wang et al., 2025a). We study skill transfer between writers and executors; SKILLTUTOR adapts skill presentation using executor feedback.

Agent evaluation and diagnosis. ReAct interleaves reasoning and action (Yao et al., 2023); grounding and tool learning connect language to environments and APIs (Ahn et al., 2022; Schick et al., 2023; Qin et al., 2024). Benchmarks measure task outcomes (Liu et al., 2024; Jimenez et al., 2024; Mialon et al., 2024), while trajectory diagnostics inspect the decisions behind them (Ou et al., 2025). SKILLDIAG compares evidence and skill contexts under a fixed evaluation harness to diagnose skill transfer.

Information and decoding. Information theory distinguishes retained task information from decoder mismatch (Cover & Thomas, 2006; Tishby et al., 2000; Wyner & Ziv, 1976; Blackwell, 1953). We apply this distinction to explain the Writer–Executor Gap and motivate SKILLDIAG. Appendix B gives the theoretical assumptions and the relationship between ideal quantities and measured scores.

3 UNDERSTANDING THE WRITER–EXECUTOR GAP

The Writer–Executor Gap describes a mismatch between producing a skill and using it effectively. We study it by holding the task and evaluation protocol fixed and comparing the utility $Q(S_w, E_e)$ of each writer's skill across executors. Writer–executor interactions measure how a writer's advantage changes across executors; ranking reversals are a particularly direct manifestation. The communication model below explains why information preserved by a writer can still be difficult for an executor to use.

3.1 FROM EXPERIENCE TO SKILL TO ACTION

Let X be a source trace, S its written skill, and A^* a target action drawn from a prespecified law. Executor e receives side information $Y_e = (O, Z_e)$, where O contains shared observations and tool feedback and Z_e models executor-specific access to task-relevant knowledge. The writer is an encoder $p_\phi(s \mid x)$; the fixed executor is a decoder $q_e(a \mid s, y_e)$. We assume discrete variables, finite entropies and finite expected log-loss.

Definition 1 (Strict Skill Protocol)**.** *The encoder obeys $S - X - (A^*, Y_e)$, equivalently $p(x, a^*, y_e, s) = p(x, a^*, y_e)p_\phi(s \mid x)$.*

3.2 THREE SOURCES OF TRANSFER FAILURE

Define $\mathcal{L}_e(p_\phi) = \mathbb{E}[-\log q_e(A^* \mid S, Y_e)]$ and $D_e(p_\phi) = \mathbb{E}_{S,Y_e} D_{\mathrm{KL}}(p(\cdot \mid S, Y_e) \| q_e(\cdot \mid S, Y_e))$. The conditional cross-entropy identity (Cover & Thomas, 2006) yields the following decomposition.

Theorem 1 (Executor-relative decision loss)**.** *Under the Strict Skill Protocol,*

$$\mathcal{L}_e(p_\phi) = \underbrace{H(A^* \mid X, Y_e)}_{\textit{source insufficiency}} + \underbrace{I(A^*; X \mid S, Y_e)}_{\textit{compression loss}} + \underbrace{D_e(p_\phi)}_{\textit{decoding mismatch}} . \tag{1}$$

All three terms are nonnegative.

The three terms describe different points at which transfer can fail. *Source insufficiency* is uncertainty remaining even with the full experience and executor side information. *Compression loss* is decision information discarded when experience becomes a written skill. *Decoding mismatch* is the executor's excess loss when using the information that remains. A skill can therefore contain the required information without making it usable for a particular executor.

3.3 WHY A BETTER WRITER NEED NOT YIELD BETTER EXECUTION

Compare two writers p_1, p_0 on the same source distribution, target law and executor. Write $\Delta I_e = I(A^*; S_1 \mid Y_e) - I(A^*; S_0 \mid Y_e)$ and $\Delta D_e = D_e(p_1) - D_e(p_0)$.

Proposition 1 (Pairwise writer-loss identity)**.**

$$\mathcal{L}_e(p_1) - \mathcal{L}_e(p_0) = -\Delta I_e + \Delta D_e. \tag{2}$$

If writer 1 retains at least as much action-relevant information, it incurs higher loss exactly when $\Delta D_e > \Delta I_e$.

A gain in retained information can be outweighed by greater decoding difficulty (Eq. 2). Both terms depend on the executor, so the preferred writer can change. Appendix B provides full proofs, the conditional information bottleneck and a compatibility frontier describing the role of executor knowledge.

4 SKILLDIAG: DIAGNOSING SKILL TRANSFER

SKILLDIAG examines where skill transfer breaks down by comparing the use of source evidence with the use of its written skill. It combines two steps: paired scoring to expose different loss patterns, and controlled interventions to test their interpretation. Figure 2 connects this diagnostic procedure with SKILLTUTOR's repair process.

4.1 PAIRED EVIDENCE AND SKILL SCORES

On an interactive state h, let $\mathcal{A}(h)$ be the scored candidate commands and $\mathcal{A}^+(h)$ their verified successful-continuation subset. For context C, let $q_M^{\mathcal{A}}$ be the model's distribution normalized over these candidates. We measure

$$\ell(M, C; h) = -\log \sum_{a \in \mathcal{A}^+(h)} q_M^{\mathcal{A}}(a \mid h, C), \qquad L(M, C) = \mathbb{E}_t \mathbb{E}_{h|t} \ell(M, C; h). \tag{3}$$

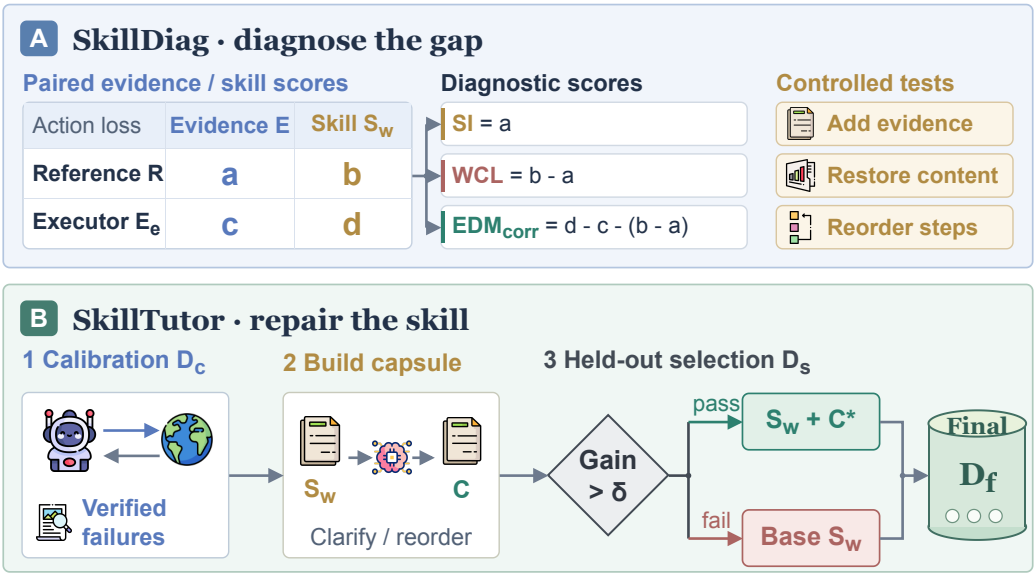


Figure 2: **Diagnosing and repairing the Writer–Executor Gap.** SKILLDIAG compares evidence and skill contexts and tests changes to evidence, content and order. SKILLTUTOR uses calibration failures to construct teaching capsules, selects a deployment on separate data, and evaluates it on held-out tasks. Icons by Freepik/Magnific via Flaticon; environment adapted from Wang et al. (2026).

Tasks receive equal weight. Fixed oracle-prefix states make the contexts comparable; free-running success is measured separately. With reference R, target executor E_e, masked evidence context E and skill S_w, we define diagnostics for source insufficiency (SI), writer compression loss (WCL) and executor decoding mismatch (EDM):

$$\widehat{\mathrm{SI}} = L(R, E), \qquad \widehat{\mathrm{WCL}}_w = L(R, S_w) - L(R, E), \tag{4}$$

$$\widehat{G}_e = L(E_e, E) - L(R, E), \qquad \widehat{\mathrm{EDM}}_{we}^{\mathrm{corr}} = L(E_e, S_w) - L(R, S_w) - \widehat{G}_e. \tag{5}$$

The four measured scores satisfy

$$L(E_e, S_w) = \widehat{\mathrm{SI}} + \widehat{\mathrm{WCL}}_w + \widehat{G}_e + \widehat{\mathrm{EDM}}_{we}^{\mathrm{corr}}. \tag{6}$$

The reference's evidence loss supplies the source-insufficiency diagnostic. Its skill-versus-evidence difference measures the writing-loss diagnostic. The target executor's additional skill penalty, after adjusting for its evidence-conditioned reference gap $\widehat{G}_e$, gives corrected decoding mismatch. These reference-relative contrasts can be negative. Appendix B.7 derives their relationship to the entropy and KL terms in Theorem 1.

4.2 TESTING THE DIAGNOSIS WITH CONTROLLED CHANGES

SKILLDIAG probes three distinct questions. *Evidence:* does adding independent source evidence reduce the reference's uncertainty? *Content:* does restoring omitted decision information help more than equally long repetition or unrelated content? *Presentation:* does reordering existing skill lines improve the target executor's use of them, relative to the reference? These controls distinguish changes in available information from changes in its use. The theoretical predictions appear in Appendix B.6; Section 6.3 reports their empirical responses.

5 SKILLTUTOR: REPAIRING SKILL TRANSFER

SKILLTUTOR uses the target executor's failures to construct short teaching capsules that supplement the original skill. Each capsule selects, restates or emphasizes relevant instructions. The source skill and executor weights remain frozen.

Table 1: Evaluation sets and statistical units. NLL states are aggregated within tasks.

Benchmark	Calibration / selection	Final evaluation
ALFWorld	60 / 30 train tasks	60 tasks, 168 states; 274 rollouts
WebShop	30 / 30 tasks	150 products, 599 states
DAPO-Math	100 discovery / 90 unseen	240 unseen problems

Constructing a repair. A tutor examines verified failures on calibration tasks D_c and proposes candidate capsules. For ALFWorld and WebShop, an audit checks each candidate against the source skill and rejects added task entities, hidden answers or unsupported rules. Math uses an extractive construction: the tutor selects atom IDs and the program concatenates their original lines.

Selecting and evaluating the repair. A disjoint selection set D_s compares audited candidates with the original skill (Base). The best candidate is deployed only if it meets the selection criterion; otherwise the deployment stays exactly Base. Final tasks D_f evaluate the frozen choice. Generic reminders and another executor's capsule for the same writer (Shuffled) provide controls. Algorithm 1 gives the basic procedure.

Algorithm 1 SkillTutor: repair and basic selection

Require: Frozen skill S_w, executor E_e, tutor T; disjoint D_c, D_s, D_f; metric m and gate g

$F \leftarrow \mathrm{VerifiedFailures}(E_e, S_w, D_c)$
$\mathcal{C} \leftarrow \{C \in \mathrm{Propose}(T, S_w, F) : \mathrm{SourceAudit}(C, S_w) = \mathrm{pass}\}$
$u_0 \leftarrow \mathrm{Evaluate}(E_e, S_w, D_s; m)$
$u_C \leftarrow \mathrm{Evaluate}(E_e, S_w + C, D_s; m)$ for each $C \in \mathcal{C}$
$C^* \leftarrow \arg\max_{C \in \mathcal{C}} \mathrm{Gain}_m(u_C, u_0)$, or $\varnothing$ if $\mathcal{C} = \varnothing$
$S^{\mathrm{deploy}} \leftarrow S_w$ ▷ Exact Base fallback
if $C^* \neq \varnothing$ and $g(u_{C^*}, u_0)$ accepts **then**
    $S^{\mathrm{deploy}} \leftarrow S_w + C^*$
end if
Freeze deployment; evaluate Base, Generic and Shuffled on D_f
Evaluate S^{deploy} on D_f, reusing Base outcomes after fallback
return Final outcomes and the gate decision

For action NLL, $\mathrm{Gain}_m(u_C, u_0) = u_0 - u_C$; for accuracy it is $u_C - u_0$. ALFWorld selects by lower task-mean NLL. Math requires at least two additional correct answers on 90 selection problems and no higher invalid-output count. WebShop audits category-specific candidates. The Gemini-Tutor extension adds a disjoint stability set D_v: after selection on D_s, deployment also requires the gain's paired-bootstrap 95% lower bound on D_v to exceed zero. The rule is frozen before a new final evaluation (Appendix P).

6 Experiments

Our experiments test the Gap across models and tasks, SkillDiag's response to controlled interventions, and SkillTutor's repair effects. The primary study uses Qwen3-4B-Instruct, Qwen3-8B and Qwen3-14B as writers and executors; 4B, 8B and 14B in the primary tables refer to these models, respectively. Qwen3-32B serves as reference/tutor, with thinking disabled (Yang et al., 2025). We evaluate ALFWorld, WebShop and DAPO-Math (Shridhar et al., 2021; Yao et al., 2022; Yu et al., 2025). Further ALFWorld studies evaluate common-source Gemma/Gemini skills and Gemma-31B/Gemini-tutored repairs (Appendices N–P). Table 1 gives the primary evaluation sets. Action-loss measurements apply to ALFWorld and WebShop; Math uses answer accuracy. Paired intervals resample tasks after within-task state aggregation; WebShop's repeated contrasts cluster by task. Run standard deviations describe execution variability. The 60-task ALFWorld NLL set lies within its 274-task endpoint set; both are held out from selection.

Table 2: Native writer–executor success rates. Columns are writers; rows are executors.

Benchmark	Executor	Qwen3-4B-Instruct writer	Qwen3-8B writer	Qwen3-14B writer
ALFWorld	Qwen3-4B-Instruct	35.8	32.2	30.2
	Qwen3-8B	44.4	45.6	39.3
	Qwen3-14B	67.0	58.8	64.1
WebShop	Qwen3-4B-Instruct	30.0	32.7	35.3
	Qwen3-8B	30.0	32.7	33.3
	Qwen3-14B	26.7	30.0	30.7
DAPO-Math	Qwen3-4B-Instruct	27.5	36.2	24.2
	Qwen3-8B	34.6	37.5	35.4
	Qwen3-14B	37.9	39.2	36.2

6.1 Observing the Writer–Executor Gap

Writer preferences differ across executors. Table 2 reports the Qwen matrices. ALFWorld's three-run means favor the 4B writer for the 4B and 14B executors, and the 8B writer for the 8B executor. Selecting the largest writer would miss the best-performing skill in each ALFWorld row.

Interaction tests support executor-relative evaluation. Table 3 tests whether writer effects vary across executors. The task-clustered tests give $p = 0.0038$ for ALFWorld success, $p = 5.43 \times 10^{-5}$ for WebShop action NLL and $p = 0.0464$ for Math accuracy. These interactions motivate diagnosing transfer at the writer–executor pairing level.

Table 3: Writer–executor interactions across benchmarks. Task-clustered omnibus tests use four degrees of freedom; ALFWorld uses the initial success matrix.

Benchmark	Outcome	Wald χ^2	p
ALFWorld	Task success	15.483	3.8×10^{-3}
WebShop	Action NLL	24.837	5.43×10^{-5}
DAPO-Math	Answer accuracy	9.670	4.64×10^{-2}

The Gap also appears with a shared source across models. All three Gemma/Gemini writers use the same 100 frozen ALFWorld trajectories. Table 4 crosses their skills with three executors on the same 274 tasks. Gemini-written skills achieve 46.4% with Gemma-12B, while Gemma-12B-written skills achieve 30.7%. For Gemma-31B, their point-estimate ordering reverses: 58.8% versus 63.9%. Because the writers share source trajectories, this comparison isolates variation in the written skills and their use from differences in source experience. Appendix N gives paired comparisons and action-format diagnostics.

6.2 How skill content and presentation affect the Gap

Figure 3 shows that executors respond differently to the same representation changes. The best granularity is G3 for 4B, G1 for 8B and G4 for 14B. Increasing detail therefore does not produce a shared optimum: the representation should be chosen with the target executor in view. Cumulative deletion lowers 14B success, while 4B/8B rebound at later deletions. Since these variants change both content and length, Table 5 adds factorial and word-matched controls. Adding information helps all three executors in the factorial run. In the word-matched run, padding does not help 4B, whereas added facts raise success from 24.5% to 33.6%; for 8B, information and padding tie at 40.1%, but reordering raises success to 44.2%. These controls distinguish the benefit of added decision content from the benefit of making content easier to use, motivating SkillDiag's separate content and presentation tests.

Table 4: Cross-model validation of the Writer–Executor Gap on ALFWorld. All writers use the same 100 source trajectories; each cell evaluates the same 274 tasks. Entries are success rates (%); bold marks the best skill per executor. G12, G31 and GF denote Gemma-4-12B, Gemma-4-31B and Gemini-3.8-Flash.

Executor	No Skill	G12 writer	G31 writer	GF writer
G12	26.3	30.7	31.0	**46.4**
G31	60.9	**63.9**	58.4	58.8
GF	97.8	96.7	97.8	**98.5**

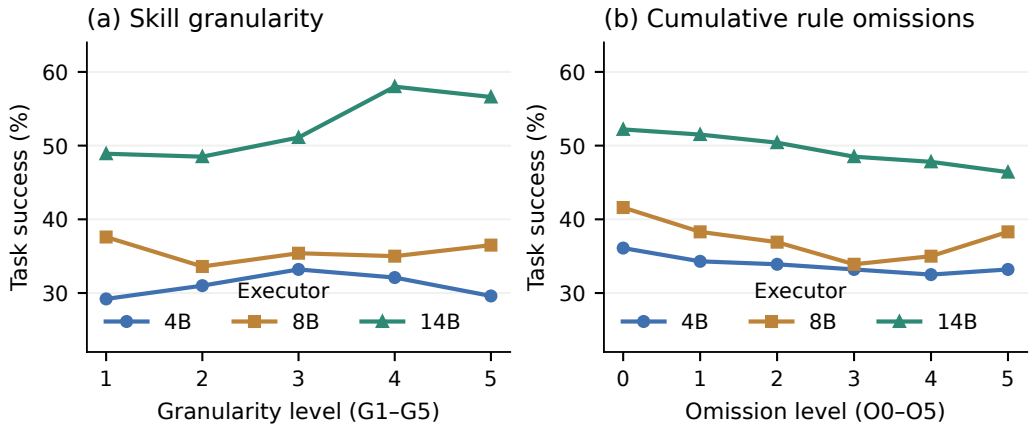


Figure 3: **Executor-dependent representation response.** G1–G5 add increasingly detailed instructions; O0–O5 cumulatively remove five decision rules. Points are single-run success rates on 274 tasks per condition. Lines connect evaluated variants.

