SOCRATIC SKILL LEARNING: INTERLEAVING TEACHING AND PRACTICE FOR TRANSFERABLE AGENT SKILLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent Skills improve LLM agents by making task knowledge reusable across sessions. Learning transferable Skills requires evidence about a procedure and its conditions of use, which a successful task attempt can leave untested. We introduce Socratic Skill Learning (SSL), which interleaves expert teaching with Student practice. An Educator guides investigation while the Student remains the sole environment operator and Skill author, able to question, test, and adapt advice. To test reuse beyond the learning task, we construct SKILLTRANSFERBENCH, with 35 task families and 408 executable cases, pairing public variation specifications with held-out evaluation in fresh solver sessions. On the main 22-family, 155-case cohort, Fixed-Window SSL reaches 0.801 utility, versus 0.723 without a Skill, 0.761 for adapted Trace2Skill (trajectory distillation), and 0.751 for adapted SKILL-KD (contrastive distillation). Across three model pairs on shared families, interleaved guidance outperforms expert authoring and post-hoc teaching. Process analysis links guidance to Student writing and package revision; a case study shows the Student testing and refining a proposed procedure for later reuse. As demand for new lessons falls, the cost of fixed-window review grows. Issue-Guided SSL tracks unresolved questions across Skill revisions to select and focus review, reaching 0.805 utility with 61.7% fewer Educator tokens and 16.9% fewer total acquisition tokens than Fixed-Window SSL on the main cohort.

1 INTRODUCTION

Agent Skills extend LLM agents through reusable instructions and executable resources, without updating model weights (Anthropic, 2025). Curated Skills improve success on tasks requiring specialized expertise (Li et al., 2026b). Acquiring transferable Skills requires identifying which steps to reuse, under what conditions, and when to adapt them.

Recent methods distill execution trajectories into Skills and refine packages through testing and feedback (Ni et al., 2026; Ma et al., 2026; Shi et al., 2026). SkillLearnBench's Teacher Feedback uses expert advice after failed attempts to guide revision (Zhong et al., 2026). Such refinement depends on which hypotheses the learner tests and which failures it investigates. Guidance during an ongoing attempt can redirect these choices before an untested assumption becomes a reusable rule. We therefore ask: *how should external expertise guide the learner while it is still gathering evidence?*

We introduce **Socratic Skill Learning (SSL)**, interleaving expert teaching with Student practice (Figure 3).[1] An Educator observes recent Student work and the live Skill, then explains, challenges, diagnoses, or remains silent. The Student operates the environment and authors the package. Inspired by Socratic inquiry, the interaction examines assumptions through dialogue and practical checks; the Student can question advice, adapt it, and decide what to retain. We instantiate SSL with FIXED-WINDOW SSL, which reviews at regular tool-call intervals, and ISSUE-GUIDED SSL, which focuses review on unresolved learning issues.

Existing benchmarks measure Skill utility and reuse (Li et al., 2026b; Zhong et al., 2026). Studying guided acquisition also requires specifying what the learner should investigate beyond the practice

[1] Code, benchmark, and trajectories: anonymous repository.

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

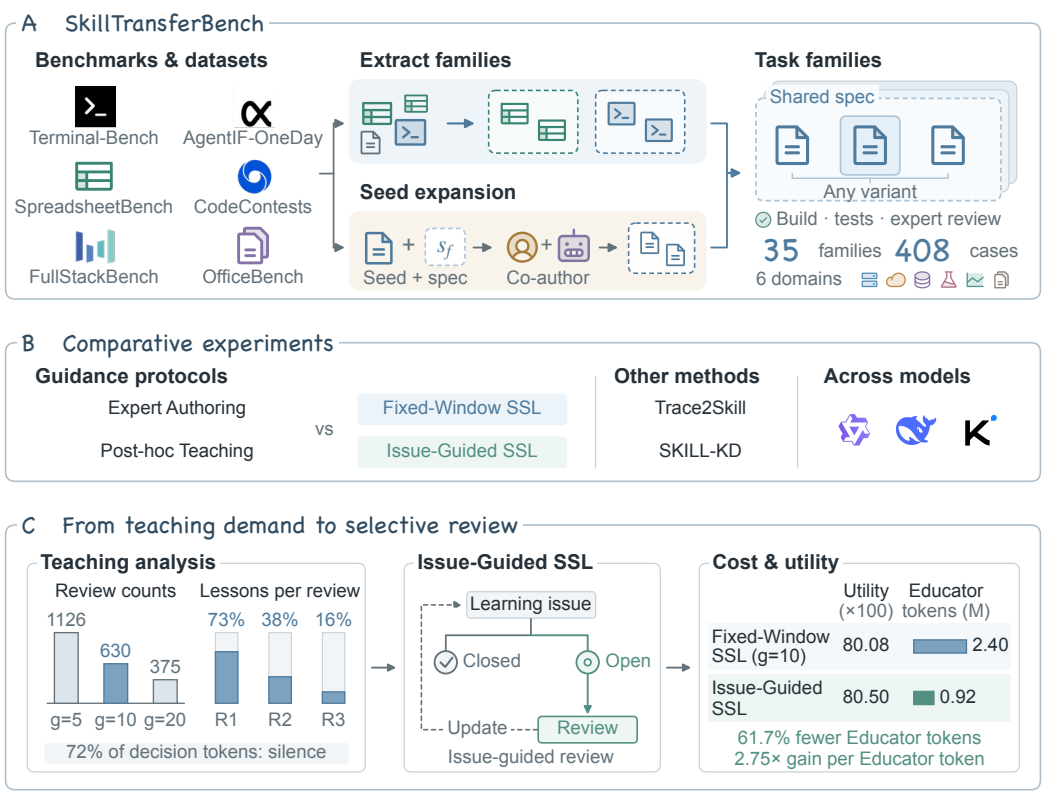


Figure 1: **Overview of the study.** Panels show (A) task-family construction from six source benchmarks, (B) acquisition-protocol comparisons, external methods, and model pairs, and (C) the path from teaching demand to issue-guided review with utility and cost. Panel C summarizes Qwen experiments on the 22-family main cohort.

task. We construct SKILLTRANSFERBENCH, a benchmark of 35 **task families** and 408 executable cases across six domains. Each family pairs an anchor task with a public specification of invariants and permitted variation. This specification guides exploration; fresh solvers test the acquired package on held-out family members.

We organize the study around four questions (Figure 1): **RQ1:** How well does SSL transfer relative to existing induction methods? **RQ2:** How should external expertise guide Skill acquisition? **RQ3:** How does guidance enter Skill development as learning progresses? **RQ4:** How should review be organized to sustain transfer with less expert supervision?

On shared families across three model pairs, both interleaved protocols outperform direct expert authoring and post-hoc teaching. Process analysis links lessons to Student development; a case study shows the Student testing an Educator-proposed procedure and refining it for later reuse. ISSUE-GUIDED SSL reaches 0.805 utility with 61.7% fewer Educator tokens on the main cohort.

Our contributions are threefold:

- We construct SKILLTRANSFERBENCH, containing 35 task families and 408 executable cases across six domains, with public family specifications that guide acquisition and held-out evaluation that tests frozen Skills in fresh solver sessions.
- We introduce Socratic Skill Learning (SSL), in which the Student owns exploration, verification, and Skill authoring while an Educator guides practice. Issue-Guided SSL uses persistent learning issues to select and focus review around unresolved questions.
- We evaluate Skill transfer across three model pairs and complement these comparisons with process analysis and case studies, linking guidance to Student development and showing how selecting and focusing review sustains transfer with less expert supervision.