6.3 SkillDiag identifies actionable transfer barriers

The controlled interventions in Table 6 connect SkillDiag's three scores to concrete ways of improving transfer. On the 16 tasks selected for high baseline SI, additional independent evidence reduces SI by 0.34 nats $[0.04, 0.64]$, supporting the use of the evidence-conditioned score to identify tasks that benefit from richer source experience. Restoring a structured procedure reduces WCL by 0.41 nats $[0.17, 0.66]$ and improves over equal-length repetition by 0.23 nats $[0.03, 0.45]$. This contrast supports examining decision content when the written skill loses utility relative to its source.

The presentation intervention directly motivates SkillTutor: reordering the same skill lines reduces corrected EDM by 0.11 nats $[0.07, 0.15]$ and executor NLL by 0.13 nats $[0.09, 0.18]$ across seven accepted cells. Because the instructions are retained, these gains show that changing how available information is presented can improve its use. Together, the interventions support diagnosing evidence, content and presentation separately, with the latter providing a repair opportunity even when the skill's content is fixed. Appendices E–F give the full protocols and auxiliary controls.

6.4 Improving execution with SkillTutor

Capsules help a smaller executor use different writers' skills. Table 7 examines the 4B executor across all three writers. Action-NLL gains are 0.74, 0.82 and 0.69 nats; both these gains and the reference-corrected gains have paired intervals above zero. The improvement therefore exceeds the reference's response to the same capsules, supporting the intended focus on the target executor's use of existing instructions. Appendix D reports all executor scales.

Failure-guided capsules improve task success. Across eight accepted ALFWorld cells, SkillTutor improves endpoint success over Generic by 2.78 pp $[1.00, 4.61]$ (Table 8). This comparison

Table 5: Static intervention ablations on ALFWorld. Values are success rates on 274 tasks per cell. Panels A and B are separate controlled runs with different Base values. IA adds decision facts; RA changes their representation.

Condition	Qwen3-4B-Instruct executor	Qwen3-8B executor	Qwen3-14B executor
A. Content and representation			
Base	24.1	38.3	51.5
RA only	25.2	42.0	48.9
IA only	35.6	42.7	58.8
IA + RA	36.9	42.3	57.3
B. Content, length, and ordering			
Base	24.5	38.0	54.4
Base + padding	24.5	40.1	53.3
IA	33.6	40.1	59.5
IA reordered	36.5	44.2	59.9
IA compact	35.8	44.2	57.3

Table 6: Controlled SKILLDIAG interventions on ALFWorld. Changes are intervention minus comparator (nats); negative values indicate reductions. The high-SI subset is baseline-selected. Protocol restoration and reordering are separate experiments; values are rounded to two decimals.

Population	Metric	Comparator	Change [95% CI]
A. Extra evidence: testing SI			
All 156 tasks	SI	Base	-0.02 $[-0.04, +0.00]$
16 high-SI tasks	SI	Base	-0.34 $[-0.64, -0.04]$
B. Structured protocol: testing WCL			
Protocol evaluation set	WCL	Base	-0.41 $[-0.66, -0.17]$
Same set	WCL	Repetition	-0.23 $[-0.45, -0.03]$
Same set	WCL	Wrong-family	$+0.02$ $[-0.19, +0.23]$
C. Strict reordering: testing EDM and related losses			
7 accepted cells	Corrected EDM	Base	-0.11 $[-0.15, -0.07]$
Same cells	WCL	Base	-0.02 $[-0.05, +0.01]$
Same cells	Executor NLL	Base	-0.13 $[-0.18, -0.09]$

supports constructing teaching content from observed failures rather than relying on a general reminder.

Repairs translate into WebShop task success. Across three executions, SKILLTUTOR improves success by 2.30 pp $[0.72, 3.95]$ over Base and 3.11 pp $[1.51, 4.79]$ over Generic. The Base and Generic comparisons support an end-to-end benefit from the constructed capsules. Appendix D reports the pairing-level results and target/off-target comparisons.

The repair procedure transfers across tutor families. Table 9 evaluates Gemma-31B and Gemini-3.8-Flash tutoring. Gemma yields reference-corrected gains of 0.15 and 0.16 nats in the Qwen-only and mixed matrices, with Top-1 improvements of 2.85 and 4.34 pp. Across the three

Table 7: SkillTutor action gains for the ALFWorld 4B executor, across all three writers (60 tasks, 168 states). Gains are in nats with paired 95% intervals. Corrected gain subtracts the reference's improvement.

Writer	NLL gain	Corrected gain
4B	0.74 $[0.47, 1.01]$	0.63 $[0.37, 0.89]$
8B	0.82 $[0.57, 1.07]$	0.59 $[0.36, 0.82]$
14B	0.69 $[0.35, 1.04]$	0.71 $[0.38, 1.03]$

Table 8: Endpoint gains of deployed Personalized (P) capsules in percentage points, with paired 95% intervals. ALFWorld pools eight accepted cells; WebShop uses three executions; Math pools all nine cells with exact Base fallback.

Benchmark	Gate	P − Base	P − Generic	P − Shuffled
ALFWorld	8/9	$+1.50\ [-0.20, +3.20]$	$+2.78\ [+1.00, +4.61]$	$+0.23\ [-1.37, +1.87]$
WebShop	9/9	$+2.30\ [+0.72, +3.95]$	$+3.11\ [+1.51, +4.79]$	$+0.99\ [-0.44, +2.44]$
Math	2/9	$+0.32\ [-0.46, +1.11]$	$+0.79\ [-1.02, +2.64]$	$+0.09\ [-1.57, +1.76]$

Table 9: Cross-family tutoring on ALFWorld, pooling all nine cells including fallback. Gains are in nats; Top-1 changes are in pp. ExecGain and Top-1 intervals exclude zero throughout; CapsuleGain does so only for Gemma. Gemini uses two-stage selection and new final tasks. Full results: Appendices O–P.

Tutor	Matrix	ExecGain	CapsuleGain	ΔTop-1
Gemma-31B	Qwen-only	+0.29	+0.15	+2.85
Gemma-31B	Mixed	+0.31	+0.16	+4.34
Gemini-3.8-Flash	Qwen-scale	+0.07	+0.03	+1.50
Gemini-3.8-Flash	Large-mixed	+0.09	+0.05	+1.81
Gemini-3.8-Flash	Confirm	+0.04	+0.01	+1.08

Gemini-tutored matrices, action-NLL gains are 0.07, 0.09 and 0.04 nats, with Top-1 improvements of 1.50, 1.81 and 1.08 pp. All executor-NLL and Top-1 intervals exclude zero. These results extend the action-prediction benefit to different tutors and mixed writer–executor settings. Appendices O–P give full protocols and intervals.

6.5 Repair gains persist across repeated execution

Table 10 evaluates the consistency of WebShop repairs under repeated execution. Success gains over Base are 2.44, 2.37 and 2.07 pp, with a run SD of 0.20 pp. Gains over Generic are also positive in each run: 2.89, 3.04 and 3.41 pp. The benefit appears in every measured execution; both task-clustered aggregate intervals in Table 8 remain above zero. Together with the cross-family action results, this supports evaluating repairs through both local decisions and repeated task completion. Further budget and extraction-seed studies appear in Appendix I.

Table 10: WebShop Personalized gains across three executions (pp), with mean ± run SD in the final row. Task-clustered pooled intervals appear in Table 8.

Execution	vs Base	vs Generic	vs Shuffled
Run 1	$+2.44$	$+2.89$	$+0.89$
Run 2	$+2.37$	$+3.04$	$+0.74$
Run 3	$+2.07$	$+3.41$	$+1.33$
Mean ± SD	$+2.29 \pm 0.20$	$+3.11 \pm 0.27$	$+0.99 \pm 0.31$

7 Conclusion

In this paper, we identify the Writer–Executor Gap, introduce SkillDiag to investigate its sources, and propose SkillTutor to improve skill use through executor feedback. Our experiments yield three findings. First, skill effectiveness is executor-dependent: writer rankings vary across executors, making compatibility important for skill reuse. Second, controlled interventions show how evidence, content and presentation influence skill use, with cross-family studies further highlighting instruction-format compatibility. Third, SkillTutor improves WebShop task success and action prediction across tutor families, supporting failure-guided adaptation of skill presentation.

AI USE STATEMENT

AI tools assisted manuscript writing, translation and the presentation of theoretical arguments. The authors take full responsibility for the content and correctness of this work.

REPRODUCIBILITY STATEMENT

The appendices detail the experimental protocols, data splits, controls and statistical units. Appendix B provides the theoretical assumptions and proofs. Appendices C–J present complete matrices, controlled interventions and repeated-run analyses; Appendix M lists the lengths and hashes of 21 frozen skills. Appendices N–P specify the cross-model evaluations, candidate selection and held-out tests, including the separate stability-validation and final sets used for Gemini tutoring.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, 2022.

David Blackwell. Equivalent comparisons of experiments. *The Annals of Mathematical Statistics*, 24(2):265–272, 1953. doi: 10.1214/aoms/1177729032.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2 edition, 2006.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. HiAgent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *Annual Meeting of the Association for Computational Linguistics*, 2025.

Carlos E. Jimenez et al. SWE-bench: Can language models resolve real-world GitHub issues? *International Conference on Learning Representations*, 2024.

Xiangyi Li, Yimin Liu, Wenbo Chen, et al. SkillsBench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026. URL `https://arxiv.org/abs/2602.12670v4`.

Xiao Liu, Hao Yu, Hanchen Zhang, et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024.

Grégoire Mialon et al. GAIA: A benchmark for general AI assistants. *International Conference on Learning Representations*, 2024.

Tianyue Ou, Wanyao Guo, Apurva Gandhi, Graham Neubig, and Xiang Yue. AgentDiagnose: An open toolkit for diagnosing LLM agent trajectories. In *Proceedings of EMNLP: System Demonstrations*, 2025.

Yujia Qin, Shihao Liang, Yining Ye, et al. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations*, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, et al. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. *International Conference on Learning Representations*, 2021.

Yifan Song, Weimin Xiong, Xiutian Zhao, et al. AgentBank: Towards generalized LLM agents via fine-tuning on 50000+ interaction trajectories. In *Findings of the Association for Computational Linguistics: EMNLP*, 2024.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. URL https://arxiv.org/abs/physics/0004057. Manuscript dated September 1999.

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.

Hanlin Wang, Jian Wang, Chak Tou Leong, and Wenjie Li. STeCa: Step-level trajectory calibration for LLM agent learning. In *Findings of the Association for Computational Linguistics: ACL*, 2025a.

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. Reinforcement learning for self-improving agent with skill library. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1529–1550. Association for Computational Linguistics, July 2026. doi: 10.18653/v1/2026.acl-long.69. URL https://aclanthology.org/2026.acl-long.69/.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *International Conference on Machine Learning*, 2025b.

Aaron D. Wyner and Jacob Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, 1976. doi: 10.1109/TIT.1976.1055508.

Yiheng Xu, Dunjie Lu, Zhennan Shen, Junli Wang, Zekun Wang, Yuchen Mao, Caiming Xiong, and Tao Yu. AgentTrek: Agent trajectory synthesis via guiding replay with web tutorials. In *International Conference on Learning Representations*, 2025.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, 2022.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Shanshan Zhong, Yi Lu, Jingjie Ning, et al. SkillLearnBench: Benchmarking continual learning methods for agent skill generation on real-world tasks. *arXiv preprint arXiv:2604.20087*, 2026.

A EXPERIMENTAL ORGANIZATION

The appendices provide the theoretical derivations, full writer–executor matrices, SKILLDIAG interventions and SKILLTUTOR outcomes. Appendices C–J cover the three primary benchmarks and auxiliary controls; Appendix K examines six decision-state cases. The AppWorld and AIME studies in Appendix L use separate evaluation sets and experience sources. Appendix M describes skill lengths and measured overhead. Cross-family experiments appear in Appendices N, O and P.

B THEORY, PROOFS AND OPERATIONAL INTERPRETATION

This appendix supplies the assumptions and proofs behind Sections 3 and 4, derives the intervention predictions, and separates the theoretical action loss from SKILLDIAG's practical set-valued score. The entropy decomposition, information-bottleneck objective and comparison-of-experiments argument use standard information-theoretic tools to organize the Writer–Executor Gap and its controlled tests.

B.1 PROBABILITY MODEL AND SCOPE

All random variables are discrete. Entropies and the expected losses used in comparisons are finite, and logarithms are natural. In particular, $q_e(a \mid s, y) > 0$ on the support of $p(a \mid s, y)$ and the expected log-loss is finite.

The source X contains the experience made available to the writer. The target A^* is a random action drawn from a fixed evaluation law, which may be degenerate at a canonical action. For a writer comparison, the joint law $p(x, a^*, y_e)$ is fixed. The encoder may be randomized, but the Strict Skill Protocol requires its randomness to be independent of the target and side information conditional on X. A separate retrieved document correlated with the target must be included in X if it is supplied to the writer.

The protocol specifies information flow; source audits constrain the assertions in a generated skill. The intervention controls record both the source restriction and the construction/evaluation split.

The executor signal Z_e abstracts task information accessible through its prior and interface. The comparisons below fix the observation interface, tool set, parsing rules and rollout budget.

B.2 PROOF OF THE DECISION-LOSS DECOMPOSITION

For a fixed writer and executor, adding and subtracting the true conditional log probability gives

$$\begin{aligned}\mathcal{L}_e(p_\phi) &= \mathbb{E}[-\log p(A^* \mid S, Y_e)] + \mathbb{E}\log \frac{p(A^* \mid S, Y_e)}{q_e(A^* \mid S, Y_e)} \\ &= H(A^* \mid S, Y_e) + D_e(p_\phi). \end{aligned} \tag{7}$$

The Markov condition implies $A^* \perp S \mid X, Y_e$. Consequently,

$$\begin{aligned} I(A^*; X \mid S, Y_e) &= H(A^* \mid S, Y_e) - H(A^* \mid X, S, Y_e) \\ &= H(A^* \mid S, Y_e) - H(A^* \mid X, Y_e). \end{aligned} \tag{8}$$

Substitution into Eq. 7 proves Theorem 1. Conditional entropy, conditional mutual information and expected KL divergence are nonnegative under the stated assumptions (Cover & Thomas, 2006).

The source term depends on Y_e, so it need not be identical across executors. Compression loss is zero precisely when $A^* \perp X \mid S, Y_e$, meaning the skill is sufficient for the target relative to the complete trace and that executor's side information. Mismatch is zero precisely when the fixed decoder equals the true posterior almost surely. Sufficient skill content therefore does not imply that the actual executor uses it correctly.

For any admissible encoder and decoder, $\mathcal{L}_e \geq H(A^* \mid X, Y_e)$. A source-only skill cannot remove uncertainty that remains even with the full source and the same side information. A finite model's Full-Trace loss additionally includes decoder mismatch.

B.3 CONDITIONAL INFORMATION BOTTLENECK

For a fixed executor, compression loss equals $I(A^*; X \mid Y_e) - I(A^*; S \mid Y_e)$. Skill curation therefore admits the conditional information-bottleneck formulation (Tishby et al., 2000)

$$\max_{p(s|x)} I(A^*; S \mid Y_e) \quad \text{subject to} \quad I(X; S \mid Y_e) \leq B. \tag{9}$$

Here B is an information budget. Token count is a practical surrogate, and optimizing retained information alone leaves D_e uncontrolled.

Expanding $I(A^*; X, S \mid Y_e)$ in two orders and using the protocol gives

$$\begin{aligned} I(A^*; X \mid Y_e) &= I(A^*; X, S \mid Y_e) \\ &= I(A^*; S \mid Y_e) + I(A^*; X \mid S, Y_e). \end{aligned} \tag{10}$$

The first term on the left is fixed across writers sharing the source and executor law. Minimizing compression loss subject to a rate budget is therefore equivalent to Eq. 9, the conditional analogue of the information bottleneck (Tishby et al., 2000). For a multiplier $\beta \geq 0$, a corresponding Lagrangian maximizes $I(A^*; S \mid Y_e) - \beta I(X; S \mid Y_e)$.

An implementation may instead compare $\widehat{\mathcal{L}}_e(S) + \lambda \operatorname{Tokens}(S)$. That practical criterion includes fixed-decoder behavior and uses tokenizer length rather than conditional mutual information. The granularity and omission curves explore this practical tradeoff; equal-length controls compare content and presentation at a fixed word budget.

B.4 PROOF OF THE COMPATIBILITY FRONTIER

Following decoder-side-information theory (Wyner & Ziv, 1976), define a single-letter decision frontier using an auxiliary code U and an unrestricted matched decoder r_e:

$$R_e(D) = \inf_{p(u|x), r_e} I(X; U \mid Y_e) \quad \text{s.t.} \quad \mathbb{E}[-\log r_e(A^* \mid U, Y_e)] \leq D, \tag{11}$$

where $U - X - (A^*, Y_e)$. The frontier measures the conditional information required to achieve loss D with a matched decoder.

Proposition 2 (Compatibility frontier)**.** *If Y_s Blackwell-dominates Y_w, with joint law $p(x, a^*, y_s)K(y_w \mid y_s)$, then $R_s(D) \leq R_w(D)$ for every weak-side feasible D.*

Richer side information can reduce the code needed for a given ideal loss under this assumption.

For a common encoder, use the joint law

$$p(x, a^*, y_s, y_w, u) = p(x, a^*, y_s)K(y_w \mid y_s)p(u \mid x). \tag{12}$$

Take any code $p(u \mid x)$ and decoder r_w feasible for the weak signal at distortion D. Construct a strong-side predictive distribution by mixing the weak decoder over the garbling kernel:

$$r_s(a \mid u, y_s) = \sum_{y_w} K(y_w \mid y_s) r_w(a \mid u, y_w). \tag{13}$$

Convexity of $-\log$ and Eq. 12 imply

$$\begin{aligned} \mathbb{E}[-\log r_s(A^* \mid U, Y_s)] &\leq \mathbb{E}\left[\sum_{y_w} K(y_w \mid Y_s)[-\log r_w(A^* \mid U, y_w)]\right] \\ &= \mathbb{E}[-\log r_w(A^* \mid U, Y_w)] \leq D. \end{aligned} \tag{14}$$

Thus the same encoder is feasible for the strong side. Further, $U - Y_s - Y_w$ and $U - X - (Y_s, Y_w)$ yield

$$\begin{aligned} I(X; U \mid Y_s) &= H(U \mid Y_s) - H(U \mid X) \\ &\leq H(U \mid Y_w) - H(U \mid X) = I(X; U \mid Y_w). \end{aligned} \tag{15}$$

Taking an infimum over weak-feasible encoders proves Proposition 2. If the infimum is not attained, apply the argument to an arbitrarily close feasible sequence.

The ordering uses Blackwell's comparison of information channels (Blackwell, 1953).

B.5 Writer comparison and action success

Applying Theorem 1 and Eq. 10 separately to each writer gives

$$\mathcal{L}_e(p_i) = H(A^* \mid X, Y_e) + I(A^*; X \mid Y_e) - I(A^*; S_i \mid Y_e) + D_e(p_i). \tag{16}$$

The first two terms are common. Subtracting proves Eq. 2. If $\Delta I_e \geq 0$, higher loss for writer 1 occurs exactly when $\Delta D_e > \Delta I_e$. Between two executors, strict writer-order reversal in log-loss occurs when the nonzero quantities $\Delta D_e - \Delta I_e$ have opposite signs.

A finite example. Let $A^* = X$ be a fair bit and let side information be constant. Writer 0 emits no information; an executor assigning $1/2$ to either action has loss $\log 2$. Writer 1 transmits the bit exactly, but the same executor assigns probability 0.4 to the correct action when reading that code. Its loss is $-\log 0.4 > \log 2$, despite retained information increasing by $\log 2$. Here source uncertainty is zero, writer 1's compression loss is zero, and its mismatch exceeds the information gain. A second executor assigning 0.9 to the transmitted bit instead prefers writer 1.

A unique-target argmax decision satisfies

$$\Pr(\widehat{A}_e \neq A^*) \leq \frac{\mathcal{L}_e(p_\phi)}{\log 2}, \tag{17}$$

because an incorrect argmax requires $q_e(A^* \mid S, Y_e) \leq 1/2$. This is only an upper bound, and reducing the bound does not order actual accuracy. For example, across two equally weighted binary decisions, correct-action probabilities $(0.99, 0.49)$ give lower mean log-loss but lower accuracy than $(0.60, 0.60)$. The valid-set score is derived in Appendix B.7. Free-running episodes visit policy-dependent states, whereas our action scores use fixed oracle-prefix states.

The Qwen native comparisons use skills extracted from each writer's own trajectories. The Gemma/Gemini experiment in Appendix N fixes the source trajectories to compare writer artifacts from common experience.

B.6 Intervention predictions and their assumptions

The decomposition specifies what each intervention would change under ideal controls. These are population statements under explicitly related channels; the tests in the main paper measure finite-predictor counterparts.

Expanding source evidence. Suppose additional evidence V becomes available and the source changes from X to (X, V) under a common joint law. Then

$$H(A^* \mid X, Y_e) - H(A^* \mid X, V, Y_e) = I(A^*; V \mid X, Y_e) \geq 0. \tag{18}$$

Only action-relevant additional evidence strictly lowers the ideal source term. A finite reference score may increase even when conditioning reduces ideal entropy.

Restoring omitted content. Let (S, C) be a refined code satisfying $(S, C) - X - (A^*, Y_e)$, with the original S recoverable by projection. The compression terms obey

$$I(A^*; X \mid S, Y_e) - I(A^*; X \mid S, C, Y_e) = I(A^*; C \mid S, Y_e) \geq 0. \tag{19}$$

The total loss change is $-I(A^*; C \mid S, Y_e) + D_e(S, C) - D_e(S)$. Restoring information therefore need not improve a fixed decoder if its mismatch increases. When C is only a deterministic restatement of S, the information gain is zero.

Changing representation without changing information. Let $\widetilde{S} = h(S)$ be a deterministic bijection on the skill support, using a fixed publicly specified transformation. Then the conditional sigma fields generated by (S, Y_e) and $(\widetilde{S}, Y_e)$ agree, and

$$I(A^*; \widetilde{S} \mid Y_e) = I(A^*; S \mid Y_e), \qquad \mathcal{L}_e(\widetilde{S}) - \mathcal{L}_e(S) = D_e(\widetilde{S}) - D_e(S). \tag{20}$$

This gives a pure mismatch intervention under the bijection assumption. The empirical reordering test preserves the line multiset and byte count, and uses reference-side controls to assess information-use changes: order and adaptive selection can also carry information.

Calibration and the information boundary. If a capsule is generated from S and independent randomness alone, it supplies no additional target information conditional on (S, Y_e), although it can change a fixed executor's predictions. Actual SkillTutor construction also uses calibration failures. Source audits constrain its assertions, while disjoint construction, selection and final pools constrain information leakage. Generic and shuffled controls test alternative explanations.

B.7 SET-VALUED LOSS AND FINITE-REFERENCE CONTRASTS

The theoretical target is a random action with a specified law. The interactive score instead measures total predicted mass on a nonempty verified set $\mathcal{A}^+(h)$ within a scored candidate list. At one fixed state and context, let q be the candidate-normalized predictor, let $Q^+ = \sum_{a\in\mathcal{A}^+} q(a) > 0$, and fix a target distribution π supported on $\mathcal{A}^+$. Since $q(a) = Q^+ q(a \mid \mathcal{A}^+)$ on that set,

$$-\sum_a \pi(a)\log q(a) = -\log Q^+ + H(\pi) + D_{\mathrm{KL}}\big(\pi \| q(\cdot \mid \mathcal{A}^+)\big). \tag{21}$$

The set-valued score omits the within-valid-set cross-entropy. It equals canonical-action loss for a singleton valid set, but otherwise the two scores can rank predictors differently.

The empirical identity has a different basis. Write $a = L(R, E)$, $b = L(R, S_w)$, $c = L(E_e, E)$ and $d = L(E_e, S_w)$. Then

$$a + (b - a) + (c - a) + (d - c - b + a) = d. \tag{22}$$

This is Eq. 6, independent of Bayes optimality or the Strict Skill Protocol. Corrected EDM is a difference of skill-versus-evidence contrasts. It removes the executor's evidence-conditioned gap relative to the reference, but it does not generally equal an absolute KL divergence. For example, $(a, b, c, d) = (1, 2, 3, 3.5)$ gives corrected EDM -0.5 despite every observed loss being positive.

Even for canonical-action loss and a reference matched to the same side information, $L_R(S) - L_R(X)$ equals compression loss plus $D_R(S) - D_R(X)$. It identifies compression only if the reference mismatch difference vanishes, in particular when the reference is Bayes optimal in both contexts. Under that ideal reference, the corrected executor contrast is still $D_e(S) - D_e(X)$; it equals absolute skill-conditioned mismatch only when $D_e(X) = 0$ as well.

Accordingly, empirical SI/WCL/EDM are operational diagnostics. Evidence addition, repetition, wrong-family content and ordering controls probe their interpretation.

C FULL NATIVE MATRICES AND RANKING EVIDENCE

The primary matrices and task-level contrasts describe distinct notions of compatibility. ALFWorld exhibits task-pair reversals, WebShop exhibits local-versus-endpoint ranking differences, and Math exhibits a skill-and-budget-dependent writer contrast. Omnibus and multiplicity-adjusted pairwise tests assess these differences at different levels. The WebShop and Math omnibus statistics are 24.837 ($p = 5.43 \times 10^{-5}$) and 9.670 ($p = 0.0464$), respectively. Representative Holm-adjusted values are 0.00154 for a WebShop writer contrast and 0.0018 for Math's 4B-executor 14B-minus-8B contrast.

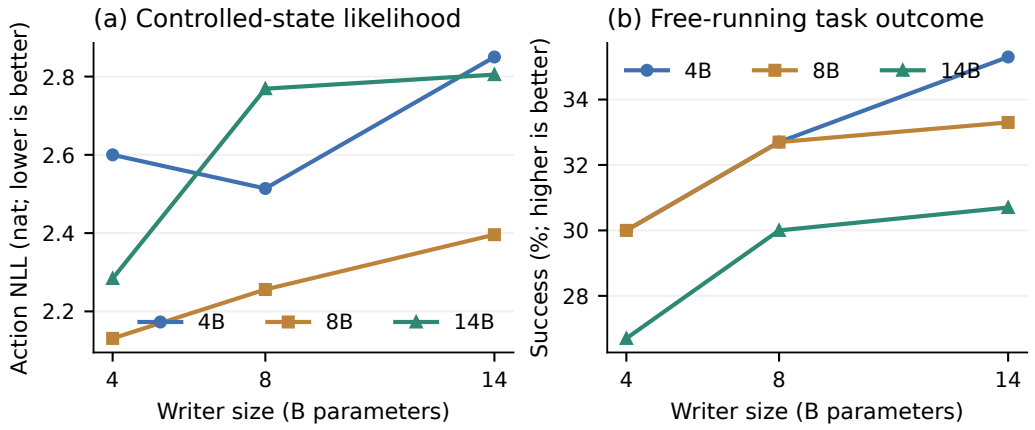


Figure 4: WebShop writer rankings depend on whether the outcome is controlled-state action NLL or free-running success. Lines connect the three measured writers.

Table 11: Initial native ALFWorld matrix (one run). Rows are writers; columns are executors. This Qwen matrix is distinct from the three-run mean in the main table.

Writer	4B executor	8B executor	14B executor
4B	37.2%	43.1%	64.2%
8B	31.0%	46.7%	59.5%
14B	31.8%	39.8%	65.3%

The omnibus interaction tests appear in main Table 3.

Table 12: Task-level ranking reversals conditional on both writer comparisons being decisive. Denominators count comparable task–writer-pair instances, not all tasks.

Executor pair	Reversals / comparable	Rate
4B / 8B	20/53	37.7%
4B / 14B	8/33	24.2%
8B / 14B	16/43	37.2%

Table 13: WebShop action NLL. Rows are writers and columns are executors; lower is better.

Writer	4B executor	8B executor	14B executor
4B	2.600	2.131	2.284
8B	2.514	2.256	2.769
14B	2.850	2.396	2.805

Table 14: WebShop paired action-loss difference, 14B writer minus 4B writer. Positive means larger-writer skills have worse local likelihood.

Executor	NLL difference	95% CI
4B	+0.251	[+0.021,+0.477]
8B	+0.265	[+0.105,+0.433]
14B	+0.521	[+0.366,+0.687]

D COMPLETE SKILLTUTOR OUTCOMES

Endpoint Base measurements come from concurrent SkillTutor runs rather than the initial native-matrix evaluation. On ALFWorld the 4B-writer/8B-executor candidate was rejected; a separate identical-prompt execution changed success from 46.4% to 42.7%. That 3.6-point serving-repeat change is excluded from the deployed treatment effect. The accepted-cell NLL and endpoint measurements use overlapping task sets.

We distinguish three gains: executor NLL gain, reference-corrected capsule gain, and matched-versus-off-target specificity. In eight accepted ALFWorld cells the respective pooled executor and corrected gains are 0.4686 and 0.3803 nats; the matched/off-target aggregate has an interval crossing zero. The WebShop category audit evaluates 135 candidates, of which 108 pass the information-boundary auditor.

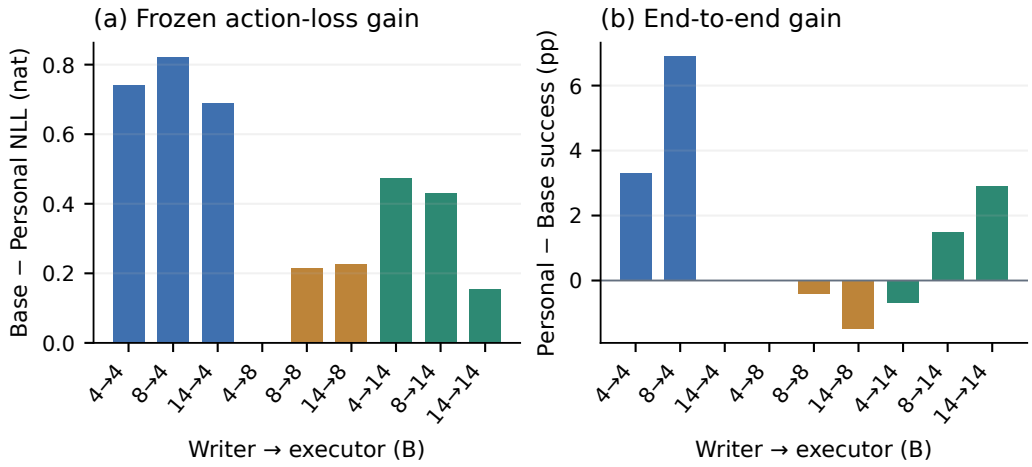


Figure 5: ALFWorld paired-cell NLL and endpoint gains. The rejected cell is zero by exact fallback. NLL and endpoint changes vary across cells.

Table 15: Full ALFWorld SkillTutor endpoint results. The rejected cell is deployed as exact Base; the identical-prompt repeat measures serving variability.

Writer	Exec.	Gate	Base	P	Delta pp [95% CI]	W/L	p
4B	4B	accept	33.6%	36.9%	+3.3 [-1.8,+8.4] pp	30 / 21	0.2624
8B	4B	accept	23.7%	30.7%	+6.9 [+2.9,+11.0] pp	26 / 7	0.0013
14B	4B	accept	37.6%	37.6%	+0.0 [-5.1,+5.1] pp	25 / 25	1.0000
4B	8B	fallback	46.4%	46.4%	+0.0 [+0.0,+0.0] pp	–	–
8B	8B	accept	37.6%	37.2%	-0.4 [-5.1,+4.4] pp	21 / 22	1.0000
14B	8B	accept	46.7%	45.3%	-1.5 [-6.2,+3.3] pp	20 / 24	0.6516
4B	14B	accept	64.2%	63.5%	-0.7 [-5.1,+3.6] pp	18 / 20	0.8714
8B	14B	accept	55.1%	56.6%	+1.5 [-3.6,+6.6] pp	28 / 24	0.6778
14B	14B	accept	64.6%	67.5%	+2.9 [-1.5,+7.3] pp	24 / 16	0.2682

Table 16: All WebShop final conditions (150 tasks per cell). P denotes Personalized.

Writer	Exec.	Gate	Base	P	Generic	Shuffled	Delta pp [95% CI]
4B	4B	accept	30.0%	32.0%	28.0%	28.7%	+2.0 [-2.0,+6.0] pp
8B	4B	accept	32.7%	31.3%	28.0%	32.0%	-1.3 [-5.3,+2.7] pp
14B	4B	accept	35.3%	34.7%	34.7%	35.3%	-0.7 [-4.7,+3.3] pp
4B	8B	accept	30.0%	36.7%	34.0%	35.3%	+6.7 [+2.7,+11.3] pp
8B	8B	accept	32.7%	36.7%	33.3%	35.3%	+4.0 [-0.7,+8.7] pp
14B	8B	accept	33.3%	38.0%	34.0%	38.0%	+4.7 [+0.0,+9.3] pp
4B	14B	accept	26.7%	30.7%	28.7%	28.7%	+4.0 [+1.3,+7.3] pp
8B	14B	accept	30.0%	29.3%	29.3%	29.3%	-0.7 [-4.7,+3.3] pp
14B	14B	accept	30.7%	34.0%	27.3%	32.7%	+3.3 [-1.3,+8.0] pp

Table 17: All DAPO-Math final conditions (240 distinct problems). Seven rejected Personalized cells are set to exact Base.

Writer	Exec.	Gate	Base	P	Generic	Shuffled	Delta pp [95% CI]
4B	4B	fallback	27.5%	27.5%	27.5%	30.4%	+0.0 [+0.0,+0.0] pp
8B	4B	fallback	36.2%	36.2%	32.5%	34.2%	+0.0 [+0.0,+0.0] pp
14B	4B	accept	24.2%	27.9%	28.7%	27.1%	+3.8 [-1.7,+9.2] pp
4B	8B	fallback	34.6%	34.6%	34.2%	37.1%	+0.0 [+0.0,+0.0] pp
8B	8B	fallback	37.5%	37.5%	39.2%	36.7%	+0.0 [+0.0,+0.0] pp
14B	8B	fallback	35.4%	35.4%	33.3%	33.3%	+0.0 [+0.0,+0.0] pp
4B	14B	accept	37.9%	37.1%	36.2%	37.5%	-0.8 [-6.2,+4.6] pp
8B	14B	fallback	39.2%	39.2%	41.7%	37.9%	+0.0 [+0.0,+0.0] pp
14B	14B	fallback	36.2%	36.2%	31.2%	36.7%	+0.0 [+0.0,+0.0] pp

Table 18: Frozen ALFWorld NLL scores and executor gain, on 60 tasks / 168 states.

Exec.	Writer	Accept	Base NLL	P NLL	Gain [95% CI]
4B	4B	yes	0.9847	0.2440	0.7407 [0.4696, 1.0051]
4B	8B	yes	1.8726	1.0506	0.8220 [0.5703, 1.0707]
4B	14B	yes	1.1312	0.4430	0.6882 [0.3461, 1.0379]
8B	4B	no to Base	0.2367	0.2367	0.0000
8B	8B	yes	0.4056	0.1917	0.2140 [0.0410, 0.4148]
8B	14B	yes	0.4438	0.2185	0.2253 [0.0452, 0.4499]
14B	4B	yes	0.7176	0.2445	0.4731 [0.2651, 0.6688]
14B	8B	yes	1.0545	0.6238	0.4307 [0.2718, 0.6103]
14B	14B	yes	0.4408	0.2858	0.1550 [0.0607, 0.2466]

Table 19: Reference-corrected capsule gains and top-1 action changes for the same NLL evaluation.

Exec.	Writer	Capsule gain [95% CI]	Δ top-1
4B	4B	0.6261 [0.3662, 0.8887]	+0.0778
4B	8B	0.5932 [0.3628, 0.8248]	+0.0389
4B	14B	0.7098 [0.3834, 1.0346]	+0.0500
8B	4B	0.0000	0.0000
8B	8B	0.0640 [-0.0889, 0.2237]	+0.0028
8B	14B	0.2779 [0.0941, 0.4947]	+0.0389
14B	4B	0.3509 [0.1313, 0.5602]	+0.0708
14B	8B	0.3023 [0.1762, 0.4451]	+0.0139
14B	14B	0.1186 [0.0131, 0.2108]	+0.0500

Table 20: ALFWorld target-matched versus off-target capsule comparisons. Positive values favor the matched capsule.

Exec.	Writer	Accepted	Matched minus off-target	Corrected specificity
4B	4B	yes	0.1391 [0.0451, 0.2501]	0.1357 [0.0367, 0.2582]
4B	8B	yes	-0.0731 [-0.2635, 0.1028]	-0.1635 [-0.3490, 0.0051]
4B	14B	yes	-0.1252 [-0.2433, -0.0158]	-0.1084 [-0.2157, -0.0134]
8B	4B	no	-0.0161 [-0.0409, 0.0040]	0.0039 [-0.0250, 0.0290]
8B	8B	yes	0.0656 [-0.0375, 0.2357]	0.0987 [-0.0078, 0.2607]
8B	14B	yes	0.0118 [-0.0024, 0.0284]	0.0686 [0.0216, 0.1193]
14B	4B	yes	0.0282 [-0.0089, 0.0636]	0.0114 [-0.0301, 0.0491]
14B	8B	yes	0.0644 [-0.0120, 0.1438]	0.1218 [0.0263, 0.2236]
14B	14B	yes	0.0094 [-0.0458, 0.0606]	-0.0642 [-0.1392, 0.0012]

Table 21: ALFWorld action parse failures.