2 Related Work

Constructing Skills from execution evidence. Skills can retain experience as guidance or executable code (Zhao et al., 2024; Wang et al., 2023), consolidate trajectory lessons (Ni et al., 2026), or evolve through execution and verification (Ma et al., 2026; Liu et al., 2026b). SKILL-KD's consolidation agent derives patches from Teacher–Student trajectory contrasts and validates them through Student reruns (Shi et al., 2026).

Feedback during Skill development. Self-Refine and Reflexion improve outputs or attempts through feedback (Madaan et al., 2023; Shinn et al., 2023). For agent diagnosis (Zhao et al., 2026), CodeTracer reconstructs state transitions to localize failures (Li et al., 2026a). SkillLearnBench uses a reference Skill to guide revision after failure (Zhong et al., 2026); AgentBrew validates teacher-authored corrections through student retries (Jiang & Huang, 2026). HASP includes teacher review of student-proposed skill programs (Liu et al., 2026a), while SkillHone preserves revision evidence and decisions (Li & Hu, 2026). SSL places Educator guidance within Student investigation and verification, leaving the Student to author the Skill. Unresolved learning questions select and focus review. This design connects Socratic dialogue (Liu et al., 2024a) with learner construction (Chi et al., 2001) and feedback on the learning process (Hattie & Timperley, 2007).

Scope and transfer in Skill evaluation. SkillsBench measures Skill utility (Li et al., 2026b); SkillLearnBench and EconSkills evaluate reuse across parameterized instances (Zhong et al., 2026; Quan & Liu, 2026). Trace2Skill tests held-out transfer (Ni et al., 2026), while ContinualSkillBench uses ordered task streams (Guan et al., 2026). SkillTransferBench defines acquisition scope through a public family specification. Across six domains, evaluation in fresh sessions connects guided acquisition to held-out transfer and supervision cost.

3 SkillTransferBench

3.1 Task families and Skill transfer

Skill acquisition develops a reusable procedure for a task family f. Acquisition uses an executable anchor $x_{f,0}$ for practice and a public specification s_f defining invariant goals and permitted variation (e.g., WAL encoding, integrity, and recovery depth). The specification directs investigation beyond the anchor; held-out family members test transfer across these variations.

The acquired Skill is a package $\mathcal{K} = (M, \mathcal{R})$, containing instructions $M =$ `SKILL.md` and optional scripts, references, and assets $\mathcal{R}$. We evaluate the frozen package with the target solver π_S in a new session for each case (a fresh solver). Transfer utility is

$$U_f(\mathcal{K}; \pi_S) = \mathbb{E}_{x \sim \mathcal{D}_f}\left[r_f(\pi_S(x, \mathcal{K}))\right], \tag{1}$$

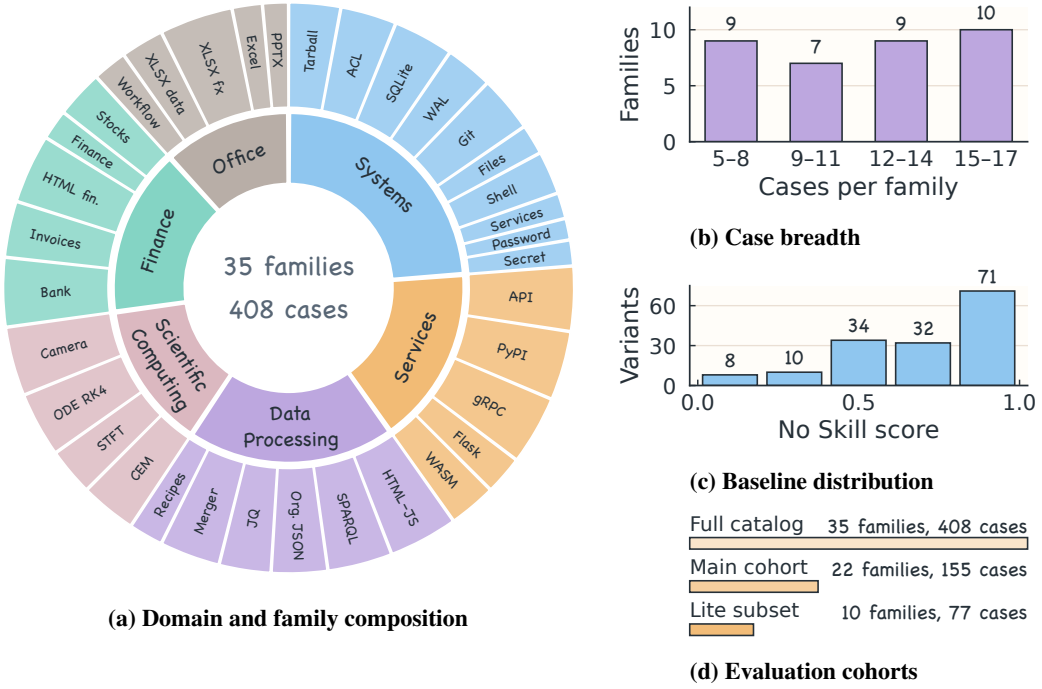
where $\mathcal{D}_f$ is the case distribution and r_f the verifier score.

3.2 Benchmark construction

We construct the benchmark in three stages: adapt source tasks, build task families through two complementary routes, and validate the resulting cases.

Source adaptation. We select tasks with clear requirements and meaningful variation from six benchmarks: Terminal-Bench and AgentIF-OneDay (Merrill et al., 2026; Chen et al., 2026), SpreadsheetBench (Ma et al., 2024), CodeContests and FullStackBench (Li et al., 2022; Liu et al., 2024b), and OfficeBench (Wang et al., 2024). We normalize their formats and align descriptions, fixtures, and scorers.

Task-family construction. Experts in coding and agent benchmarking collaborate with agents through two routes (Figure 1A): **extraction and adaptation** groups compatible source cases under a shared specification and aligns their statements and tests; **seed expansion** starts from one task, defines its invariant goal and permitted variation, and authors additional cases within that scope. Both routes yield a public family specification and executable variants. One fixed case serves as the acquisition anchor; the remaining cases provide candidates for held-out evaluation.



(a) Domain and family composition

(b) Case breadth

(c) Baseline distribution

(d) Evaluation cohorts

Figure 2: **Task-family catalog and evaluation cohorts.** (a) Domain and family composition, (b) family sizes, and (c) Qwen No-Skill Avg@3 scores on the 155 main-cohort cases. (d) Bar length is proportional to case counts in the full catalog, main cohort, and lite subset shared by the additional model pairs.

Validation. Each case includes a resettable environment, a reference solution, and private tests. Programmatic checks verify Docker builds, execute tests, and confirm that reference solutions pass them; an expert other than the author then reviews each case before acceptance. Reference solutions and private tests remain outside agent workspaces (Appendix B).

Catalog and evaluation cohorts. The **full catalog** contains 35 families and 408 cases across six domains, with 5–17 cases per family, including the anchor (Figure 2). The **main cohort** selects families with at least 5 verified held-out cases and at least two non-perfect outcomes in catalog-wide No-Skill screening, yielding 22 families and 155 held-out cases (Table 5); its **lite subset** covers 10 families and 77 cases across all six domains for comparisons across model pairs. Each pair independently acquires its own packages.