Exec. / writer	Base failure rate	P failure rate
4B / 4B	14.9%	13.2%
4B / 8B	12.2%	12.5%
4B / 14B	11.7%	12.2%
8B / 4B	16.6%	17.6%
8B / 8B	17.7%	20.6%
8B / 14B	14.7%	16.3%
14B / 4B	13.4%	15.6%
14B / 8B	15.7%	14.8%
14B / 14B	15.0%	14.2%

E DIAGNOSTIC DEVELOPMENT AND FAILED ROUTING

The pilot precedes the final masked-evidence reference condition. An initial reference failure on the light-examination family motivated the revised gate. The final gate reaches 97.5% top-1 over 60 tasks, with remaining uncertainty for light and heat.

A preliminary routing test used masked evidence, task-specific information addition (IA), and a representation capsule (RA). Only 14.5% of state–cell instances were assigned a non-LOW dominant component. Negative matched advantages mean the proposed correspondence between diagnoses and treatments was not validated by that exploratory test.

Table 22: Pilot WCL and corrected EDM before adoption of the final masked-evidence reference condition.

Writer	WCL [CI]	EDM 4B [CI]	EDM 8B [CI]	EDM 14B [CI]
4B	0.0503 [-0.0606, 0.1628]	0.5218 [0.2532, 0.7881]	-0.4484 [-0.7894, -0.1136]	0.3781 [0.0863, 0.6874]
8B	0.2265 [0.1181, 0.3285]	1.2336 [0.9618, 1.5069]	-0.4557 [-0.7006, -0.2048]	0.5389 [0.2222, 0.8783]
14B	-0.0140 [-0.1082, 0.0809]	0.7326 [0.3490, 1.1078]	-0.1771 [-0.6458, 0.3109]	0.1656 [-0.0746, 0.4407]

Table 23: Pilot pairwise crossover contrasts with task-level intervals.

Executor pair	Estimate	95% CI
4B vs 8B	0.7190	[0.3908, 1.0910]
4B vs 14B	0.4233	[0.1149, 0.7290]
8B vs 14B	0.6519	[0.3820, 0.9473]

Table 24: Masked-evidence reference gate by ALFWorld family. Each family contains ten tasks.

Family	Tasks	Top-1	NLL
look_at_obj_in_light	10	0.900	0.3620
pick_and_place_simple	10	1.000	0.0002
pick_clean_then_place	10	1.000	0.0014
pick_cool_then_place	10	1.000	0.0008
pick_heat_then_place	10	0.950	0.6750
pick_two_obj_and_place	10	1.000	0.0009
Overall	60	0.975	0.1734

Table 25: Baseline dominant-component classification over 1,512 state–cell instances. LOW denotes no dominant component under the classification criterion.

Class	Instances	Proportion
SI	23	1.5%
WCL	118	7.8%
corrected EDM	78	5.2%
LOW	1293	85.5%

Table 26: Exploratory diagnosis-by-treatment routing: gains and matched-treatment advantages.

Diagnosis	Evidence gain	IA gain	RA gain	Matched advantage
SI	-2.0090	-1.3999	+1.0522	-1.8351
WCL	+0.0995	+0.3282	+0.7391	-0.0911
EDM	+6.2550	+5.3300	+4.4260	-1.3665
LOW	-0.0509	-0.0046	+0.0007	0.0000

Table 27: Exploratory routing component changes. SI, WCL and EDM here are diagnosis-group averages; changes are treatment minus Base.

Group	SI	WCL	EDM	Evidence ΔL	IA ΔL	RA ΔWCL	RA ΔEDM	RA ΔL
SI	6.7918	-0.4491	-1.5599	+2.0090	+1.3999	-0.5857	-0.4664	-1.0522
WCL	0.0280	1.1734	-1.0739	-0.0995	-0.3282	-0.3246	-0.4145	-0.7391
EDM	0.2818	0.4918	5.7632	-6.2550	-5.3300	-0.5188	-3.9073	-4.4260
LOW	0.0008	0.0040	-0.0549	+0.0509	+0.0046	+0.0015	-0.0022	-0.0007

F STRICT EVIDENCE, CONTENT AND ORDERING CONTROLS

The strict follow-ups evaluate evidence, content and ordering interventions against matched controls. The first evidence study's high-SI estimate uses four independent tasks. The expanded study uses 156 tasks, with 16 and 9 tasks represented in higher-SI subsets.

WCL has several different interventions. Masked-plan IA fails against a token-matched donor placebo. Missing-Atom uses transitions extracted from 90 independent training tasks and fails the repetition gate despite a diagnosed-subset WCL reduction of 0.2970 nats. The later structured protocol improves WCL by 0.4066 nats and beats repetition, but not wrong-family structure. Main Table 6 gives the later protocol contrasts.

Strict RA permutes the exact line multiset and bytes. Seven of nine cells pass selection among trigger-first, verification-first and reverse orders. In baseline EDM-diagnosed instances the corrected loss decreases by 1.2199 nats, with the reference-side interval crossing zero. Only the three 4B-executor cells enter its separately gated endpoint test. Their pooled gain is $+1.9$ pp $[-0.9, +4.6]$. One local bootstrap interval excludes zero while its exact McNemar p-value is 0.0596, so the tests are reported separately.

Table 28: Expanded independent-evidence intervention. Task counts and top-1 accuracy use different aggregation units.

Population	Tasks	Base SI	ΔSI [95% CI]	Top-1 before / after
All	156	0.133	-0.015 [-0.036,+0.003]	98.2% to 98.9%
SI ≥ 0.05	16	2.646	-0.338 [-0.639,-0.035]	52.9% to 70.6%
SI ≥ 0.5	9	4.564	-0.564 [-0.901,-0.264]	11.1% to 55.6%

Table 29: Earlier small-sample independent-evidence follow-up; the high-SI subset contains four independent tasks.

Population	ΔSI [95% CI]
All: 60 tasks, 168 states	-0.0384 [-0.0831, +0.0011]
High-SI states (4 tasks)	-1.0954 [-1.2950, -0.8959]

Table 30: WCL masked-plan intervention versus an equal-token donor-plan placebo.

Writer	IA ΔWCL	Placebo ΔWCL	IA minus placebo
4B	-0.0173 [-0.0837, +0.0440]	-0.0666 [-0.1338, -0.0018]	+0.0493 [+0.0099, +0.0913]
8B	-0.0260 [-0.0958, +0.0417]	-0.0750 [-0.1508, -0.0012]	+0.0490 [-0.0581, +0.1599]
14B	+0.0207 [-0.0454, +0.0839]	+0.0015 [-0.0518, +0.0555]	+0.0192 [-0.0042, +0.0455]

Table 31: Missing-Atom source repair with equal-token repetition and wrong-family controls. Population-wide estimates shown here differ from the separately diagnosed-subset summary.

Writer	Atom	Repetition	Wrong family	Atom minus repeat	Atom minus wrong
4b	-0.0379 [-0.1073, +0.0321]	-0.0684 [-0.1271, -0.0127]*	-0.0553 [-0.1226, +0.0163]	+0.0304 [-0.0096, +0.0763]	+0.0174 [-0.0418, +0.0789]
8b	-0.0526 [-0.1133, +0.0046]	-0.0301 [-0.0589, +0.0058]	-0.0489 [-0.1108, +0.0118]	-0.0225 [-0.0860, +0.0436]	-0.0037 [-0.0681, +0.0550]
14b	-0.0156 [-0.0401, +0.0065]	-0.0070 [-0.0320, +0.0146]	-0.0133 [-0.0430, +0.0113]	-0.0086 [-0.0375, +0.0248]	-0.0022 [-0.0189, +0.0147]

Table 32: Selection outcomes for content-preserving rule permutations.

Exec.	Writer	Selected order	Accepted
4B	4B	verification-first	yes
4B	8B	reverse	yes
4B	14B	trigger-first	yes
8B	4B	verification-first	yes
8B	8B	trigger-first	yes
8B	14B	Base fallback	no
14B	4B	Base fallback	no
14B	8B	trigger-first	yes
14B	14B	trigger-first	yes

Table 33: Strict rule reordering on baseline EDM-diagnosed instances.

Quantity	Treatment minus Base [95% CI]
Delta WCL	+0.0194 [-0.0623, +0.0959]
Delta corrected EDM	-1.2199 [-1.7168, -0.7762]
Delta executor NLL	-1.2005 [-1.7058, -0.7392]

Table 34: Strict reordering on the 4B executor. The 8B-writer interval excludes zero, while the exact McNemar test gives p=0.0596.

Writer	Order	Base	RA	Delta pp [95% CI]	W/L	p
4B	verification-first	32.5%	31.8%	-0.7 pp [-5.5, +4.0]	22/24	0.8830
8B	reverse	23.4%	28.1%	+4.7 pp [+0.4, +9.1]	27/14	0.0596
14B	trigger-first	38.0%	39.8%	+1.8 pp [-3.3, +6.9]	27/22	0.5682

G STATIC SKILL CONSTRUCTION AND ABLATIONS

The five canonicalization conditions are alternative deployments. B standardizes admissible-action syntax, object identity, family phases, transformation prerequisites and recovery. B supplies and clarifies protocol information. C adds executor teach-back; D appends the same 183-word recovery reminder; E builds a 120–220-word patch from eight target-executor training failures. Mean prompt length increases from about 702 to 1462 words with teach-back, without improving the aggregate. The four post-canonical matrices and their summary follow; the initial A matrix is Table 11.

The granularity study uses nested G1 abstract principles, G2 task-family rules, G3 explicit steps, G4 state checklists and G5 recovery examples. The omission study cumulatively removes recovery, destination checks, transformation gates, identity/quantity and legal-action binding. Main Table 5 and the leave-one-out controls below separate content changes from length effects.

An earlier compiler used 96 calibration and 48 selection tasks with a one-success acceptance threshold. Three of nine adaptations were accepted, but all three declined on final tasks; the overall score changed from 50.8% to 49.4%. Some reruns used temperature 0.2. The stricter renderer used two disjoint 120-task sets and a five-net-success gate; zero of three pairs accepted. An independent atomic-patch study then used 160 training tasks and selected a destination guard with eight net wins. Its two held-out changes, -1.4 and $+1.5$ pp, were inconclusive.

Table 35: ALFWorld canonical matrix. Rows are writers and columns executors, 274 tasks per cell.

Writer	4B executor	8B executor	14B executor
4B	41.6%	46.4%	65.3%
8B	40.1%	46.7%	62.8%
14B	41.2%	47.1%	65.7%

Table 36: ALFWorld teachback matrix. Rows are writers and columns executors, 274 tasks per cell.

Writer	4B executor	8B executor	14B executor
4B	33.9%	46.7%	62.8%
8B	35.8%	46.4%	64.6%
14B	36.1%	44.2%	62.8%

Table 37: ALFWorld generic matrix. Rows are writers and columns executors, 274 tasks per cell.

Writer	4B executor	8B executor	14B executor
4B	38.3%	43.4%	69.0%
8B	38.7%	44.9%	61.7%
14B	37.2%	42.7%	65.7%

Table 38: ALFWorld coarse personal matrix. Rows are writers and columns executors, 274 tasks per cell.

Writer	4B executor	8B executor	14B executor
4B	36.5%	45.6%	65.0%
8B	38.3%	46.7%	63.9%
14B	31.8%	47.4%	63.1%

Table 39: Five alternative static skill conditions.

Condition	Overall	Cross-writer	Diagonal gap	Seen	Unseen
A	46.5%	44.9%	+4.9 pp	44.8%	48.3%
B	50.8%	50.5%	+0.9 pp	48.7%	53.0%
C	48.1%	48.4%	-0.7 pp	44.1%	52.3%
D	49.1%	48.8%	+0.9 pp	45.7%	52.6%
E	48.7%	48.7%	+0.1 pp	44.4%	53.2%

Table 40: Granularity results underlying Figure 3. G1 to G5 cumulatively add task-family rules, action steps, state checklists and recovery examples.

Executor	G1	G2	G3	G4	G5
4B	29.2%	31.0%	33.2%	32.1%	29.6%
8B	37.6%	33.6%	35.4%	35.0%	36.5%
14B	48.9%	48.5%	51.1%	58.0%	56.6%

Table 41: Cumulative omissions. O1 removes recovery, O2 destination checking, O3 transformation prerequisites, O4 identity/quantity, and O5 admissible-action binding.

Executor	O0	O1	O2	O3	O4	O5
4B	36.1%	34.3%	33.9%	33.2%	32.5%	33.2%
8B	41.6%	38.3%	36.9%	33.9%	35.0%	38.3%
14B	52.2%	51.5%	50.4%	48.5%	47.8%	46.4%

Table 42: Equal-word-count leave-one-out controls. The final column is removal of a decision rule minus removal of equally many padding words.

Exec.	Rule	Full	Rule removed	Padding removed	Content difference
4B	Loop	34.7%	34.3%	35.8%	-1.5 pp
4B	Destination	34.7%	32.8%	35.4%	-2.6 pp
4B	Transform	34.7%	36.5%	37.2%	-0.7 pp
4B	Identity/count	34.7%	34.7%	40.5%	-5.8 pp
4B	Action binding	34.7%	34.3%	35.4%	-1.1 pp
8B	Loop	38.0%	39.4%	39.8%	-0.4 pp
8B	Destination	38.0%	42.7%	41.6%	+1.1 pp
8B	Transform	38.0%	37.6%	40.5%	-2.9 pp
8B	Identity/count	38.0%	40.5%	40.9%	-0.4 pp
8B	Action binding	38.0%	39.4%	40.5%	-1.1 pp
14B	Loop	50.0%	48.9%	52.6%	-3.6 pp
14B	Destination	50.0%	49.6%	51.5%	-1.8 pp
14B	Transform	50.0%	53.3%	52.6%	+0.7 pp
14B	Identity/count	50.0%	52.2%	53.3%	-1.1 pp
14B	Action binding	50.0%	51.5%	53.6%	-2.2 pp

Table 43: Stricter renderer gate on three writer–executor pairs, with 120 calibration and 120 disjoint gate tasks per pair. All three choose Base.

Writer to executor	Calib. Base	Best adaptation	Selected	Gate Base
14B to 4B	35.8%	34.2%	B	40.0%
4B to 8B	43.3%	37.5%	B	51.7%
8B to 14B	58.3%	53.3%	B	65.0%

Table 44: Train-only single-rule patch selection on 160 tasks. The acceptance threshold is at least five net additional successes.

Candidate	Correct	Success	Wins / losses	Net
B	62/160	38.8%	0/0	+0
P_LOOP	62/160	38.8%	9/9	+0
P_TRANSFORM	66/160	41.2%	17/13	+4
P_IDENTITY	60/160	37.5%	11/13	-2
P_DESTINATION	70/160	43.8%	17/9	+8
P_SEARCH	63/160	39.4%	13/12	+1

H RUNTIME AUGMENTATION AND FAIR CONTROLS

Runtime interventions are evaluated as changes to the agent system. Static IR specifies phase and state fields in text. STATE maintains those fields externally from observations. FULL adds a verifier that corrects actions violating prerequisites or causing repeated unproductive behavior.

The 120-task control matches intervention frequency approximately: FULL makes 373 action overrides and RANDOM makes 388. FULL outperforms RANDOM by 10.0 points in this training-only comparison. The independent seen/unseen study includes padding and random-action controls. Random overrides themselves improve over Base, but targeted FULL adds 7.9 and 11.9 points beyond them.

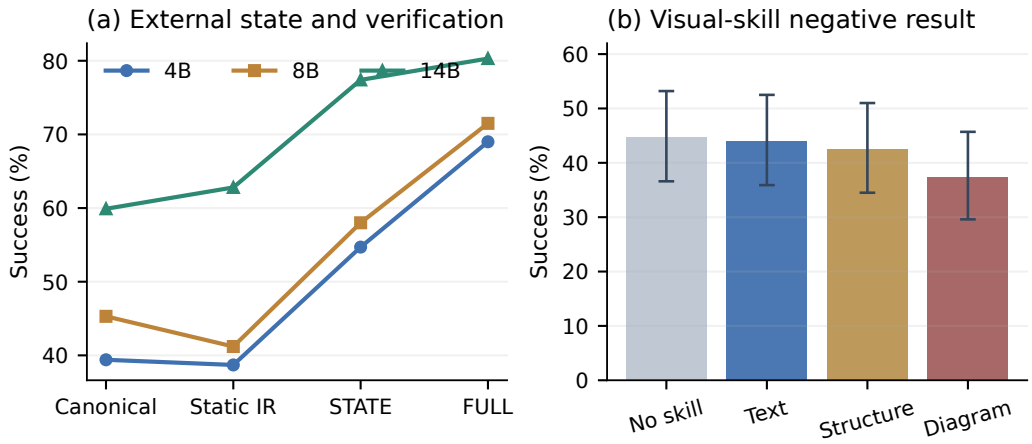


Figure 6: Two separate auxiliary studies. Left: state and verifier augmentation changes the runtime. Right: visual-skill conditions with marginal Wilson intervals; the paired comparisons are in Appendix J.

Table 45: Runtime augmentation: static IR, externally maintained state and state plus verifier.

Executor	Canonical	IR	STATE	FULL
4B	39.4%	38.7%	54.7%	69.0%
8B	45.3%	41.2%	58.0%	71.5%
14B	59.9%	62.8%	77.4%	80.3%

Table 46: Matched-frequency runtime control on 120 train-only tasks with the 4B executor. FULL performs 373 overrides; RANDOM performs 388.

Condition	Correct	Success	Wins	Losses	Overrides
B	56/120	46.7%	–	–	0
B_PAD	53/120	44.2%	11	14	0
IR	56/120	46.7%	13	13	0
STATE	71/120	59.2%	21	6	0
FULL	79/120	65.8%	28	5	373
RANDOM	67/120	55.8%	20	9	388

Table 47: Held-out atomic patch and runtime comparisons. Ref. and Treat. denote reference and treatment success rates.

Comparison	Ref.	Treat.	Delta	W/L	p	95% CI
Atomic seen	44.3%	42.9%	-1.4 pp	10/12	0.8318	[-7.9,+5.0]
Atomic unseen	41.8%	43.3%	+1.5 pp	11/9	0.8238	[-5.2,+8.2]
seen B_PAD_vs_B	39.3%	42.9%	+3.6 pp	15/10	0.4244	[-3.6,+10.7]
seen RANDOM_vs_B	39.3%	56.4%	+17.1 pp	30/6	6.96e-05	[+9.3,+25.0]
seen FULL_vs_RANDOM	56.4%	64.3%	+7.9 pp	18/7	0.04329	[+0.7,+15.0]
seen FULL_vs_B	39.3%	64.3%	+25.0 pp	40/5	7.878e-08	[+16.4,+33.6]
unseen B_PAD_vs_B	39.6%	44.8%	+5.2 pp	13/6	0.1671	[-0.7,+11.2]
unseen RANDOM_vs_B	39.6%	61.9%	+22.4 pp	32/2	6.938e-08	[+14.9,+29.9]
unseen FULL_vs_RANDOM	61.9%	73.9%	+11.9 pp	28/12	0.01659	[+3.0,+20.9]
unseen FULL_vs_B	39.6%	73.9%	+34.3 pp	52/6	3.158e-10	[+24.6,+44.0]

Table 48: Selected significant paired checks by model and seen/unseen split.

Comparison	Delta	W/L	p	95% CI
4b/seen/IA_vs_BASE	+14.3 pp	24/4	0.00018	[+7.1, +21.4] pp
4b/seen/IA_SHUFFLE_vs_BASE	+19.3 pp	32/5	7.428e-06	[+11.4, +27.1] pp
4b/seen/IA_COMPACT_vs_BASE	+14.3 pp	25/5	0.0003249	[+7.1, +21.4] pp
4b/seen/STATE_vs_B	+13.6 pp	23/4	0.0003107	[+6.4, +20.7] pp
4b/seen/FULL_vs_B	+25.0 pp	40/5	7.878e-08	[+16.4, +33.6] pp
4b/unseen/IA_COMPACT_vs_BASE	+8.2 pp	17/6	0.03469	[+1.5, +14.9] pp
4b/unseen/STATE_vs_B	+17.2 pp	26/3	1.524e-05	[+10.4, +24.6] pp
4b/unseen/FULL_vs_B	+34.3 pp	52/6	3.158e-10	[+24.6, +43.3] pp
8b/seen/STATE_vs_B	+8.6 pp	19/7	0.02896	[+1.4, +15.7] pp
8b/seen/FULL_vs_B	+16.4 pp	30/7	0.0001911	[+8.6, +24.3] pp
8b/unseen/STATE_vs_B	+17.2 pp	30/7	0.0001911	[+9.0, +26.1] pp
8b/unseen/FULL_vs_B	+36.6 pp	52/3	1.542e-12	[+27.6, +45.5] pp
14b/seen/STATE_vs_B	+11.4 pp	23/7	0.005223	[+4.3, +18.6] pp
14b/seen/FULL_vs_B	+14.3 pp	26/6	0.0005351	[+7.1, +21.4] pp
14b/unseen/STATE_vs_B	+23.9 pp	35/3	6.678e-08	[+15.7, +32.1] pp
14b/unseen/FULL_vs_B	+26.9 pp	40/4	1.705e-08	[+17.9, +35.8] pp

I BUDGET, EXTRACTION AND EXECUTION ROBUSTNESS

The Math budget study uses the same 240 frozen questions at 8192 output tokens. Its native matrix changes the writer ranking. The three no-skill truncation counts are 24, 28 and 34; native-skill counts aggregate three writer cells per executor and therefore have a different denominator. Padded variants use approximately 604–607 tokens.

The extraction study resamples each writer's own 400 trajectories using seeds 41, 3141 and 2718. The fixed-skill study instead executes the seed-41 skill three times. These two SDs quantify extraction and execution variability, respectively. A separate matched-source setting uses 250 Qwen3-32B trajectories.

ALFWorld's complete three-run matrix appears in main Table 2. WebShop's repeat-specific and clustered contrasts appear in main Table 10. Its no-skill comparison averages Base over writers and runs, unlike the primary single-run 3×3 matrix. Mistral transfer accuracy remains near the floor.

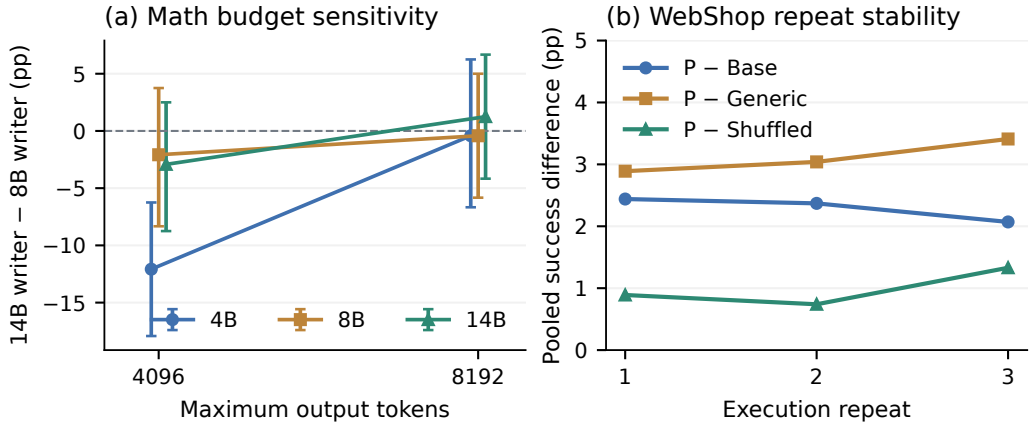


Figure 7: **Output budget and repeated execution.** Left: paired 95% intervals at the two evaluated output budgets on the same 240 Math problems; x positions are slightly offset for visibility. Right: the three measured WebShop repeat effects. The repeated P–Shuffled aggregate remains inconclusive despite positive point estimates.

Table 49: Math native skills at an 8192-token output budget. Rows are writers, columns executors.

Writer	4B executor	8B executor	14B executor
4B	30.8%	35.4%	35.0%
8B	31.2%	35.0%	35.8%
14B	30.8%	34.6%	37.1%

Table 50: No-skill accuracy and truncation counts at 8192 tokens; the last column compares 14B and 8B writers.

Executor	No skill	Truncated (no skill)	Truncated (3 skills)	14B minus 8B [95% CI]
4B	35.4%	24	60	-0.42 [-6.67, +6.25] pp
8B	35.4%	28	56	-0.42 [-5.83, +5.00] pp
14B	34.6%	34	83	+1.25 [-4.17, +6.67] pp

Table 51: Math extraction-artifact mean ± SD over seeds 41, 3141 and 2718, using each writer's own trajectory source.

Writer	4B executor	8B executor	14B executor
4B	32.4% ± 3.6	31.1% ± 5.0	35.8% ± 1.8
8B	33.5% ± 3.2	36.2% ± 2.0	37.8% ± 2.5
14B	31.0% ± 4.3	38.1% ± 1.4	36.0% ± 2.3

Table 52: Math mean ± run SD for three executions with a fixed seed-41 skill.

Writer	4B executor	8B executor	14B executor
4B	31.4% ± 0.4	35.8% ± 1.6	36.9% ± 1.4
8B	34.9% ± 2.7	36.5% ± 1.4	37.1% ± 1.2
14B	27.1% ± 2.7	35.3% ± 1.0	36.8% ± 0.2

Table 53: WebShop no-skill control across repeated executions. Base is averaged over writers. Differences use unrounded outcomes.

Executor	No skill	Mean Base	Base minus no skill
4B	20.7%	32.8%	+12.11 pp
8B	27.0%	32.4%	+5.44 pp
14B	27.7%	28.9%	+1.22 pp

Table 54: Math transfer to Mistral-7B-Instruct-v0.3.

Condition	Accuracy	Correct	Truncated	vs no skill [95% CI]
no_skill	3.3%	8/240	7	–
native_w4b	1.2%	3/240	4	-2.08 [-5.00, +0.42] pp
native_w8b	1.7%	4/240	5	-1.67 [-4.17, +0.83] pp
native_w14b	0.4%	1/240	7	-2.92 [-5.42, -0.83] pp

J VISUAL-SKILL NEGATIVE EXPERIMENT

The visual study uses Qwen3-VL-8B-Instruct with temperature 0, seed 41 and a 256-token action-generation budget. Four conditions share 134 tasks and a five-rule generic skill: no skill, text, text plus structure, and text plus diagram. The runner uses numbered admissible actions, a 50-step cap, a recent 12-step summary and eight retained turns. The diagram is compiled from the skill before execution.

Of 18,905 model calls, 42 length terminations are concentrated in one Text episode. Across the 106 tasks with no parsing failure in any condition, success counts are 55, 50, 50 and 45. All pairwise exact tests are inconclusive, and Diagram has the lowest point estimate.

Table 55: Qwen3-VL-8B visual-skill controls on 134 ALFWorld tasks. Intervals are marginal Wilson intervals.

Condition	Success	Wilson 95% CI	Mean steps	Parse-failure episodes
No Skill	44.8%	36.6-53.2%	34.8	11
Text	44.0%	35.9-52.5%	34.5	4
Text + Structure	42.5%	34.5-51.0%	35.1	10
Text + Diagram	37.3%	29.6-45.7%	36.6	12

Table 56: Visual-skill success counts by task family, not percentages.

Family	N	No skill	Text	Structure	Diagram
Light	18	6	4	5	4
Put	24	24	24	23	23
Clean	31	18	19	16	13
Cool	21	2	2	2	1
Heat	23	2	3	3	3
Two objects	17	8	7	8	6

Table 57: All paired visual-skill comparisons. None has exact McNemar p below 0.05.

Comparison	First-only wins	Second-only wins	p
Text vs No Skill	13	14	1.000
Structure vs Text	14	16	0.856
Diagram vs Text	11	20	0.150
Diagram vs Structure	10	17	0.248
Diagram vs No Skill	8	18	0.076

K DECISION-STATE CASES

Case selection ranks baseline diagnostic values and keeps the top two distinct tasks for each component, independently of treatment outcomes. The six cases examine local decision states rather than complete rollouts.

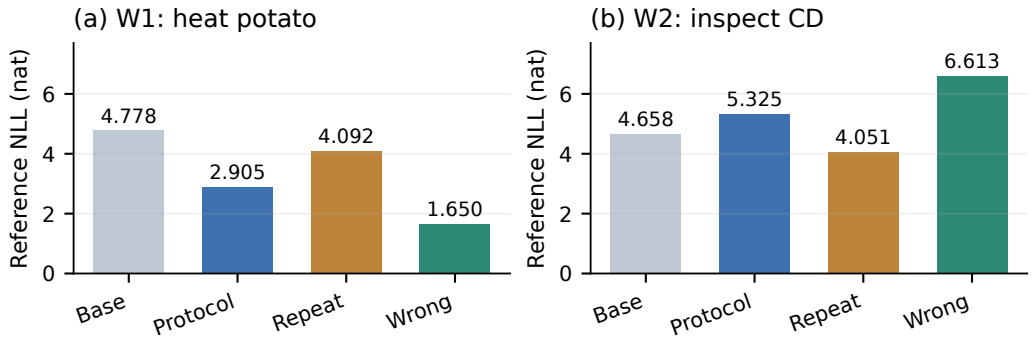


Figure 8: Reference action NLL in the W1 and W2 decision-state cases. The three conditions compare protocol restoration, repetition and wrong-family structure.

Table 58: Six baseline-ranked cases. Each diagnostic selects the top two distinct tasks by the pre-treatment score, independently of treatment outcomes.

Case	State ID	Task	Observation (nat)
S1	unseen:62410bea7e123fdd:2	put a cool mug in coffeemachine.	SI 13.571 to 13.876
S2	unseen:e100f774cee6965d:2	heat some mug and put it in coffeemachine.	SI 13.439 to 14.501
W1	unseen:d41f7aa002d42ab6:2	heat some potato and put it in garbagecan.	WCL 4.706; delta -1.872
W2	seen:f91da0fb129fac1d:2	look at cd under the desklamp.	WCL 4.657; delta +0.667
E1	unseen:aa99121a333549e9:4	examine the book with the desklamp.	EDM 18.498 to 17.873
E2	seen:3cfcc5368f007d3e:3	examine the statue with the desklamp.	EDM 17.849 to 13.235

Table 59: Reference NLL in the two WCL cases. Wrong-family structure achieves the lowest NLL in W1; the protocol increases NLL in W2.

Case	Base	Protocol	Repeat	Wrong family
W1: potato	4.778	2.905	4.092	1.650
W2: CD	4.658	5.325	4.051	6.613

S1 and S2: independent evidence does not resolve source binding. Both tasks require `take mug 2 from countertop 1`; before and after extra evidence, the selected action is `take mug 2 from coffeemachine 1`. SI rises from 13.571 to 13.876 in S1 and from 13.439 to 14.501 in S2. The added donor schemas describe a generic take–transform–deliver sequence and deliberately omit task-specific object identity and location. The unchanged action points to unresolved object-location binding.

W1 and W2: content restoration is not uniformly specific. For W1 the task is to heat a potato and place it in a garbagecan. The correct next command is `take potato 1 from countertop 1`. A training-derived phase protocol lowers reference NLL to 2.905, but wrong-family structure reaches 1.650. W2 asks to inspect a CD under a desklamp, with correct command `take cd 1 from desk 1`. Its reference NLL instead rises from 4.658 to 5.325 under the protocol; repetition achieves 4.051.

E1 and E2: likelihood changes without a changed top action. E1 requires `use desklamp 1`; the executor continues choosing `examine book 1` while corrected EDM decreases from 18.498 to 17.873. E2 requires `take statue 2 from sidetable 1`; it continues choosing `look` while corrected EDM drops from 17.849 to 13.235. Both reorder exactly the same skill lines, improving the corrected likelihood without changing the selected action.

Representative source text. W1's protocol tells the executor to treat phase arrows as prerequisite gates and preserve object identity through transformation and delivery. E1's reordered prefix begins with recovery after a failed step, while E2 foregrounds refreshing observations after a state change.

L ADDITIONAL APPWORLD AND MATHEMATICAL STUDIES

These additional studies use separate evaluation data and experience-source policies from the three primary benchmarks. AppWorld has normal and challenge partitions and a different action/API interface from ALFWorld.

Table 60: AppWorld writer–executor matrix, with 168 normal and 417 challenge tasks per cell.

Split	Executor	4B writer	8B writer	14B writer
Normal	4B	14/168 (8.33%)	12/168 (7.14%)	13/168 (7.74%)
Normal	8B	26/168 (15.48%)	40/168 (23.81%)	34/168 (20.24%)
Normal	14B	41/168 (24.40%)	45/168 (26.79%)	50/168 (29.76%)
Challenge	4B	16/417 (3.84%)	16/417 (3.84%)	17/417 (4.08%)
Challenge	8B	28/417 (6.71%)	28/417 (6.71%)	25/417 (6.00%)
Challenge	14B	38/417 (9.11%)	37/417 (8.87%)	46/417 (11.03%)

The Qwen mathematical study contains 100 distinct ID tasks and 30 AIME 2026 problems evaluated eight times each. The Luna condition uses 100 selected common failures as an experience source; the contrastive condition pairs Qwen failures with a Luna trajectory. These conditions vary the experience source, whereas the primary runs use each writer's own trajectories.

Table 61: Mathematical skill transfers under alternative experience sources. The AIME set comprises 30 distinct AIME 2026 problems with eight generations per problem. Rows are executors, columns writers.

Source	Evaluation	Exec.	4B writer	8B writer	14B writer
Qwen self	ID: 100	4B	28.00%	34.00%	30.00%
		8B	37.00%	40.00%	33.00%
		14B	40.00%	40.00%	43.00%
Qwen self	AIME: 30 x 8	4B	9.17%	8.33%	5.83%
		8B	8.75%	11.25%	11.25%
		14B	9.58%	10.00%	11.25%
Luna trajectories	ID: 100	4B	35.00%	30.00%	25.00%
		8B	33.00%	39.00%	34.00%
		14B	40.00%	38.00%	39.00%
Luna trajectories	AIME: 30 x 8	4B	8.75%	5.83%	3.33%
		8B	10.42%	7.08%	8.33%
		14B	11.67%	7.50%	8.75%
Contrastive	ID: 100	4B	29.00%	29.00%	34.00%
		8B	29.00%	41.00%	42.00%
		14B	29.00%	41.00%	44.00%
Contrastive	AIME: 30 x 8	4B	9.17%	9.58%	7.08%
		8B	8.33%	9.58%	12.08%
		14B	7.92%	10.00%	11.25%

Table 62: No-skill controls for the Qwen mathematical study. Each executor shares the same baseline across writer conditions.

Executor	ID: 100 tasks	AIME: 30 tasks x 8
4B	28/100 (28.00%)	26/240 (10.83%)
8B	37/100 (37.00%)	21/240 (8.75%)
14B	39/100 (39.00%)	20/240 (8.33%)

The AIME accuracies aggregate 240 generations over 30 independent problems, with eight generations clustered under each problem ID.

M FROZEN SKILL FILES AND MEASURED OVERHEAD

The three-benchmark studies use 21 complete skills: three combined ALFWorld skills, three native Math skills and five WebShop category files for each writer. Table 63 gives their lengths and identifying hashes.

Table 63: The 21 frozen skills used in the three-benchmark studies. SHA-256 prefixes identify the skill texts. Words are counted by whitespace.

ID	Benchmark	Writer	Bytes	Words	SHA-256 prefix
1	ALFWorld	4B	2300	348	b2d23969be76
2	ALFWorld	8B	7513	1114	db276c2b5b9a
3	ALFWorld	14B	2882	424	d3005be291f0
4	Math	4B	1417	217	887a9250de32
5	Math	8B	2411	387	f6ae0e0012c4
6	Math	14B	2274	353	f1a6554a6199
7	WebShop	4B	991	171	c33d06a535a5
8	WebShop	4B	1334	212	7440aace2b00
9	WebShop	4B	1158	204	f8bf21d04a91
10	WebShop	4B	1252	205	3b58e2b19841
11	WebShop	4B	1367	224	a50fa9389d58
12	WebShop	8B	1091	176	67f80cb62c6d
13	WebShop	8B	889	146	80e8a0fc6f4c
14	WebShop	8B	1058	179	625f1b04818f
15	WebShop	8B	1245	204	08800c4a73f7
16	WebShop	8B	957	158	34aaabf5e3bb
17	WebShop	14B	1354	218	a1f65749af87
18	WebShop	14B	1056	171	25555221012c
19	WebShop	14B	1924	293	9a3003e12072
20	WebShop	14B	1316	220	9c2ba00fb692
21	WebShop	14B	1073	171	94a3e0ab194e

Skill length does not increase monotonically with writer size. The ALFWorld 8B skill is 7,513 bytes and 1,114 whitespace words, compared with 2,300/348 and 2,882/424 for 4B and 14B. In Math the 8B skill is 2,411 bytes and 387 words; the 14B skill is 2,274 bytes and 353 words. Thus the 14B-versus-8B failure occurs despite the shorter skill.

Measured overhead also includes the 702-to-1462-word teach-back expansion, 135 WebShop tutor candidates, 108 valid audit outcomes, and runtime override counts.

N CROSS-MODEL VALIDATION OF THE WRITER–EXECUTOR GAP

This experiment extends the writer–executor comparison to Gemma and Gemini using common source experience. Gemini-written skills achieve the highest success rate for the Gemma-4-12B executor, whereas Gemma-4-12B-written skills perform best for the Gemma-4-31B executor.