3.3 Evaluation and Metrics

Private checks score each fresh solver's submitted artifacts or state. $R_{f,v,r}$ is the fraction passed for case v of family f in repeat r; recorded timeouts score zero. All methods share the cases and verifier. The main study uses 3 independent acquisition runs per family and condition, each evaluated on all held-out cases; No Skill uses 3 fresh-solver repeats. Family-macro Avg@3 is

$$\widehat{U} = \frac{1}{3|\mathcal{F}|} \sum_{r=1}^{3} \sum_{f \in \mathcal{F}} \frac{1}{|V_f|} \sum_{v \in V_f} R_{f,v,r}. \tag{2}$$

Here $\mathcal{F}$ is the family set and V_f its held-out cases. Families receive equal weight; additional model pairs use a single repeat. **Additional metrics:** domain scores average their families; transfer gain is $100(\widehat{U}_{\text{Skill}} - \widehat{U}_{\text{No Skill}})$ percentage points; Qwen main-cohort full-pass rate replaces $R_{f,v,r}$ with $\mathbf{1}[R_{f,v,r} = 1]$ in the same average. **Uncertainty:** paired 95% bootstrap intervals resample families after averaging their cases and repeats; process analyses likewise resample whole families to retain within-family dependence.

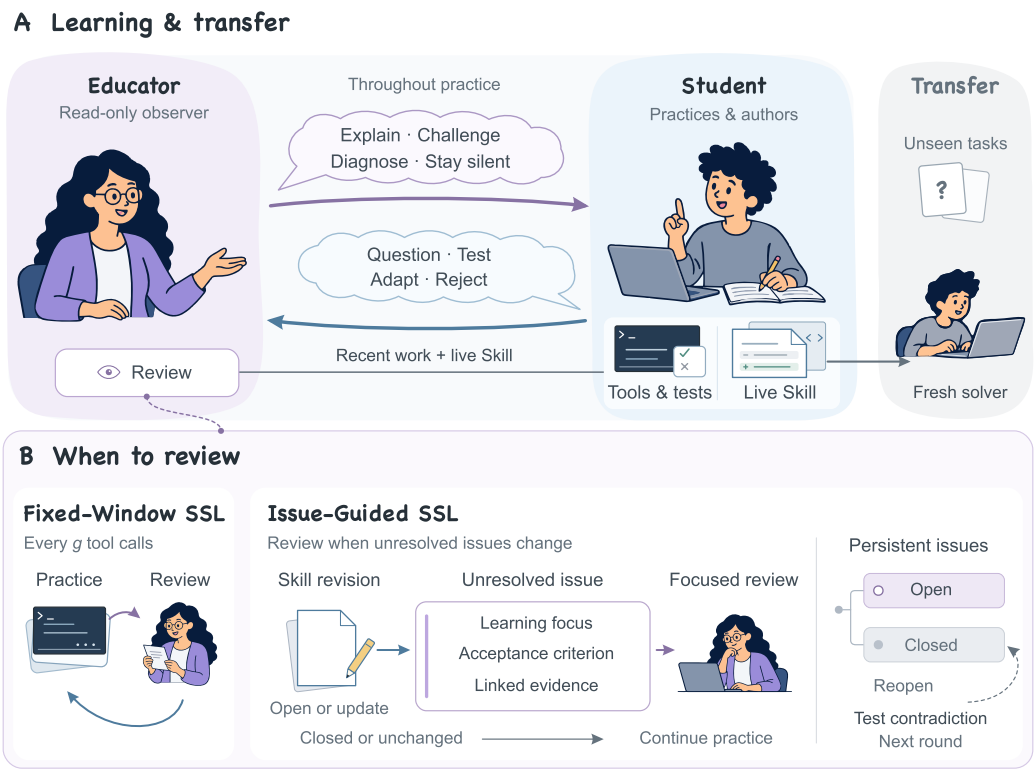


Figure 3: **Socratic Skill Learning. (A)** The Student tests guidance and authors the Skill; the Educator reviews recent work and the live package read-only. Only the Skill transfers to fresh solvers. **(B)** FIXED-WINDOW SSL reviews every g tool calls; ISSUE-GUIDED SSL reviews unresolved issues when they open, change, or reopen, using focused context.

4 SOCRATIC SKILL LEARNING

We introduce Socratic Skill Learning (SSL), in which an Educator guides ongoing investigation while a Student tests the guidance and authors a transferable Skill. We develop two instantiations: FIXED-WINDOW SSL periodically reviews Student progress, while ISSUE-GUIDED SSL maintains learning issues to select and focus subsequent reviews. We first describe their shared learning process (Figure 3A), then present each protocol (Figure 3B).

4.1 SOCRATIC DIALOGUE AND STUDENT PRACTICE

Both protocols use the target solver's model as the Student, the sole environment operator and author of $\mathcal{K}$. Guided by the family specification, it explores the anchor, tests candidate procedures, and decides what evidence to gather and what knowledge to retain (Section 3.1).

Educator review and Student response. The Educator has read-only access to recent Student work and the live Skill. It can explain, challenge, diagnose, propose checks, or remain silent. The Student can question, test, adapt, or reject advice, responding through dialogue, environment experiments, and package revisions. This interaction examines the assumptions behind a procedure and the evidence needed to retain it, with the Student deciding how the evidence changes the Skill.

Rounds and transfer. A fresh solver tests each round's package on the reset anchor, checking whether it carries the procedure beyond the author's session. The resulting trajectory and verifier feedback guide the next round. Student and Educator sessions persist; evaluation sessions are fresh. Only the final package transfers to held-out family members.

4.2 FIXED-WINDOW SSL

Fixed-Window SSL organizes acquisition as repeated periods of Student practice and Educator review. After every g completed Student tool calls, the Educator examines recent reasoning and tool results alongside the live Skill, informed by the family specification, prior lessons, and anchor feedback. This recurring review can challenge an assumption or suggest a missing check while an investigation is still open, shaping the evidence gathered before a procedure is retained.

The Student can clarify or contest a lesson, then resumes exploration and Skill revision. The resulting work informs the next review. Silence lets practice continue directly. The schedule thus fixes when review occurs, while guidance responds to the Student's progress. The Student determines when to finish the round; a final review covers any work since the previous review, allowing Student action before the snapshot (Appendix C.2).

4.3 ISSUE-GUIDED SSL

Teaching demand changes as a Skill develops: practice can settle a question, while new evidence can call for further guidance. Issue-Guided SSL tracks unresolved questions across revisions to select when review is needed and focus it on relevant evidence.

Persistent learning issues. An Educator lesson supplies a learning focus and an acceptance criterion: what to investigate and what observable evidence would resolve it. A learning issue retains these together with evidence and package revisions for subsequent review. Issues form a persistent LEARNING-ISSUE TREE: a concern arising within an investigation becomes a child; closing it returns attention to the nearest unresolved ancestor. Resolved branches preserve what has already been addressed.

Revisions and resolution. The Student groups file edits into a coherent revision, declaring its summary, evidence, and any issue addressed. An addressed issue can close with a changed package when the Student verifies the revision, documents the procedure's applicability with evidence, or rejects the guidance on evidential grounds. Verified closure requires a successful post-lesson check. Investigation can therefore settle a question without another Educator call.