Protocol. Each writer receives the same 100 frozen ALFWorld trajectories: 43 successful and 57 failed episodes. It first extracts reusable lessons from individual trajectories and then merges them into one global skill. Each executor uses each of the three resulting skills and a No Skill baseline, giving 12 conditions. Every condition evaluates the same 274 held-out ALFWorld TextWorld tasks: 140 `valid_seen` tasks and 134 `valid_unseen` tasks. Execution uses a maximum of 50 steps, temperature 0, thinking disabled and the same admissible-action interface. A response that cannot be parsed into a legal admissible action triggers the runner's predefined fallback and the episode continues. Task outcomes are saved individually. G12, G31 and GF denote Gemma-4-12B, Gemma-4-31B and Gemini-3.8-Flash, respectively.

Table 64: Complete cross-model ALFWorld success counts. All conditions have $N = 274$. G12 = Gemma-4-12B; G31 = Gemma-4-31B; GF = Gemini-3.8-Flash.

	Executor		
Skill writer	G12	G31	GF
No Skill	72/274 (26.3%)	167/274 (60.9%)	268/274 (97.8%)
G12	84/274 (30.7%)	175/274 (63.9%)	265/274 (96.7%)
G31	85/274 (31.0%)	160/274 (58.4%)	268/274 (97.8%)
GF	127/274 (46.4%)	161/274 (58.8%)	270/274 (98.5%)

Table 65: Cross-model success counts and rates on the two held-out ALFWorld splits.

Skill writer	Executor	`valid_seen`	`valid_unseen`
No Skill	G12	38/140 (27.1%)	34/134 (25.4%)
No Skill	G31	86/140 (61.4%)	81/134 (60.4%)
No Skill	GF	138/140 (98.6%)	130/134 (97.0%)
G12	G12	43/140 (30.7%)	41/134 (30.6%)
G12	G31	87/140 (62.1%)	88/134 (65.7%)
G12	GF	137/140 (97.9%)	128/134 (95.5%)
G31	G12	39/140 (27.9%)	46/134 (34.3%)
G31	G31	81/140 (57.9%)	79/134 (59.0%)
G31	GF	137/140 (97.9%)	131/134 (97.8%)
GF	G12	60/140 (42.9%)	67/134 (50.0%)
GF	G31	84/140 (60.0%)	77/134 (57.5%)
GF	GF	138/140 (98.6%)	132/134 (98.5%)

Paired effects. Table 66 compares each skill with No Skill on the same tasks. A win means that the skill condition succeeds and No Skill fails; a loss is the reverse; a tie means equal binary outcomes. For GF-written skills on the G12 executor, the 65 wins and 10 losses yield a net gain of 55 successes, or $+20.1$ percentage points. The paired bootstrap interval is $[+14.2, +25.9]$ points, with exact McNemar $p = 5.154 \times 10^{-11}$.

Table 66: Paired skill-versus-No-Skill effects on 274 tasks. Intervals are task-level paired bootstrap 95% intervals. p denotes the unadjusted two-sided exact McNemar test. W/L/T = wins/losses/ties. Effects and intervals are in percentage points.

Writer	Executor	Effect	95% CI	p	W/L/T
G12	G12	$+4.4$	$[-0.4, +9.1]$	0.08843	27/15/232
G12	G31	$+2.9$	$[-2.2, +8.0]$	0.3317	30/22/222
G12	GF	-1.1	$[-2.9, +0.7]$	0.4531	2/5/267
G31	G12	$+4.7$	$[+0.0, +9.5]$	0.07245	29/16/229
G31	G31	-2.6	$[-7.7, +2.6]$	0.3916	21/28/225
G31	GF	$+0.0$	$[-1.1, +1.1]$	1	1/1/272
GF	G12	$+20.1$	$[+14.2, +25.9]$	5.154×10^{-11}	65/10/199
GF	G31	-2.2	$[-8.8, +4.4]$	0.5811	38/44/192
GF	GF	$+0.7$	$[-0.7, +2.2]$	0.625	3/1/270

Writer–executor interactions. Let $R(w, e)$ be the success rate of writer w's skill on executor e. For model pair A, B, we evaluate

$$\Delta_{A,B} = [R(A, A) - R(B, A)] - [R(A, B) - R(B, B)]. \tag{23}$$

The corresponding task-level paired bootstrap intervals appear in Table 67. For G12/GF, $\Delta = -13.9$ percentage points with interval $[-20.1, -8.0]$: the advantage of GF-written over G12-written skills is substantially larger on G12 than on GF. The interaction measures how the size of a writer advantage changes across executors. The matrix also contains a point-estimate reversal between the G12 and GF writers across the G12 and G31 executors: GF leads by 15.7 points on G12, while G12 leads by 5.1 points on G31.

Table 67: Writer–executor interaction contrasts from Eq. 23. Effects and paired bootstrap 95% intervals are in percentage points.

Model pair (A, B)	N	$\Delta_{A,B}$	95% CI
G12, G31	274	-5.8	$[-13.1, +1.5]$
G12, GF	274	-13.9	$[-20.1, -8.0]$
G31, GF	274	$+0.4$	$[-7.3, +7.7]$

Action-format compatibility. Table 68 reports parse failures per action step. For the G31 executor, the failure rate ranges from 16.4% with G12-written skills to 33.7% with GF-written skills. For the GF executor, it ranges from 8.3% to 12.7% across skill conditions. These measurements show that writer choice changes adherence to the execution interface as well as task outcomes. The end-to-end matrix therefore measures compatibility of the complete skill with the executor, including action formatting and procedural guidance.

Table 68: Cross-model action-format diagnostics. A parse failure invokes the runner's predefined fallback. The denominator is the number of action steps, not the number of tasks.

Skill writer	Executor	Parse failures	Action steps	Failure rate
No Skill	G12	974	11472	8.5%
No Skill	G31	1648	7955	20.7%
No Skill	GF	226	2491	9.1%
G12	G12	1583	11038	14.3%
G12	G31	1267	7740	16.4%
G12	GF	325	2565	12.7%
G31	G12	1904	11241	16.9%
G31	G31	2217	8138	27.2%
G31	GF	240	2500	9.6%
GF	G12	898	9783	9.2%
GF	G31	2950	8764	33.7%
GF	GF	200	2399	8.3%

O CROSS-FAMILY SKILLTUTOR WITH A GEMMA-31B TUTOR

A Gemma-4-31B tutor produces information-bounded, executor-conditioned skill capsules that improve held-out action prediction across a Qwen-only 3×3 matrix and remain effective when Gemma-4-12B is introduced as both writer and executor.

O.1 TWO CROSS-FAMILY EXPERIMENTS

Qwen-only control. The writers and executors are Qwen3-4B-Instruct, Qwen3-8B and Qwen3-14B, forming a full 3×3 matrix. Gemma-4-31B-it serves as the tutor. This experiment tests whether SKILLTUTOR remains effective when the tutor comes from a different model family.

Mixed matrix. The writers and executors are Qwen3-4B-Instruct, Qwen3-8B and Gemma-4-12B, again forming a full 3×3 matrix. The tutor remains Gemma-4-31B-it. This experiment extends the writer, executor and tutor chain across model families. Qwen3-32B is the auditor and reference model in both experiments. All inference is local, with thinking disabled. The fixed seed is 41. In the tables, Q4, Q8, Q14 and G12 denote Qwen3-4B-Instruct, Qwen3-8B, Qwen3-14B and Gemma-4-12B, respectively.

O.2 PROTOCOL AND MEASUREMENTS

The tutor reads the original skill and execution errors exposed by Calibration, then generates three short candidate capsules: c1 directly binds triggers to actions; c2 contrasts appropriate and inappropriate actions; and c3 explicitly binds object, receptacle and inventory states to action prerequisites. The tutor reorganizes, compresses and emphasizes information already in the skill.

An auditor filters candidates containing new commands, environment facts, search locations, recovery rules or causal claims not entailed by the source skill. Among eligible candidates, Selection chooses the one with the lowest task-mean NLL and accepts it only if it improves on Base. The chosen capsule is then frozen and evaluated once on Final. Generic is a length-controlled, deterministic summary that is not executor-conditioned; Shuffled supplies the personalized capsule for a different executor.

Table 69: Shared data split for the two Gemma-31B tutor experiments. The Final set contains 30 seen and 30 unseen tasks.

Stage	Tasks	States	Critical states	Role
Calibration	60	351	168	Diagnose errors and generate candidates
Selection	30	173	85	Choose and freeze one capsule per cell
Final	60	353	168	Evaluate the frozen selection

Calibration, Selection and Final have no overlapping tasks. The six families are Light (look at an object in light), Put (pick and place), Clean (clean then place), Cool (cool then place), Heat (heat then place) and Two objects (pick two objects and place). Final contains five tasks per family and split, for 30 seen and 30 unseen tasks. Targets are environment-validated sets of correct actions; 37 Final states admit multiple correct actions. Measurements first average critical states within each task and then weight tasks equally. Confidence intervals use 20,000 paired task-bootstrap samples, stratified by family and split.

Metrics. NLL is the negative log-likelihood of the correct-action set. ExecGain is executor NLL under Base minus executor NLL under Personal. RefGain is the corresponding reduction for Qwen3-32B on the same texts. CapsuleGain is ExecGain minus RefGain, measuring the executor's additional gain after reference correction. The Generic and Shuffled contrasts are their respective NLLs minus Personal NLL. ΔTop-1 is Personal minus Base next-action accuracy, in percentage points. Positive differences favor Personal. Measurements use fixed key states.

O.3 POOLED AND SPLIT RESULTS

Table 70: Pooled results across nine writer–executor cells in each matrix. Brackets give the 95% confidence intervals; positive differences favor Personal. Top-1 changes are percentage points.

Metric	Qwen-only control	Mixed matrix
ExecGain	+0.2887 [+0.1730, +0.3938]	+0.3098 [+0.2281, +0.3898]
RefGain	+0.1432 [+0.0985, +0.1876]	+0.1479 [+0.1076, +0.1881]
CapsuleGain	+0.1455 [+0.0486, +0.2428]	+0.1620 [+0.0904, +0.2372]
vs Generic	+0.3285 [+0.2007, +0.4461]	+0.3422 [+0.2355, +0.4442]
vs Shuffled	+0.0800 [+0.0421, +0.1231]	+0.0616 [-0.0127, +0.1412]
ΔTop-1 (pp)	+2.85 pp [+1.48, +4.44]	+4.34 pp [+3.09, +5.65]

The Qwen-only control yields ExecGain $+0.2887$ and CapsuleGain $+0.1455$; their 95% intervals are $[+0.1730, +0.3938]$ and $[+0.0486, +0.2428]$. Personal also improves over Generic and Shuffled, with intervals above zero, and increases Top-1 accuracy by 2.85 points. With Gemma-4-12B introduced as writer and executor, ExecGain is $+0.3098$ and CapsuleGain is $+0.1620$, with intervals $[+0.2281, +0.3898]$ and $[+0.0904, +0.2372]$. The mixed matrix retains positive significant improvements over Generic and in Top-1 accuracy, which increases by 4.34 points. The mixed Shuffled contrast is $+0.0616$ with interval $[-0.0127, +0.1412]$.

The positive reference-corrected pooled effects in both experiments indicate greater NLL reductions for the target executors than for the reference model on the same texts. Tables 71 and 72 report the seen and unseen results separately.

Table 71: Qwen-only control: seen and unseen Final results, with 30 tasks in each split. Brackets give the 95% confidence intervals.

Metric	Seen	Unseen
ExecGain	+0.3596 [+0.1749, +0.5290]	+0.2177 [+0.0789, +0.3368]
CapsuleGain	+0.2346 [+0.0703, +0.3967]	+0.0564 [-0.0497, +0.1619]
vs Generic	+0.3743 [+0.1800, +0.5493]	+0.2827 [+0.1168, +0.4329]
vs Shuffled	+0.0833 [+0.0394, +0.1339]	+0.0767 [+0.0188, +0.1466]

Table 72: Mixed matrix: seen and unseen Final results, with 30 tasks in each split. Brackets give the 95% confidence intervals.

Metric	Seen	Unseen
ExecGain	+0.2962 [+0.1576, +0.4322]	+0.3234 [+0.2354, +0.4077]
CapsuleGain	+0.1688 [+0.0326, +0.3125]	+0.1551 [+0.1056, +0.2038]
vs Generic	+0.3062 [+0.1456, +0.4634]	+0.3783 [+0.2326, +0.4962]
vs Shuffled	+0.0457 [-0.0954, +0.1949]	+0.0776 [+0.0353, +0.1215]

O.4 COMPLETE WRITER–EXECUTOR CELLS

All nine cells in each experiment have positive Final ExecGain. The notation E←W identifies the executor and the writer of its original skill.

Table 73: Qwen-only control: all nine Final cells. The cell notation is executor ← writer. All nine cells use a selected capsule. Brackets give the 95% confidence intervals.

E←W	Base	Personal	ExecGain	CapsuleGain
Q4←Q4	0.8197	0.7993	+0.0204 [-0.1449, +0.2073]	-0.0762 [-0.2782, +0.1336]
Q4←Q8	1.0539	0.3036	+0.7503 [+0.3990, +1.0838]	+0.4404 [+0.0980, +0.7734]
Q4←Q14	1.6389	0.5631	+1.0757 [+0.6394, +1.4417]	+1.1050 [+0.6905, +1.4558]
Q8←Q4	0.3415	0.2913	+0.0501 [-0.0005, +0.1102]	-0.0736 [-0.1520, +0.0049]
Q8←Q8	0.3704	0.2945	+0.0759 [+0.0124, +0.1465]	-0.2133 [-0.2819, -0.1410]
Q8←Q14	0.4236	0.2287	+0.1949 [+0.0497, +0.3996]	+0.1739 [+0.0156, +0.3784]
Q14←Q4	0.3620	0.3418	+0.0202 [-0.0548, +0.1038]	-0.0918 [-0.1907, +0.0113]
Q14←Q8	0.4934	0.3707	+0.1227 [+0.0681, +0.1858]	-0.1766 [-0.2855, -0.0473]
Q14←Q14	0.5659	0.2784	+0.2875 [+0.1674, +0.4187]	+0.2217 [+0.1115, +0.3367]

Table 74: Qwen-only control: Generic, Shuffled and Top-1 controls for every Final cell. Brackets give the 95% confidence intervals; Top-1 changes are percentage points.

E←W	vs Generic	vs Shuffled	ΔTop-1 (pp)
Q4←Q4	+0.3245 [+0.0710, +0.6160]	-0.0606 [-0.1936, +0.0756]	+0.83 pp [-1.11, +3.06]
Q4←Q8	+0.8814 [+0.4909, +1.2579]	+0.6462 [+0.3349, +0.9572]	+5.83 pp [+2.78, +9.17]
Q4←Q14	+0.8707 [+0.4978, +1.2042]	+0.0664 [-0.0780, +0.1889]	+6.67 pp [+3.33, +10.00]
Q8←Q4	+0.0266 [-0.0206, +0.0738]	+0.0413 [+0.0053, +0.0823]	+1.67 pp [+0.00, +4.17]
Q8←Q8	+0.1068 [+0.0234, +0.2094]	-0.0823 [-0.2232, +0.0252]	+2.50 pp [+0.00, +5.00]
Q8←Q14	+0.2213 [+0.0546, +0.4181]	+0.0121 [-0.0309, +0.0650]	+2.78 pp [+0.28, +5.28]
Q14←Q4	+0.0263 [-0.0409, +0.0976]	-0.0196 [-0.0689, +0.0321]	-0.56 pp [-1.67, +0.00]
Q14←Q8	+0.2178 [+0.1459, +0.2907]	-0.0994 [-0.1680, -0.0364]	+2.50 pp [+0.83, +5.00]
Q14←Q14	+0.2810 [+0.1916, +0.3827]	+0.2160 [+0.0985, +0.3389]	+3.47 pp [+0.97, +6.25]

Table 75: Mixed matrix: all nine Final cells. The cell notation is executor ← writer. All nine cells use a selected capsule. Brackets give the 95% confidence intervals.

E←W	Base	Personal	ExecGain	CapsuleGain
Q4←Q4	0.8179	0.7983	+0.0197 [-0.1446, +0.2064]	-0.0769 [-0.2778, +0.1326]
Q4←Q8	1.0457	0.3036	+0.7421 [+0.3948, +1.0733]	+0.4327 [+0.0939, +0.7628]
Q4←G12	1.9622	0.8158	+1.1463 [+0.8454, +1.5037]	+1.1303 [+0.8277, +1.4825]
Q8←Q4	0.3415	0.2913	+0.0501 [-0.0005, +0.1102]	-0.0733 [-0.1519, +0.0051]
Q8←Q8	0.3704	0.2945	+0.0759 [+0.0124, +0.1465]	-0.2142 [-0.2830, -0.1415]
Q8←G12	0.5335	0.3713	+0.1621 [+0.0038, +0.3246]	+0.1036 [-0.0525, +0.2649]
G12←Q4	0.6554	0.4459	+0.2096 [+0.0732, +0.3401]	+0.1231 [+0.0130, +0.2302]
G12←Q8	0.4506	0.3029	+0.1477 [-0.0010, +0.3103]	-0.1632 [-0.3059, -0.0135]
G12←G12	0.6819	0.4469	+0.2350 [+0.0960, +0.3586]	+0.1956 [+0.0532, +0.3223]

Table 76: Mixed matrix: Generic, Shuffled and Top-1 controls for every Final cell. Brackets give the 95% confidence intervals; Top-1 changes are percentage points.

E←W	vs Generic	vs Shuffled	ΔTop-1 (pp)
Q4←Q4	+0.3266 [+0.0732, +0.6182]	-0.0595 [-0.1922, +0.0764]	+0.00 pp [-1.11, +1.11]
Q4←Q8	+0.8813 [+0.4908, +1.2578]	+0.6462 [+0.3349, +0.9572]	+5.83 pp [+2.78, +9.17]
Q4←G12	+1.0890 [+0.7614, +1.4350]	+0.0384 [-0.1743, +0.2528]	+8.33 pp [+5.56, +11.11]
Q8←Q4	+0.0265 [-0.0206, +0.0738]	-0.0334 [-0.2751, +0.1915]	+1.67 pp [+0.00, +4.17]
Q8←Q8	+0.1068 [+0.0234, +0.2094]	-0.1219 [-0.3069, +0.0304]	+2.50 pp [+0.00, +5.00]
Q8←G12	+0.2334 [+0.0995, +0.3777]	-0.1777 [-0.3509, -0.0378]	+3.06 pp [-0.83, +7.22]
G12←Q4	+0.0228 [-0.1182, +0.1619]	+0.0686 [-0.0974, +0.2318]	+7.22 pp [+4.17, +10.28]
G12←Q8	+0.2015 [+0.0701, +0.3561]	+0.0479 [-0.0963, +0.2563]	+4.72 pp [+0.83, +8.61]
G12←G12	+0.1923 [+0.0788, +0.3102]	+0.1461 [+0.0649, +0.2364]	+5.69 pp [+1.94, +9.44]

O.5 Selection and information audit

In the Qwen-only control, all 27 candidates pass the audit and all nine cells accept a capsule: one c1, eight c2 and no c3. In the mixed matrix, 22 of 27 candidates pass. Five c2 candidates with unsupported claims are filtered, while every cell retains eligible candidates. All nine cells accept a capsule: four c1, three c2 and two c3. Selection-to-Final effects retain a positive direction in all nine cells in both experiments; the Pearson correlations are 0.939 and 0.961, respectively. Thus, the selected gains carry over to the isolated Final set, and the audit filters unsupported content while preserving coverage of all cells.