Selective review and reopening. An initial review occurs at the first revision or after a warm-up window of tool calls. Thereafter, an unresolved issue must open, change, or reopen before further review. The Educator receives the active issue and its ancestors, the latest revision, and recent local work, with read access to the complete package. A lesson returns the Student to investigation; silence closes the current review episode. Contradictory fresh-solver anchor feedback can reopen the associated issue in the next round while preserving resolved siblings. Appendix C.2 gives event rules and boundary handling.

5 CAN SSL DEVELOP TRANSFERABLE AGENT SKILLS?

To assess how teaching supports Skill acquisition, we investigate four research questions (RQs): **RQ1** evaluates transfer to held-out family members; **RQ2** compares how expert guidance is organized across model pairs; **RQ3** traces teaching, Student actions, and package revisions during acquisition; and **RQ4** examines whether issue-guided review preserves transfer while reducing expert supervision. Evaluations follow the shared protocol in Section 3.

5.1 RQ1: TRANSFER TO HELD-OUT FAMILY MEMBERS

Do Skills acquired through interleaved teaching transfer to unseen members of the same task families?

Table 1 compares overall and domain-level transfer on the Qwen 22-family, 155-case main cohort. Qwen3.7 Plus serves as Student and fresh solver; Qwen3.7 Max supplies external expertise as Educator or direct package author. We compare the two SSL schedules with No Skill, direct Expert Authoring, Post-hoc Teaching, and adapted Trace2Skill and SKILL-KD. SSL, Expert Authoring, and Post-hoc Teaching use 3 acquisition rounds, with $g = 10$ for Fixed-Window SSL. Appendix C details

model settings, baseline adaptations, and session boundaries; Appendix D provides per-family results, case-level gains and regressions, and uncertainty estimates.

The two SSL schedules rank first and second overall and in each of the six domains. Fixed-Window SSL exceeds adapted Trace2Skill by 3.99 percentage points (95% paired family-bootstrap CI: $[1.45, 6.49]$) and adapted SKILL-KD by 4.99 points. These gains measure package reuse beyond the anchor and acquisition session.

The utility improvement extends to complete task solutions. Family-macro full-pass rate, requiring every private check to pass, rises from 32.5% without a Skill to 49.5% with Fixed-Window SSL and 50.8% with Issue-Guided SSL.

Table 1: **Overall and domain transfer on the Qwen main cohort.** Family-macro Avg@3 utilities across 22 families and 155 cases on a 0–100 scale; superscripts show gain over No Skill (pp). Domains: Sys. (systems), Serv. (services), Data (data processing), Sci. (scientific computing), Fin. (finance), and Office (office automation). Bold and underlined scores mark the best and second-best learning methods in each column.

Skill method	All	Sys.	Serv.	Data	Sci.	Fin.	Office
No Skill score	72.3	78.6	58.2	68.1	66.8	78.0	90.4
EXPERT AUTHORING	72.8$^{+0.5}$	77.9$^{-0.6}$	58.7$^{+0.6}$	71.6$^{+3.5}$	72.4$^{+5.7}$	72.5$^{-5.5}$	91.4$^{+1.0}$
POST-HOC TEACHING	74.8$^{+2.4}$	73.6$^{-4.9}$	64.4$^{+6.3}$	73.7$^{+5.6}$	76.4$^{+9.7}$	78.1$^{+0.0}$	91.3$^{+0.9}$
TRACE2SKILL	76.1$^{+3.8}$	85.0$^{+6.4}$	63.4$^{+5.2}$	71.0$^{+2.9}$	73.3$^{+6.5}$	77.9$^{-0.1}$	90.0$^{-0.4}$
SKILL-KD	75.1$^{+2.8}$	85.5$^{+6.9}$	60.9$^{+2.7}$	73.2$^{+5.1}$	68.2$^{+1.5}$	75.7$^{-2.4}$	90.3$^{-0.1}$
FIXED-WINDOW SSL	<u>80.1</u>$^{+7.8}$	**88.7**$^{+10.1}$	<u>67.3</u>$^{+9.1}$	<u>75.1</u>$^{+6.9}$	<u>77.0</u>$^{+10.2}$	<u>82.4</u>$^{+4.4}$	<u>94.1</u>$^{+3.7}$
ISSUE-GUIDED SSL	**80.5**$^{+8.2}$	<u>85.7</u>$^{+7.2}$	**71.0**$^{+12.8}$	**75.9**$^{+7.8}$	**77.1**$^{+10.3}$	**83.6**$^{+5.6}$	**94.6**$^{+4.2}$

5.2 RQ2: ORGANIZING EXPERT GUIDANCE DURING SKILL ACQUISITION

How should external expertise guide Skill acquisition?

Table 2(a) compares the four authoring protocols on the shared 10 families across the Qwen, DeepSeek, and Kimi model pairs, complemented by the Qwen main cohort. Within each pair, the solver and tasks are fixed; protocols differ in who investigates and writes the package, when guidance is available, and how the agents interact (Appendices C and D).

Guidance during practice consistently produces stronger transfer than reviewing the Student's completed authoring trajectory. Fixed-Window SSL and Post-hoc Teaching both preserve the Student as environment operator and package author, but Fixed-Window SSL outperforms Post-hoc Teaching on all three model pairs. On the main cohort, this advantage is 5.31 percentage points in utility with 21% fewer acquisition tokens. Interleaving lets the Student test advice while an investigation is still open and use the resulting evidence in subsequent exploration and writing. Post-hoc review arrives after those earlier choices have shaped the package. The results favor feedback that can redirect evidence gathering throughout practice.

The ordering also distinguishes guided practice from simply retaining a Student author. On the shared families, Post-hoc Teaching trails Expert Authoring in every model pair, whereas both interleaved protocols surpass Expert Authoring. This ordering favors coupling Student authorship with expert guidance during investigation. RQ3 examines how this interaction appears in Student actions and package revisions.

5.3 RQ3: HOW TEACHING ENTERS SKILL DEVELOPMENT

How do teaching, Student actions, and package revisions change as Skill acquisition progresses?

Figure 4 traces 630 reviews across 66 Qwen FIXED-WINDOW SSL runs with 3 rounds each, linking Educator-reported intents, Student actions, and package snapshots. We compare behavior following

Table 2: **Cross-model transfer and selective review.** (a) Utility gain over No Skill (pp) on the shared lite subset (10 families, 77 cases). (b) Utility and Educator savings: Qwen main cohort (22 families, 155 cases); DeepSeek and Kimi lite subset. Savings and efficiency ratios compare Issue-Guided with Fixed-Window SSL. Qwen uses three repeats; DeepSeek and Kimi use one. Model pairs are defined in Appendix C.1.

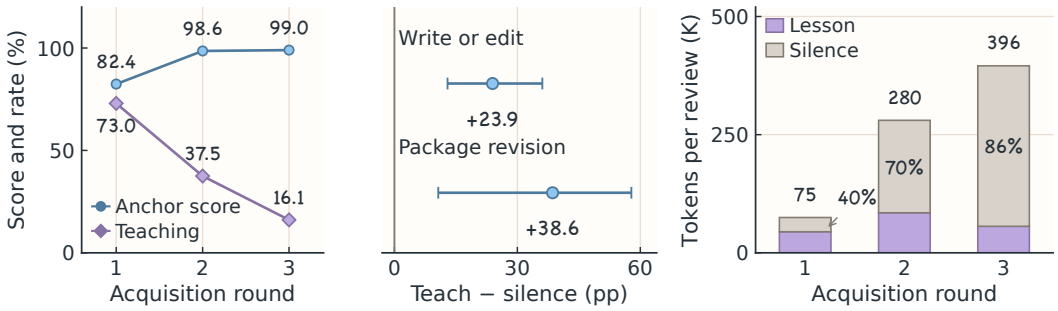
(a) Protocol comparison

Protocol	Qwen	DeepSeek	Kimi
FIXED-WINDOW SSL	+4.1	**+3.1**	**+8.2**
ISSUE-GUIDED SSL	**+4.5**	+2.5	+7.4
EXPERT AUTHORING	+1.1	+1.2	+5.8
POST-HOC TEACHING	−0.7	+0.5	+4.1

(b) Selective review

Metric	Qwen	DeepSeek	Kimi
Fixed-window utility	0.801	0.784	0.762
Issue-guided utility	0.805	0.779	0.753
Educator saved (%)	61.7	63.8	77.4
Efficiency ratio	2.75	2.2	3.9

teaching and silence at regular reviews, standardizing by round. Appendices E.1 and E.2 detail the taxonomy and estimation.



(a) Progress and teaching **(b) Writing and revision** **(c) Review cost**

Figure 4: **Teaching demand, Student development, and review cost.** Qwen Fixed-Window SSL ($g = 10$), 22 families and 66 acquisition runs. Panels show (a) anchor scores and teaching rates at regular reviews, (b) teaching–silence differences in writing on the next tool call and Skill changes over ten calls, standardized by round with 95% family-bootstrap intervals, and (c) tokens per review before follow-up, with silent shares.

Teaching demand falls while Skill development continues. As anchor utility improves, the fraction of regular reviews delivering a lesson falls from 73.0% in round one to 16.1% in round three, and advice shifts from reasoning toward diagnosis. Under the three-round protocol, packages still change in 98 of 99 rounds following a perfect anchor score, including rounds with no new lesson.

Compared with silence, lessons are followed by a 23.9-point higher probability of file writing or editing on the next call and a 38.6-point higher probability of a Skill change over the next ten calls, after adjustment for round (Figure 4b).

The Qwen HTML-finance case shows how Student checks refine teaching advice. The Educator proposes deriving reported values from other entries to detect inconsistency. The Student's initial implementation flags 4 metrics where only one is wrong. After inspecting this output, the Student replaces generic arithmetic with financial relations and reverse derivations, isolating the actual error (Table 3). The final Skill retains these relations and distinguishes an untested value from a contradicted one. On an unseen cash-flow report, a fresh solver reads the package, checks accounting relations,

Table 3: **Testing teaching advice.** Qwen Fixed-Window SSL, HTML-finance family, repeat 1. Recorded anchor checks; the erroneous report entry is line 210.

Verifier	Flagged	Selected line
Generic arithmetic	4	140
Financial relations	1	**210**

and passes 4/4 checks (Appendix G). The lesson initiates an investigation; the Student's execution evidence determines how the proposal becomes reusable knowledge.

Fixed review intervals become increasingly costly as this need for guidance declines. Average tokens per review grow 5.3-fold from the first to the last round as context accumulates; reviews returning silence account for 85.8% of final-round review tokens, before follow-up dialogue. This mismatch motivates selecting reviews around unresolved work while allowing the Student's investigation to continue.

5.4 RQ4: Selecting and focusing review around unresolved issues

Can selecting and focusing review around unresolved issues preserve transfer while reducing expert supervision?

Figure 5 compares fixed reviews every $g = 5, 10, 20$ Student tool calls with Issue-Guided SSL on the Qwen main cohort, using 3 rounds and 66 runs per schedule. We prioritize Educator tokens to measure demand for external expertise, and track how Student work changes total acquisition cost. Table 2(b) extends the comparison to DeepSeek and Kimi on the shared 10 families. Appendix C.3 defines the cost components for each model pair.

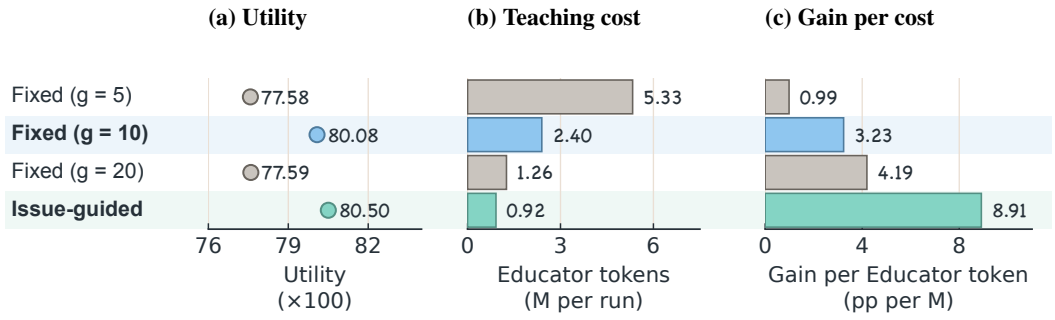


Figure 5: **Transfer utility and teaching efficiency across review schedules.** Qwen main cohort: 22 families, 155 cases, 3 repeats. Panels compare (a) family-macro utility, (b) Educator tokens, and (c) transfer gain over No Skill per million Educator tokens, with shared schedule rows.

Issue-guided review reaches 0.805 utility versus 0.801 for the standard $g = 10$ window, using 61.7% fewer Educator tokens and 16.9% fewer acquisition tokens. It achieves 2.75 times the transfer gain per Educator token.

Simply changing the interval does not recover this balance: $g = 5$ increases Educator cost, whereas $g = 20$ reduces it, but both yield lower utility. Compared with $g = 10$, Issue-Guided SSL reduces review count by 47.8% and tokens per review (including follow-up dialogue) by 26.7%.

On DeepSeek and Kimi, Issue-Guided SSL uses 63.8% and 77.4% fewer Educator tokens, respectively, while retaining utility within one percentage point of Fixed-Window SSL. Total acquisition tokens also fall for Kimi; additional Student work raises the total for DeepSeek (Appendix F).

6 Discussion and Conclusion

On shared SkillTransferBench families across three model pairs, Socratic Skill Learning (SSL) improves transfer over direct expert authoring and post-hoc teaching. Process and case analyses show how guidance enters Skill development: advice identifies a direction to investigate, while execution evidence informs how the Student tests, adapts, and retains a procedure.

As teaching demand changes, unresolved questions provide a basis for allocating expert attention. Issue-Guided SSL tracks these questions across revisions, retaining transfer utility with 2.75 times the gain per Educator token of Fixed-Window SSL on the main cohort. The study connects the development of reusable knowledge with decisions about where further guidance is needed.

Reproducibility Statement

Benchmark tasks, acquisition and evaluation code, and released trajectories are available at https://anonymous.4open.science/r/SocraticSkillLearning/. Appendix B documents benchmark construction and evaluation cohorts; Appendix C specifies model settings, acquisition protocols, session boundaries, and reproduction commands. Appendices E.2 and F detail the process analyses and token accounting.