Table 77: Qwen-only control: candidate selection on the separate Selection set. All nine selected candidates improve on Base and are accepted. Eligible counts refer to candidates that pass the information audit.

E←W	Base	Style	Selected	ΔNLL	Eligible
Q4←Q4	0.7499	c2	0.4764	+0.2735	3/3
Q4←Q8	1.1919	c2	0.4240	+0.7679	3/3
Q4←Q14	1.4833	c2	0.6307	+0.8526	3/3
Q8←Q4	0.4523	c2	0.3101	+0.1422	3/3
Q8←Q8	0.3821	c1	0.3043	+0.0778	3/3
Q8←Q14	0.3624	c2	0.2340	+0.1284	3/3
Q14←Q4	0.4399	c2	0.4103	+0.0296	3/3
Q14←Q8	0.6225	c2	0.4750	+0.1476	3/3
Q14←Q14	0.5504	c2	0.4064	+0.1440	3/3

Table 78: Mixed matrix: candidate selection on the separate Selection set. All nine selected candidates improve on Base and are accepted. Eligible counts refer to candidates that pass the information audit.

E←W	Base	Style	Selected	ΔNLL	Eligible
Q4←Q4	0.7525	c2	0.4843	+0.2682	3/3
Q4←Q8	1.1870	c2	0.4220	+0.7650	3/3
Q4←G12	2.5334	c3	1.2860	+1.2474	2/3
Q8←Q4	0.4453	c2	0.3217	+0.1236	3/3
Q8←Q8	0.3826	c1	0.3063	+0.0763	3/3
Q8←G12	0.5087	c1	0.3152	+0.1935	2/3
G12←Q4	0.8177	c1	0.7045	+0.1132	2/3
G12←Q8	0.8942	c3	0.6142	+0.2800	2/3
G12←G12	0.9772	c1	0.8376	+0.1397	2/3

O.6 Task-family results

The largest gains come from Light tasks. Table 79 reports descriptive means across the nine cells and the number of positive cells for each family.

Table 79: Task-family means across nine cells. Values are descriptive point estimates. Positive counts give the number of cells with a positive effect.

Matrix	Family	ExecGain	Positive	CapsuleGain	Positive
Qwen	Light	+1.7098	9/9	+0.9807	4/9
Qwen	Put	+0.0005	6/9	-0.0047	3/9
Qwen	Clean	+0.0008	8/9	-0.0292	1/9
Qwen	Cool	-0.0104	4/9	-0.0126	4/9
Qwen	Heat	+0.0233	5/9	-0.0646	2/9
Qwen	Two objects	+0.0080	7/9	+0.0034	3/9
Mixed	Light	+1.8079	9/9	+1.0342	5/9
Mixed	Put	-0.0101	5/9	-0.0160	1/9
Mixed	Clean	-0.0072	7/9	-0.0121	3/9
Mixed	Cool	+0.0269	5/9	+0.0309	5/9
Mixed	Heat	+0.0388	6/9	-0.0645	3/9
Mixed	Two objects	+0.0027	5/9	-0.0007	4/9

O.7 From writer skills to personalized capsules

The following examples show how the tutor condenses the parts of a longer skill most relevant to the executor's decisions into directly actionable rules. The selected capsules are reproduced in full.

Qwen3-8B writer to Qwen3-4B-Instruct executor. This Qwen-only cell selects c2, with Selection improvement $+0.7679$ and Final ExecGain $+0.7503$. The original skill includes the following instructions:

> Check receptacle state (open/closed) before interacting to avoid wasted steps. Verify the object's state (e.g., container, location, inventory) before attempting to interact with it. Prioritize inventory management: pick up objects if the task requires carrying or manipulating them.

The complete selected capsule is:

- When the target object is visible at your current location, choose to take it; do not use look or examine or navigate to another location.
- When the task requires using a desklamp to examine an object, choose to take the object or use the desklamp; do not use look.
- When searching for an object, choose to take it from the identified location; do not navigate to the destination before acquiring the object.
- When an object is required for a task, choose to take it first; do not perform look or examine actions on the receptacle.

Gemma-4-12B writer to Qwen3-4B-Instruct executor. This mixed-matrix cell selects c3, with Selection improvement $+1.2474$ and Final ExecGain $+1.1463$. The original skill includes these instructions:

> Always `open` a receptacle before attempting to `take` or `put`. Use `look`, `examine`, or `inventory` frequently to confirm object presence, refresh the world model, and verify the agent is holding the correct item before moving.

The complete selected capsule is:

- Search $\rightarrow$ Retrieve: When an object is visible, prioritize `take [object] from [location]` over generic `look` or `examine` commands.
- Interaction $\rightarrow$ Use: To interact with a device (e.g., desklamp), use the `use` command rather than `look`.
- Receptacle Protocol: Always `open` a container before attempting to `take` or `put` an item.
- Atomic Execution: Issue only one primary action per turn (e.g., `go to` then `open` in separate steps) to maintain synchronization.
- State Verification: Use `look`, `examine`, or `inventory` only to refresh the world model or verify held items, not as a substitute for retrieval.

Together, these experiments support a cross-family tutor and a cross-family writer–executor chain: Gemma-4-31B can turn a writer's skill into a short capsule suited to the specified executor, with positive pooled reference-corrected gains in both matrices.

P SKILLTUTOR WITH A GEMINI TUTOR AND STABILITY VALIDATION

With Gemini-3.8-Flash as tutor, SKILLTUTOR improves held-out action prediction in three complete writer–executor matrices spanning Qwen3.5 and Gemma-4. The deployed policy lowers correct-action NLL by 0.0703, 0.0898 and 0.0357, respectively, and improves next-action Top-1 accuracy in all three matrices. Each result includes all nine cells, including cells that retain the original skill.

P.1 THREE MATRICES AND THE TWO EVALUATION PHASES

Model combinations. The *Qwen scale matrix* uses Qwen3.5-4B, Qwen3.5-9B and Qwen3.5-27B as both writers and executors. The *large mixed matrix* uses Qwen3.5-9B, Gemma-4-12B and Gemma-4-31B. The *confirmation matrix* uses Qwen3.5-4B, Qwen3.5-9B and Gemma-4-12B. Each contains all nine writer–executor pairs. Gemini-3.8-Flash is the tutor, and local Qwen3-32B is both auditor and reference model. Executors also run locally. In the tables, Q4, Q9 and Q27 denote the three Qwen3.5 models, while G12 and G31 denote the Gemma-4 models; E←W identifies an executor and the writer of its skill.

Development and locked evaluation. The initial phase covers the Qwen scale and large mixed matrices with a single candidate-selection stage. Its 60-task Final set is then used as *stability validation* to develop the two-stage selection rule. The second phase adds the confirmation matrix and evaluates all three matrices on a new, non-overlapping 60-task locked Final set. The two-stage rule is frozen before evaluation on this set. The three matrices share tasks and some models.

P.2 CANDIDATE GENERATION, AUDIT AND TWO-STAGE SELECTION

Each writer produces an original skill from 100 frozen trajectories, comprising 43 successes and 57 failures. Calibration exposes the target executor's key-state errors. Given the original skill and this evidence, the tutor generates three short capsules: c1 binds triggers to actions, c2 contrasts appropriate and inappropriate actions, and c3 binds object visibility, inventory, container state and action prerequisites. A capsule is appended to the original skill for use by the frozen executor.

The Qwen3-32B auditor checks whether a candidate introduces information beyond its source skill. Only candidates that pass enter Selection-1. Writing s for the original skill, the first stage chooses

$$c^* = \arg \min_{c \in \mathcal{C}_{\text{audit}}} L_{E,\text{Selection-1}}(s \oplus c),$$

and requires lower task-mean NLL than Base. The second stage evaluates this fixed candidate on stability validation and deploys it only if the lower endpoint of the stratified paired-bootstrap 95% interval for

$$L_{E,\text{stability}}(s) - L_{E,\text{stability}}(s \oplus c^*)$$

is strictly above zero. If no candidate passes the audit, Selection-1 shows no improvement, or the stability interval fails this test, the policy retains Base.

Generic is a deterministic, executor-independent length control: it extracts the first 95 body-text tokens of the original skill and divides them into five items. Shuffled supplies a capsule intended for a different executor. These controls distinguish the deployed repair from a generic short addition and from an incorrectly paired capsule.

P.3 Data isolation and measurements

Table 80: Data protocol for the Gemini Tutor extension. The 100 source trajectories contain 43 successes and 57 failures. The new locked final has 350 environment states, including 168 critical states, and 30 seen plus 30 unseen tasks.

Stage	Count	Critical	Role
Writer source	100 trajectories	—	Generate and freeze the original writer skills
Calibration	60 tasks	168	Diagnose errors and generate candidates
Selection-1	30 tasks	85	Choose one audited candidate per cell
Stability validation	60 tasks	168	Accept the candidate or fall back to Base
Locked final	60 tasks	168	Evaluate once after freezing the rule

Calibration, Selection-1, stability validation and locked Final have mutually disjoint task identifiers. The locked Final set contains 350 environment states and 168 critical operating states. Its 60 tasks comprise 30 seen and 30 unseen tasks, with five tasks from each of six ALFWorld families in each split. The families are Light, Put, Clean, Cool, Heat and Two objects, as defined in Appendix O. Thirty-five states admit multiple environment-validated correct actions.

NLL is measured over the set of correct actions. Critical-state measurements are first averaged within each task and then averaged with equal task weights. Confidence intervals use paired task-bootstrap samples, stratified by family and seen/unseen split. The primary *policy-level* estimates include all nine cells in a matrix, with Base retained wherever repair is rejected; accepted-only estimates are reported separately.

ExecGain is Base NLL minus deployed-policy NLL for the target executor. RefGain is the corresponding reduction for Qwen3-32B on the same texts, and CapsuleGain is ExecGain minus RefGain. The Generic and Shuffled contrasts subtract deployed-policy NLL from the respective control NLL. ΔTop-1 is the change from Base in next-action accuracy, in percentage points. Positive values favor the deployed policy. Measurements use teacher-forced states.

P.4 Writer–Executor Gap and SkillDiag measurements

The writer ranking changes with the executor in all three matrices. An original skill can also perform worse than No Skill. For example, with Qwen3.5-4B as executor, the Q4, Q9 and Q27 skills have SkillGain -0.0993, -0.1806 and -0.2470, respectively, with all three 95% intervals below zero. In contrast, the Q4 skill has positive SkillGain $+0.0653$ for the Qwen3.5-9B executor, with interval $[+0.0366, +0.0946]$. The same writer's skill therefore need not help every executor.

The following tables report SkillGain, WCL and corrected EDM for every cell. SkillGain is No-Skill NLL minus writer-skill NLL. WCL measures the reference model's NLL increase from the source text to the writer skill; corrected EDM subtracts WCL from the executor's corresponding increase. The correlations between locked-final corrected EDM and recovery are $+0.917$, -0.229 and $+0.386$ for the Qwen scale, large mixed and confirmation matrices. Deployment follows the audit and held-out selection tests.

Table 81: Qwen scale matrix: original-skill diagnostics on the locked final. SkillGain compares the writer skill with No Skill; negative values indicate harm. Brackets give the 95% confidence intervals.

E←W	SkillGain	WCL	Corrected EDM
	-0.0993	+0.0767	+0.2098
Q4←Q4	[-0.1593, -0.0402]	[-0.0147, +0.1730]	[+0.0874, +0.3334]
	-0.1806	+0.0085	+0.3593
Q4←Q9	[-0.2899, -0.0742]	[-0.0576, +0.0650]	[+0.2174, +0.5003]
	-0.2470	+0.1972	+0.2370
Q4←Q27	[-0.3833, -0.1036]	[+0.1055, +0.2954]	[+0.1036, +0.3728]
	+0.0653	+0.0767	-0.1641
Q9←Q4	[+0.0366, +0.0946]	[-0.0159, +0.1755]	[-0.3037, -0.0285]
	-0.0117	+0.0085	-0.0188
Q9←Q9	[-0.0521, +0.0316]	[-0.0579, +0.0647]	[-0.1451, +0.1039]
	-0.0765	+0.1972	-0.1428
Q9←Q27	[-0.1940, +0.0380]	[+0.1059, +0.2944]	[-0.2643, -0.0218]
	-0.0038	+0.0767	-0.0979
Q27←Q4	[-0.0382, +0.0273]	[-0.0151, +0.1757]	[-0.2182, +0.0186]
	-0.0342	+0.0085	+0.0007
Q27←Q9	[-0.0634, -0.0052]	[-0.0569, +0.0652]	[-0.0840, +0.0881]
	-0.0301	+0.1972	-0.1921
Q27←Q27	[-0.0913, +0.0301]	[+0.1069, +0.2964]	[-0.2889, -0.1017]

Table 82: Large mixed matrix: original-skill diagnostics on the locked final. SkillGain compares the writer skill with No Skill; negative values indicate harm. Brackets give the 95% confidence intervals.

E←W	SkillGain	WCL	Corrected EDM
	-0.0117	+0.0085	-0.0188
Q9←Q9	[-0.0523, +0.0318]	[-0.0574, +0.0655]	[-0.1440, +0.1037]
	-0.0395	+0.0960	-0.0785
Q9←G12	[-0.0825, +0.0010]	[+0.0186, +0.1875]	[-0.2594, +0.0931]
	-0.0379	+0.0789	-0.0630
Q9←G31	[-0.0710, -0.0056]	[+0.0058, +0.1556]	[-0.2404, +0.1093]
	+0.0531	+0.0085	+0.0309
G12←Q9	[-0.0764, +0.1897]	[-0.0579, +0.0650]	[-0.1629, +0.2385]
	-0.0170	+0.0960	+0.0135
G12←G12	[-0.1736, +0.1521]	[+0.0194, +0.1858]	[-0.1928, +0.2145]
	-0.0307	+0.0789	+0.0442
G12←G31	[-0.2168, +0.1415]	[+0.0053, +0.1566]	[-0.1811, +0.2915]
	-0.1098	+0.0085	-0.2169
G31←Q9	[-0.2123, -0.0145]	[-0.0572, +0.0653]	[-0.5105, +0.0750]
	-0.0821	+0.0960	-0.3322
G31←G12	[-0.1819, +0.0002]	[+0.0190, +0.1877]	[-0.6726, +0.0239]
	-0.0137	+0.0789	-0.3835
G31←G31	[-0.1134, +0.0807]	[+0.0057, +0.1568]	[-0.6884, -0.0724]

Table 83: Cross-family confirmation matrix: original-skill diagnostics on the locked final. SkillGain compares the writer skill with No Skill; negative values indicate harm. Brackets give the 95% confidence intervals.

E←W	SkillGain	WCL	Corrected EDM
Q4←Q4	-0.0993 [-0.1593, -0.0402]	+0.0767 [-0.0147, +0.1730]	+0.2098 [+0.0874, +0.3334]
Q4←Q9	-0.1806 [-0.2899, -0.0742]	+0.0085 [-0.0576, +0.0650]	+0.3593 [+0.2174, +0.5003]
Q4←G12	-0.0566 [-0.1620, +0.0407]	+0.0960 [+0.0191, +0.1863]	+0.1478 [-0.0320, +0.3231]
Q9←Q4	+0.0653 [+0.0366, +0.0946]	+0.0767 [-0.0159, +0.1755]	-0.1641 [-0.3037, -0.0285]
Q9←Q9	-0.0117 [-0.0521, +0.0316]	+0.0085 [-0.0579, +0.0647]	-0.0188 [-0.1451, +0.1039]
Q9←G12	-0.0395 [-0.0823, +0.0006]	+0.0960 [+0.0186, +0.1873]	-0.0785 [-0.2578, +0.0908]
G12←Q4	+0.0938 [-0.0398, +0.2405]	+0.0767 [-0.0151, +0.1757]	-0.0781 [-0.2695, +0.1245]
G12←Q9	+0.0531 [-0.0785, +0.1919]	+0.0085 [-0.0569, +0.0652]	+0.0309 [-0.1631, +0.2419]
G12←G12	-0.0170 [-0.1736, +0.1505]	+0.0960 [+0.0192, +0.1882]	+0.0135 [-0.1958, +0.2153]

P.5 LOCKED-FINAL POLICY RESULTS

Table 84: Locked-final policy results across all nine cells in each matrix, including two Base fallbacks. Brackets give the 95% confidence intervals; positive differences favor Personal. Top-1 changes are percentage points.

Metric	Qwen	Large mixed	Confirmation
ExecGain	+0.0703 [+0.0540, +0.0879]	+0.0898 [+0.0371, +0.1553]	+0.0357 [+0.0108, +0.0633]
CapsuleGain	+0.0252 [-0.0091, +0.0586]	+0.0541 [-0.0090, +0.1229]	+0.0073 [-0.0240, +0.0396]
vs Generic	+0.0703 [+0.0506, +0.0921]	+0.0913 [+0.0530, +0.1383]	+0.0387 [+0.0207, +0.0560]
vs Shuffled	+0.0185 [+0.0081, +0.0298]	+0.0329 [+0.0074, +0.0610]	-0.0075 [-0.0328, +0.0198]
ΔTop-1 (pp)	+1.4969 [+1.0031, +1.9907]	+1.8056 [+0.7099, +2.9167]	+1.0802 [+0.4475, +1.7438]

All three matrices have positive policy-level ExecGain with 95% intervals above zero: the Qwen scale matrix yields $+0.0703$ $[+0.0540, +0.0879]$, the large mixed matrix yields $+0.0898$ $[+0.0371, +0.1553]$, and the confirmation matrix yields $+0.0357$ $[+0.0108, +0.0633]$. All three also outperform Generic and improve Top-1 accuracy significantly. The Top-1 gains are 1.4969, 1.8056 and 1.0802 percentage points, respectively. Thus, the improvements appear in both correct-action likelihood and next-action ranking.

CapsuleGain has positive pooled point estimates of $+0.0252$, $+0.0541$ and $+0.0073$, but all three 95% intervals include zero. The first two matrices significantly outperform Shuffled; the confirmation matrix's Shuffled contrast is -0.0075 with interval $[-0.0328, +0.0198]$.

Each matrix deploys seven capsules and retains Base in two cells. Table 85 gives the accepted-only results for the seven deployed cells, whose pooled ExecGain is $+0.0904$, $+0.1155$ and $+0.0459$, respectively.

Table 85: Locked-final results restricted to the seven deployed capsules in each matrix. Brackets give the 95% confidence intervals; positive differences favor Personal. Top-1 changes are percentage points.