AI Use Statement

We used generative AI tools to assist with drafting and adapting benchmark materials, implementing experimental code and analysis scripts, and analyzing recorded experimental results. We also used these tools for literature search, manuscript editing, and figure preparation. Benchmark materials were developed with human experts in coding and agent benchmarking. Each case underwent programmatic checks and review by an expert other than its author, following Appendix B. Numerical summaries were checked against recorded runs and analysis outputs. The authors take responsibility for the final content.

References

Anthropic. Agent skills overview. https://platform.claude.com/docs/en/agents-and-tools/agent-skills/overview, 2025. Accessed 2026-08-02.

Kaiyuan Chen, Qimin Wu, Taiyu Hou, Tianhao Tang, Xueyu Hu, et al. AgentIF-OneDay: A task-level instruction-following benchmark for general AI agents in daily scenarios. *arXiv preprint arXiv:2601.20613*, 2026.

Michelene T. H. Chi, Stephanie A. Siler, Heisawn Jeong, Takashi Yamauchi, and Robert G. Hausmann. Learning from human tutoring. *Cognitive Science*, 25(4):471–533, 2001.

Tianyi Guan, Yiding Wang, Haotong Yang, Siyuan Cao, Shirui Liu, Yi Hu, Jiaqi Li, and Muhan Zhang. ContinualSkillBench: Can LLM agents truly evolve their capabilities? *arXiv preprint arXiv:2608.03874*, 2026.

John Hattie and Helen Timperley. The power of feedback. *Review of Educational Research*, 77(1): 81–112, 2007.

Yangqin Jiang and Chao Huang. AgentBrew: Lifelong knowledge brewing from strong teachers to weak LLM agents. *arXiv preprint arXiv:2607.16851*, 2026.

Han Li, Yifan Yao, Letian Zhu, Rili Feng, Hongyi Ye, Jiaming Wang, Yancheng He, Pengyu Zou, Lehan Zhang, Xinping Lei, Haoyang Huang, Ken Deng, Ming Sun, Zhaoxiang Zhang, He Ye, and Jiaheng Liu. CodeTracer: Towards traceable agent states. *arXiv preprint arXiv:2604.11641*, 2026a.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, et al. SkillsBench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026b.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, et al. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097, 2022.

Zhiwei Li and Yong Hu. SkillHone: A harness for continual agent skill evolution through persistent decision history. *arXiv preprint arXiv:2606.08671*, 2026.

Hongjun Liu, Yifei Ming, Shafiq Joty, and Chen Zhao. Harnessing LLM agents with skill programs. *arXiv preprint arXiv:2605.17734*, 2026a.

Jiayu Liu, Zhenya Huang, Tong Xiao, Jing Sha, Jinze Wu, Qi Liu, Shijin Wang, and Enhong Chen. SocraticLM: Exploring socratic personalized teaching with large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024a. URL https://papers.neurips.cc/paper_files/paper/2024/hash/9bae399d1f34b8650351c1bd3692aeae-Abstract-Conference.html.

Siyao Liu, He Zhu, Jerry Liu, Shulin Xin, Aoyan Li, Rui Long, Li Chen, Jack Yang, Jinxiang Xia, Z. Y. Peng, Shukai Liu, Zhaoxiang Zhang, Ge Zhang, Wenhao Huang, Kai Shen, and Liang Xiang. FullStack Bench: Evaluating LLMs as full stack coders. *arXiv preprint arXiv:2412.00535*, 2024b.

Yuxuan Liu, Zhaochen Su, Lingyun Xie, Yuhao Zhang, Qing Zong, Jiahe Guo, Zhongwei Xie, Yiyan Ji, Yauwai Yim, Hongyu Luo, Xiyu Ren, Ruan Chenyu, Haoran Li, and Yangqiu Song. SkillRevise: Improving LLM-authored agent skills via trace-conditioned skill revision. *arXiv preprint arXiv:2606.01139*, 2026b.

Yuchen Ma, Yue Huang, Han Bao, Haomin Zhuang, Swadheen Shukla, Michel Galley, Xiangliang Zhang, and Stefan Feuerriegel. SkillGen: Verified inference-time agent skill synthesis. *arXiv preprint arXiv:2605.10999*, 2026.

Zeyao Ma, Bohan Zhang, Jing Zhang, Jifan Yu, Xiaokang Zhang, Xiaohan Zhang, Sijia Luo, Xi Wang, and Jie Tang. SpreadsheetBench: Towards challenging real world spreadsheet manipulation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 94871–94908, 2024. doi: 10.52202/079017-3007.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023.

Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, et al. Terminal-Bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *International Conference on Learning Representations*, 2026.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2Skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*, 2026.

Yinzhu Quan and Zefang Liu. EconSkills: Studying skill transfer and retrieval for web agents on live economic data. *arXiv preprint arXiv:2609.19523*, 2026.

Qiming Shi, Yibo Dou, Jiawen Zhu, Yulong Tao, Linbo Jin, Zhaolu Kang, Yunfan Zhou, and Di Weng. SKILL-KD: Contrastive skill distillation for LLM agents. *arXiv preprint arXiv:2607.28048*, 2026.

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.

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.

Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. OfficeBench: Benchmarking language agents across multiple applications for office automation. *arXiv preprint arXiv:2407.19056*, 2024.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936.

Xiangxin Zhao, Han Li, Shuaiting Li, Tianyi Zhao, Earl T. Barr, Federica Sarro, and He Ye. Failure as a process: An anatomy of CLI coding agent trajectories. *arXiv preprint arXiv:2607.09510*, 2026.

Shanshan Zhong, Yi Lu, Jingjie Ning, Yibing Wan, Lihan Feng, Yuyi Ao, Leonardo F. R. Ribeiro, Markus Dreyer, Sean Ammirati, and Chenyan Xiong. SkillLearnBench: Benchmarking continual learning methods for agent skill generation on real-world tasks. *arXiv preprint arXiv:2604.20087*, 2026.

A Positioning of Prior Skill-Learning Methods

Table 4 compares where feedback enters Skill acquisition and how package updates are produced (Ni et al., 2026; Ma et al., 2026; Shi et al., 2026; Zhong et al., 2026). These distinctions motivate the acquisition-protocol and review-schedule comparisons in the main study.

Table 4: Feedback and package updates in representative Skill-learning protocols. SSL combines external guidance during ongoing practice with Student-owned tool use and package writing.

Protocol	Feedback enters through	Skill update mechanism
Trace2Skill	Local trajectory analysis	Extract, merge, and apply lessons
SkillGen	Candidate verification	Generate, verify, and refine packages
SKILL-KD	Teacher–Student trajectory contrast and Student reruns	Consolidation agent proposes and validates patches
SkillLearnBench: Teacher Feedback	Advice after failure, using a reference Skill	Learner revises the Skill before retrying
Fixed-Window SSL	Educator review during practice at fixed tool-call intervals	Student tests advice and writes the live package
Issue-Guided SSL	Review on unresolved issue transitions with local evidence	Student revisions and anchor feedback update persistent issue state

B Benchmark Assets and Evaluation Cohorts

B.1 Construction routes

Experts in coding and agent benchmarking collaborate with agents to construct the task families. We first convert source tasks to a common format, reconciling paths, formats, ordering, precision, and boundary conditions across the task descriptions, fixtures, and scorers. For example, adapting the WAL anchor preserves the source database fixtures while revising task metadata and strengthening checks of the recovered output.

In extraction and adaptation, compatible source cases are grouped into families, and statements and tests are revised to align their requirements. In seed expansion, a case from the adapted pool seeds a specification of the shared invariant and permitted variation; experts and agents then develop additional cases within this scope. Each task includes a description, public fixtures, a resettable environment, a reference solution, and a private verifier.

For example, WAL-recovery cases vary encoding, integrity, and recovery depth while preserving the reconstruction goal. Acquisition uses a fixed `v01` anchor in each family.

B.2 Catalog and variation axes

Table 5 distinguishes the full catalog, including one anchor per family, from the main held-out cohort. A dash denotes a family outside that cohort. Table 6 lists the public variation axes for the evaluated families. The catalog contains 35 anchors and 373 non-anchor cases, with 5–17 cases per family; family members are complete executable tasks.

Table 5: Curated task-family catalog and main evaluation cohort, showing case counts for each family within domain groupings and totals across the catalog.

Domain	Family	Catalog cases	Main held-out
Systems	Deterministic tarball	12	6
	ACL inheritance	13	6
	SQLite truncation	14	6
	DB WAL recovery	11	7
	Git leak recovery	13	6
	Recover obfuscated files	7	–
	Shell deobfuscation	10	–
	Service decommissioning	6	–
	Password recovery	5	–
	Vulnerable secret	6	–
Services	API scheduler	15	10
	PyPI server	16	10
	KV-store gRPC	16	6
	Flask	9	6
	WASM pipeline	11	–
Data Processing	Filter JS from HTML	16	10
	SPARQL professors	15	10
	Organization JSON	13	8
	JQ processing	12	7
	Multi-source merger	14	–
	Cocktail recipes	8	–
Scientific Computing	Cross-entropy method	13	6
	Audio STFT peaks	11	10
	ODE RK4	15	5
	Camera layout	16	–
Finance	Bank transaction filter	16	6
	Financial documents	13	6
	HTML finance verification	16	6
	Financial analysis	7	6
	Stock analysis	11	–
Office Automation	Office workflow	8	6
	XLSX structure	10	6
	XLSX formula	17	–
	Excel automation	7	–
	PPTX editing	6	–
Total: 35 catalog families; 22 main families		408	155

B.3 Validation and acceptance

Acceptance requires a stable family invariant with declared variation, unambiguous requirements with deterministic tests, tasks that go beyond mechanical edits, and at least three separately attributable checks. Programmatic validation verifies Docker builds, executes tests, and confirms that reference solutions pass them. An expert other than the author reviews each case before acceptance. Reference solutions and private tests remain outside agent workspaces.

Curation yields 35 families and 408 cases from 37 candidate families and 444 cases. Reference solutions pass all 2,807 checks (3–34 per case).

B.4 Evaluation cohort

The main cohort selects families with at least 5 verified held-out cases and at least two non-perfect outcomes in catalog-wide No-Skill screening. It contains 22 families and 155 held-out cases, with 5–10 cases per family, retaining ceiling cases for coverage and regression measurement. Appendix D.2 reports repeat-level scores and case-level transfer directions.

Table 6: Public variation axes for each family in the main evaluation cohort, summarizing the changes across cases declared in the family specifications.

Family	What changes across cases
Filter JS from HTML	Executable constructs, preservation requirements, malformed syntax, encodings.
API scheduler	Task volume, emergencies, priority rules, dependencies, API capabilities.
SPARQL professors	Rank semantics, affiliation paths, graph noise and duplicate bindings.
Audio STFT peaks	Signal recipe, STFT settings, peak selection and shell output contract.
PyPI server	Package layout, public API, index configuration, publication and installation.
Organization JSON	Entity relationships, field normalization, aggregation distributions.
DB WAL recovery	WAL encoding, base/frame integrity, recovery depth, schema and update topology.
JQ processing	Selection and ordering, reshaping, normalization, collection-level aggregation.
Cross-entropy method	Environment dynamics, elite-update regime, shared-prefix plan evaluation.
Deterministic tarball	Source contents and exclusions, metadata normalization, archive/compressor format.
Flask	Stateful resources, coupled endpoints, validation and failure behavior.
KV-store gRPC	RPC schema, persistent state semantics, service lifecycle and request patterns.
HTML finance verification	Report structure, accounting relations, error type, output precision.
ODE RK4	Differential dynamics, evaluation schedule, step and efficiency constraints.
Bank transaction filter	Noisy identity signals, membership propagation, confusable distractors.
Git leak recovery	Object/ref location, repository topology, recovery and cleanup scope.
Financial documents	Content-based classification, OCR/modalities, amount expression, batch composition.
SQLite truncation	Corruption mechanism, value encoding, record identification, recovery scale.
ACL inheritance	User/group policy topology, effective permissions, descendant inheritance.
Financial analysis	Source formats, cross-statement joins, metric definition, locale and schema.
Office workflow	Source/destination artifacts, temporal routing, cross-output consistency.
XLSX structure	Transformation, workbook topology, alignment, output geometry, preservation.

C Protocols, Implementation, and Reproduction

C.1 Runtime, sessions, and package boundaries

Runtime and models. We abbreviate the three model pairs as Qwen, DeepSeek, and Kimi. In each pair, the Student model also serves as the fresh solver, while the Educator model supplies external expertise or writes the package under Expert Authoring. The harness uses Pi coding agent 0.81.1[2], TypeScript 5.9.3, and Docker task environments. Main Qwen runs use `qwen3.7-plus` as Student and `qwen3.7-max` as Educator, with a 1,000,000-token context window, a 65,536-token maximum completion, and thinking enabled. Sampling uses the provider defaults. DeepSeek uses `deepseek-v4-flash-0731` as Student and `deepseek-v4-pro-0813` as Educator, with the same context window, 393,216 maximum completion tokens, thinking enabled, and reasoning effort `max`. Kimi uses `kimi-k2.5` for the Student and fresh solvers (262,144-token context, 16,384 maximum completion, thinking enabled), and `kimi-k3` for the external author or Educator (1,048,576-token context, 131,072 maximum completion, reasoning effort `max`).

Session isolation and access. Within an original authoring-protocol run, the package author (the external model under Expert Authoring; the Student otherwise) and, where applicable, the Educator maintain separate continuous sessions across acquisition rounds, with automatic context compaction. Sessions never continue across families, replicates, or protocol cells. Every anchor and held-out evaluation uses a fresh target-model solver. Adapted SKILL-KD instead uses fresh executions with persistent rule-edit history (Appendix C.3.1).

The author operates in a Docker workspace containing the public task, anchor inputs, and family specification, and is the only agent that writes the live Skill. Educator tools list, read, and search the full package but cannot execute environment actions or write files. Solvers load only the frozen target package through the native Skill catalog, with ambient Skills disabled and a clean case workspace. The

[2] https://pi.dev

package contains `SKILL.md` and optional `scripts/`, `references/`, and `assets/` directories. Held-out materials remain inaccessible during acquisition, and private checks are applied only after the solver submits its solution. Snapshots retain every package file; resource reads and executable invocations are logged.