Metric	Qwen	Large mixed	Confirmation
ExecGain	+0.0904 [+0.0693, +0.1127]	+0.1155 [+0.0470, +0.1991]	+0.0459 [+0.0133, +0.0805]
CapsuleGain	+0.0324 [-0.0121, +0.0762]	+0.0696 [-0.0119, +0.1573]	+0.0094 [-0.0305, +0.0514]
vs Generic	+0.0897 [+0.0654, +0.1177]	+0.1149 [+0.0616, +0.1829]	+0.0471 [+0.0232, +0.0715]
vs Shuffled	+0.0357 [+0.0244, +0.0485]	+0.0560 [+0.0223, +0.0944]	+0.0143 [-0.0095, +0.0435]
ΔTop-1 (pp)	+1.9246 [+1.2897, +2.5595]	+2.3214 [+0.9127, +3.7698]	+1.3889 [+0.5754, +2.2421]

P.6 Complete writer–executor results

Among deployed capsules, the direction of Final ExecGain is positive in $7/7$ cells in the Qwen scale matrix, $7/7$ in the large mixed matrix and $6/7$ in the confirmation matrix. The following tables give all cells, their deployment decisions, NLLs, gains, confidence intervals and control comparisons.

Table 86: Qwen scale matrix: all nine locked-final cells. Cell notation is executor ← writer. A dagger marks Base fallback; Personal then equals Base. Brackets give the 95% confidence intervals.

E←W	Base	Personal	ExecGain	CapsuleGain
Q4←Q4	0.6309	0.4784	+0.1525 [+0.1100, +0.1943]	+0.0878 [+0.0294, +0.1431]
Q4←Q9	0.7123	0.5902	+0.1221 [+0.0945, +0.1526]	+0.0830 [+0.0330, +0.1301]
Q4←Q27	0.7787	0.5916	+0.1870 [+0.1290, +0.2473]	+0.1123 [+0.0272, +0.2006]
Q9←Q4	0.2340	0.1948	+0.0392 [+0.0060, +0.0771]	-0.0192 [-0.1072, +0.0573]
Q9←Q9†	0.3111	0.3111	+0.0000 [+0.0000, +0.0000]	+0.0000 [+0.0000, +0.0000]
Q9←Q27	0.3758	0.3455	+0.0303 [+0.0028, +0.0582]	-0.0306 [-0.0945, +0.0308]
Q27←Q4	0.1764	0.1456	+0.0308 [+0.0106, +0.0523]	-0.0057 [-0.0656, +0.0468]
Q27←Q9†	0.2068	0.2068	+0.0000 [+0.0000, +0.0000]	+0.0000 [+0.0000, +0.0000]
Q27←Q27	0.2027	0.1321	+0.0706 [+0.0303, +0.1319]	-0.0011 [-0.0548, +0.0475]

Table 87: Qwen scale matrix: Generic and Shuffled controls for every locked-final cell. Brackets give the 95% confidence intervals. Policy indicates deployment of a capsule or fallback to Base.

E←W	Policy	vs Generic	vs Shuffled
Q4←Q4	Deploy	+0.1293 [+0.0817, +0.1750]	+0.0145 [-0.0246, +0.0544]
Q4←Q9	Deploy	+0.1194 [+0.0897, +0.1487]	+0.1221 [+0.0937, +0.1522]
Q4←Q27	Deploy	+0.1474 [+0.0970, +0.2091]	+0.1375 [+0.0936, +0.1831]
Q9←Q4	Deploy	+0.0399 [+0.0070, +0.0769]	-0.0010 [-0.0240, +0.0247]
Q9←Q9	Base	-0.0033 [-0.0189, +0.0120]	+0.0000 [+0.0000, +0.0000]
Q9←Q27	Deploy	+0.0539 [+0.0203, +0.0913]	-0.0072 [-0.0310, +0.0181]
Q27←Q4	Deploy	+0.0382 [+0.0188, +0.0594]	-0.0316 [-0.0541, -0.0143]
Q27←Q9	Base	+0.0085 [-0.0040, +0.0215]	-0.0839 [-0.1272, -0.0475]
Q27←Q27	Deploy	+0.0997 [+0.0509, +0.1762]	+0.0159 [-0.0064, +0.0456]

Table 88: Large mixed matrix: all nine locked-final cells. Cell notation is executor ← writer. A dagger marks Base fallback; Personal then equals Base. Brackets give the 95% confidence intervals.

E←W	Base	Personal	ExecGain	CapsuleGain
Q9←Q9	0.3111	0.2946	+0.0165 [-0.0032, +0.0358]	+0.0272 [-0.0113, +0.0678]
Q9←G12	0.3389	0.2718	+0.0671 [+0.0188, +0.1184]	+0.0267 [-0.0376, +0.0918]
Q9←G31	0.3373	0.2593	+0.0779 [+0.0347, +0.1277]	+0.0022 [-0.0704, +0.0782]
G12←Q9	0.5274	0.5136	+0.0138 [-0.1133, +0.1250]	+0.0084 [-0.1168, +0.1216]
G12←G12	0.5975	0.3888	+0.2087 [+0.0358, +0.3997]	+0.0917 [-0.1053, +0.2970]
G12←G31	0.6111	0.4333	+0.1778 [+0.0905, +0.2704]	+0.1228 [+0.0179, +0.2307]
G31←Q9†	0.4800	0.4800	+0.0000 [+0.0000, +0.0000]	+0.0000 [+0.0000, +0.0000]
G31←G12†	0.4523	0.4523	+0.0000 [+0.0000, +0.0000]	+0.0000 [+0.0000, +0.0000]
G31←G31	0.3838	0.1371	+0.2467 [-0.0172, +0.6173]	+0.2083 [-0.0636, +0.5664]

Table 89: Large mixed matrix: Generic and Shuffled controls for every locked-final cell. Brackets give the 95% confidence intervals. Policy indicates deployment of a capsule or fallback to Base.

E←W	Policy	vs Generic	vs Shuffled
Q9←Q9	Deploy	+0.0132 [-0.0090, +0.0331]	+0.0004 [-0.0100, +0.0115]
Q9←G12	Deploy	+0.0773 [+0.0323, +0.1269]	-0.0150 [-0.0389, +0.0062]
Q9←G31	Deploy	+0.0686 [+0.0296, +0.1125]	+0.0249 [-0.0052, +0.0526]
G12←Q9	Deploy	+0.0238 [-0.0873, +0.1309]	-0.0019 [-0.0679, +0.0606]
G12←G12	Deploy	+0.2258 [+0.0809, +0.3851]	+0.1977 [+0.0566, +0.3470]
G12←G31	Deploy	+0.1647 [+0.0463, +0.2799]	-0.0609 [-0.2553, +0.1017]
G31←Q9	Base	+0.0020 [-0.0418, +0.0537]	-0.0182 [-0.0475, +0.0053]
G31←G12	Base	+0.0160 [-0.0532, +0.0877]	-0.0777 [-0.1533, -0.0191]
G31←G31	Deploy	+0.2309 [+0.0015, +0.5614]	+0.2468 [-0.0303, +0.6079]

Table 90: Cross-family confirmation matrix: all nine locked-final cells. Cell notation is executor ← writer. A dagger marks Base fallback; Personal then equals Base. Brackets give the 95% confidence intervals.

E←W	Base	Personal	ExecGain	CapsuleGain
Q4←Q4	0.6309	0.5864	+0.0445 [+0.0081, +0.0788]	-0.0165 [-0.0726, +0.0344]
Q4←Q9	0.7123	0.6660	+0.0463 [+0.0259, +0.0692]	+0.0402 [+0.0066, +0.0754]
Q4←G12†	0.5883	0.5883	+0.0000 [+0.0000, +0.0000]	+0.0000 [+0.0000, +0.0000]
Q9←Q4	0.2340	0.2260	+0.0080 [-0.0052, +0.0205]	-0.0488 [-0.1375, +0.0267]
Q9←Q9	0.3111	0.3022	+0.0089 [-0.0116, +0.0291]	-0.0001 [-0.0302, +0.0294]
Q9←G12	0.3389	0.3440	-0.0051 [-0.0328, +0.0158]	-0.0519 [-0.1098, -0.0005]
G12←Q4†	0.4867	0.4867	+0.0000 [+0.0000, +0.0000]	+0.0000 [+0.0000, +0.0000]
G12←Q9	0.5274	0.4627	+0.0647 [-0.0863, +0.2236]	+0.0764 [-0.0766, +0.2412]
G12←G12	0.5975	0.4437	+0.1538 [+0.0535, +0.2623]	+0.0668 [-0.0512, +0.1850]

Table 91: Cross-family confirmation matrix: Generic and Shuffled controls for every locked-final cell. Brackets give the 95% confidence intervals. Policy indicates deployment of a capsule or fallback to Base.

E←W	Policy	vs Generic	vs Shuffled
Q4←Q4	Deploy	+0.0213 [-0.0108, +0.0496]	-0.0030 [-0.0460, +0.0388]
Q4←Q9	Deploy	+0.0436 [+0.0123, +0.0736]	-0.0142 [-0.0325, +0.0048]
Q4←G12	Base	+0.0073 [-0.0131, +0.0287]	-0.1042 [-0.1482, -0.0605]
Q9←Q4	Deploy	+0.0087 [-0.0055, +0.0231]	-0.0044 [-0.0207, +0.0119]
Q9←Q9	Deploy	+0.0056 [-0.0212, +0.0284]	-0.0160 [-0.0335, +0.0026]
Q9←G12	Deploy	+0.0051 [-0.0220, +0.0275]	-0.0597 [-0.1160, -0.0055]
G12←Q4	Base	+0.0108 [-0.0577, +0.0768]	-0.0636 [-0.1922, +0.0667]
G12←Q9	Deploy	+0.0747 [-0.0079, +0.1631]	+0.0804 [-0.0563, +0.2281]
G12←G12	Deploy	+0.1709 [+0.0632, +0.2898]	+0.1173 [+0.0265, +0.2197]

P.7 Audit, selection and fallback decisions

The auditor passes 15 of 27 candidates in the Qwen scale matrix, 13 of 27 in the large mixed matrix and 15 of 27 in the confirmation matrix. All three matrices ultimately deploy seven capsules. In the Qwen scale matrix, two cells have no audited candidate and retain Base. In the large mixed matrix, every cell has an eligible candidate, but stability validation rejects two Gemma-4-31B executor cells. With Q9 and G12 writers, these candidates improve Selection-1 NLL by $+0.1815$ and $+0.0794$, yet their stability gains are -0.0435 $[-0.1131, +0.0236]$ and -0.0413 $[-0.0991, +0.0119]$. Both therefore retain Base on locked Final. In the confirmation matrix, the Q4 executor with a G12 writer has no eligible candidate, while the G12 executor with a Q4 writer fails the stability test.

Table 92: Qwen scale matrix: the two-stage selection record, frozen before the new locked final. The auditor accepts 15/27 candidates. Deploy requires a positive Selection-1 gain and a strictly positive stability-validation 95% lower bound.

E←W	Style	Selection-1 ΔNLL	Stability gain and 95% CI	Policy
Q4←Q4	c2	+0.1794	+0.0980 [+0.0641, +0.1331]	Deploy
Q4←Q9	c2	+0.1513	+0.1582 [+0.1132, +0.2016]	Deploy
Q4←Q27	c3	+0.2568	+0.1503 [+0.0987, +0.2066]	Deploy
Q9←Q4	c2	+0.0975	+0.0852 [+0.0533, +0.1181]	Deploy
Q9←Q9	—	+0.0000	+0.0000 [+0.0000, +0.0000]	Base
Q9←Q27	c3	+0.1007	+0.0675 [+0.0386, +0.0986]	Deploy
Q27←Q4	c2	+0.0747	+0.0336 [+0.0176, +0.0499]	Deploy
Q27←Q9	—	+0.0000	+0.0000 [+0.0000, +0.0000]	Base
Q27←Q27	c3	+0.2193	+0.1149 [+0.0457, +0.1888]	Deploy

Table 93: Large mixed matrix: the two-stage selection record, frozen before the new locked final. The auditor accepts 13/27 candidates. Deploy requires a positive Selection-1 gain and a strictly positive stability-validation 95% lower bound.

E←W	Style	Selection-1 ΔNLL	Stability gain and 95% CI	Policy
Q9←Q9	c1	+0.0601	+0.0305 [+0.0139, +0.0472]	Deploy
Q9←G12	c2	+0.0970	+0.0823 [+0.0467, +0.1242]	Deploy
Q9←G31	c3	+0.0737	+0.1113 [+0.0985, +0.1258]	Deploy
G12←Q9	c3	+0.0311	+0.1012 [+0.0065, +0.2028]	Deploy
G12←G12	c3	+0.4998	+0.3254 [+0.1720, +0.4950]	Deploy
G12←G31	c3	+0.1050	+0.1132 [+0.0493, +0.1775]	Deploy
G31←Q9	c1	+0.1815	-0.0435 [-0.1131, +0.0236]	Base
G31←G12	c3	+0.0794	-0.0413 [-0.0991, +0.0119]	Base
G31←G31	c2	+0.9277	+0.4273 [+0.1620, +0.7025]	Deploy

Table 94: Cross-family confirmation matrix: the two-stage selection record, frozen before the new locked final. The auditor accepts 15/27 candidates. Deploy requires a positive Selection-1 gain and a strictly positive stability-validation 95% lower bound.

E←W	Style	Selection-1 ΔNLL	Stability gain and 95% CI	Policy
Q4←Q4	c3	+0.0734	+0.0476 [+0.0170, +0.0764]	Deploy
Q4←Q9	c2	+0.0735	+0.0688 [+0.0296, +0.1049]	Deploy
Q4←G12	—	+0.0000	+0.0000 [+0.0000, +0.0000]	Base
Q9←Q4	c1	+0.0502	+0.0260 [+0.0104, +0.0401]	Deploy
Q9←Q9	c3	+0.0579	+0.0300 [+0.0156, +0.0429]	Deploy
Q9←G12	c3	+0.0381	+0.0400 [+0.0197, +0.0600]	Deploy
G12←Q4	c1	+0.0085	+0.0460 [-0.0876, +0.1709]	Base
G12←Q9	c3	+0.0644	+0.2287 [+0.1148, +0.3471]	Deploy
G12←G12	c1	+0.2859	+0.1234 [+0.0284, +0.2174]	Deploy

Source-faithfulness audit. In the initial phase, a candidate for the G12 writer–executor pair introduced a lamp/`use` rule not stated in its source skill and nevertheless passed the auditor. A similar unsupported lamp procedure was rejected for the Q9 writer–executor pair in the Qwen matrix.

P.8 TASK-FAMILY RESULTS

Table 95: Task-family means over the seven deployed capsules in each matrix. Positive counts give the number of deployed cells with a positive effect. Values are descriptive point estimates.

Matrix	Family	ExecGain	Positive	CapsuleGain	Positive
Qwen	Light	+0.4382	7/7	+0.2130	5/7
Qwen	Put	+0.0143	4/7	-0.0323	3/7
Qwen	Clean	+0.0752	6/7	+0.0643	5/7
Qwen	Cool	-0.0084	2/7	-0.0173	3/7
Qwen	Heat	+0.0156	6/7	+0.0049	3/7
Qwen	Two objects	+0.0072	5/7	-0.0384	2/7
Large mixed	Light	+0.6044	7/7	+0.4500	7/7
Large mixed	Put	+0.0044	7/7	-0.0534	2/7
Large mixed	Clean	+0.0801	5/7	+0.0850	6/7
Large mixed	Cool	-0.0273	2/7	-0.0932	3/7
Large mixed	Heat	+0.0384	6/7	+0.0573	6/7
Large mixed	Two objects	-0.0070	3/7	-0.0280	2/7
Confirm.	Light	+0.2420	6/7	+0.1406	5/7
Confirm.	Put	+0.0059	5/7	-0.0432	1/7
Confirm.	Clean	+0.0599	5/7	+0.0595	4/7
Confirm.	Cool	-0.0348	2/7	-0.0962	1/7
Confirm.	Heat	-0.0011	4/7	+0.0317	4/7
Confirm.	Two objects	+0.0032	5/7	-0.0359	3/7

The strongest gains arise in Light tasks. Averaged over the seven deployed cells, Light ExecGain is $+0.4382$, $+0.6044$ and $+0.2420$ in the three matrices, respectively. Clean tasks are another positive contributor, with means $+0.0752$, $+0.0801$ and $+0.0599$. Other families show smaller and sometimes negative effects: for example, the Cool means are -0.0084, -0.0273 and -0.0348. Table 95 gives these descriptive family means and positive-cell counts over deployed capsules.

P.9 THREE REPAIR AND SELECTION EXAMPLES

A deployed repair: Qwen3.5-9B writer to Qwen3.5-4B executor. In the Qwen scale matrix, Selection-1 chooses c2, the contrastive capsule. The writer skill states:

> Confirm the target object is visible in the current view before executing `take`.
> Ensure the object has reached the required state (e.g., cooled) before moving it.

The selected capsule turns the interaction decisions into short action contrasts. Its full text is:

- When the target object is visible in the current view, choose `take`; do not issue `look` or `examine`.
- When holding a dirty object at a sinkbasin, choose `clean`; do not attempt `move` before completing required state cleaning.
- When holding an object near a lamp to inspect it, choose `use`; do not issue redundant `examine` commands.
- When preparing to cool an object, choose `move` to place it in the fridge; do not attempt `cool`.
- When an object must be transported, check visible items and execute `take` first; do not navigate away before acquiring it.

On locked Final, this cell has ExecGain $+0.1221$ with interval $[+0.0945, +0.1526]$.

A rejected repair: Qwen3.5-9B writer to Gemma-4-31B executor. In the large mixed matrix, c1 improves Selection-1 NLL by $+0.1815$. Its stability gain is -0.0435 with interval

$[-0.1131, +0.0236]$, so it fails the positive-lower-bound rule and the policy retains Base. The capsule begins:

- Confirm the target object is visible in the current view and acquire it before leaving to search other storage locations.
- If the target object appears in the current view, execute take directly rather than repeating look or examine commands unnecessarily.
- Always execute take object from a valid location before attempting to place or put the object at any destination receptacle.

A deployed cell with negative Final direction: Gemma-4-12B writer to Qwen3.5-9B executor. In the confirmation matrix, c3 improves Selection-1 NLL by $+0.0381$ and has stability gain $+0.0400$ $[+0.0197, +0.0600]$, passing both stages. Its locked-final ExecGain is -0.0051 with interval $[-0.0328, +0.0158]$. This is the only negative point estimate among the 21 deployed cells, and its interval includes zero. The capsule begins:

- Observation shows target object, inventory empty: execute retrieve immediately by selecting an admissible take command rather than navigating away.
- Target object observed in container or furniture: take the target item directly instead of repeating unproductive examine or goto actions.
- Target object held, destination reached for transformation: execute the process stage from admissible commands before executing any move sequence.

Together, these experiments extend the Writer–Executor Gap observations to the three model matrices and show that SKILLTUTOR with a Gemini tutor improves the deployed policy's action prediction on a fresh, isolated Final set.