C.2 Acquisition loop and review events

Review context and interaction. Fixed-Window SSL reviews after every g completed Student tool calls, with $g = 10$ in the main condition and $g = 5, 20$ in the schedule comparison. Fixed-Window SSL provides the Educator with the family definition, recent reasoning and tool calls, prior lessons, feedback from anchor tests by fresh solvers, and the complete live package. An opening lesson may explain, propose a test, diagnose, or challenge an assumption; the Student may clarify, adapt, or contest it, followed by at most one Educator reply. Subsequent task actions and package edits remain Student decisions.

Round completion. The author ends its trajectory when it has finished the current exploration and package revision. The harness completes the scheduled terminal review, if any, and the Student's resulting work, then validates and snapshots the package for a fresh anchor test. Each round permits at most one terminal review; resulting Student work is retained without another terminal review. The tool-call window controls review frequency, while the author determines trajectory length. All 3 rounds run, with anchor feedback entering the next round.

The loop below preserves Student package ownership under both schedules.

```
create Student; create Educator; initialize package K
for round in 1..3:
  give Student public spec, clean anchor, previous feedback
  while the current authoring trajectory is active:
    let Student reason, invoke tools, and edit K
    Fixed-Window SSL: review after g completed tool calls
      or at an unreviewed end-of-round tail
    Issue-Guided SSL: on bootstrap, revision commit, or contradiction:
      resolve addressed active issue if closure conditions hold
      otherwise update, open, or reopen its persistent issue
      request review only for an unresolved transition
    if review requested:
      Educator observes local evidence and complete package
      if lesson: Student responds
        fixed-window permits one Educator follow-up
      else: retain Issue-Guided SSL issue with no-lesson closure reason
      Student resumes its own work
  freeze K; evaluate with fresh target model on clean anchor
  retain trajectory and verifier feedback for the next round
evaluate final K with a fresh target model for each held-out case
```

Revision declarations and issue state. During practice, the Student calls `commit_skill_revision` after a coherent knowledge change to the complete package. The declaration includes a summary and evidence, plus an issue ID and disposition when it addresses an existing lesson. Ordinary file edits accumulate until the Student commits such a revision. The Educator supplies a learning focus and an observable acceptance criterion when delivering a lesson.

Each persistent issue stores a focus, acceptance criterion, linked revisions, evidence, and disposition. A child records a concern within ongoing work; closure returns attention to the nearest unresolved ancestor. In Issue-Guided SSL, the first-round bootstrap occurs at the first revision or after ten tool calls, whichever comes first. At the round boundary, an uncommitted package change is processed as an implicit revision; the runtime also handles any pending bootstrap or anchor contradiction before the final snapshot. The default Educator view includes up to three active-path nodes, two recently closed issues, the latest revision, and recent local tool calls; read access to the complete package remains available. After bootstrap, another unresolved transition precedes further teaching, so triggers, context, and follow-up policy change together.

Student-declared closure requires an addressed active issue and a changed package. A verified disposition additionally requires a successful post-lesson verification event; bounded and rejected dispositions record stated evidence. A no-lesson closure records Educator silence rather than verified correctness. Contradictory feedback from an anchor test by a fresh solver targets the issue associated with that package and may reopen it while preserving resolved siblings.

C.3 Acquisition comparators

Expert Authoring assigns exploration and package writing to the external model, including feedback from multi-round anchor tests by fresh solvers. Post-hoc Teaching keeps the Student as author and offers one review after the authoring trajectory in each round. A lesson is followed by Student checks and revisions; this protocol has no immediate Educator follow-up. Table 7 summarizes the interaction structure. No Skill skips acquisition and solves each case in a fresh session. Our adaptation of the released Trace2Skill retains parallel patch proposal and consolidation over independent anchor trajectories generated from a generic seed package, replacing the spreadsheet-specific front end with a task-family adapter. The main authoring comparison contains 198 acquisition runs; Issue-Guided SSL adds 66.

Cost accounting. Acquisition tokens include cached input and exclude held-out solving. Qwen totals cover the author, Educator, and fresh-solver anchor evaluations; DeepSeek and Kimi totals cover the persistent author and Educator sessions. Teaching efficiency divides transfer gain by Educator tokens.

Table 7: Authoring and teaching protocols. Each uses 3 acquisition rounds with fresh-solver anchor feedback between rounds.

Protocol	Operator / author	Review opportunity	Continuation
Expert Authoring	External model	No separate Educator	Author revises from anchor feedback
Post-hoc Teaching	Student	Once after the authoring trajectory in each round	Student checks and revises
Fixed-Window SSL	Student	Every g tool calls and an unreviewed tail	Student responds; one optional Educator follow-up
Issue-Guided SSL	Student	Unresolved issue transitions	Student investigates and commits revisions before further review

C.3.1 Adapted SKILL-KD

Adaptation. We use the released SKILL-KD loop (Shi et al., 2026)[3] at upstream revision `f76b51c3e855`, with adapter revision `54954317354d`; full hashes are retained in the compact evidence. Each of 66 independent runs starts from a generic rule seed and makes three visits to its public anchor. A successful Student attempt leaves the rules unchanged. Otherwise, Teacher and Student trajectories inform a critic's proposed rule edits, accepted only after a fully successful fresh Student rerun. Each visit permits at most three adaptive rounds; failed candidates roll back to the visit-start rules. Qwen3.7 Plus is the Student and held-out solver; Max is Teacher and critic. Every execution uses a fresh session. Only accepted rules, not private edit history, enter the frozen Skill.

Separate cost scope. The physical-usage audit records 128.11M acquisition tokens across 66 runs (1.94M per run). The acquisition-cost columns show a dash for this baseline, since the audit follows a different scope.

[3] https://github.com/KhanCold/SKILL-KD

C.4 REPRODUCTION

The anonymous repository contains benchmark tasks in `bench/`, the execution harness in `eval/`, and released records in `traj/`. Within `eval/`, `config/scopes/main.yaml` selects the 22-family, 155-case main cohort, and `config/scopes/lite.yaml` selects the shared 10-family subset. Model pairs are specified separately in `config/models/`. Method identifiers match the paper names: `expert-authoring`, `post-hoc-teaching`, `fixed-window-ssl`, and `issue-guided-ssl`.

After installing Node.js and Docker, configure `SSL_URL` and `SSL_KEY` in `eval/.env`. The following commands run both SSL schedules or evaluate their released Qwen packages directly:

```
cd eval
npm ci
cp .env.example .env
npm run ssl -- run --scope main --model qwen \
  --methods fixed-window-ssl,issue-guided-ssl
npm run ssl -- evaluate --scope main --model qwen \
  --methods fixed-window-ssl,issue-guided-ssl \
  --frozen ../traj/qwen-22
```

Add `--dry-run` to inspect a run plan, or use `acquire` and `evaluate` as separate stages. `eval/README.md` documents the `trace2skill` and `skill-kd` baseline commands and setup. `traj/README.md` maps experiment cohorts to teaching sessions, Skill packages, and available evaluation records. Acquisition uses only the anchor and public family specification; held-out evaluation uses fresh solver sessions with the resulting package.

D SUPPLEMENTARY TRANSFER RESULTS

D.1 OVERALL AND PER-FAMILY COMPARISONS

Table 8 gives family-macro Avg@3 utility and full-pass rate, paired 95% bootstrap confidence intervals for utility gains, and mean acquisition cost on the Qwen 22-family, 155-case main cohort. A cost dash denotes no acquisition for No Skill and unavailable common-scope cost for the adapted baselines. Table 9 provides the full per-family values underlying Table 1.

Table 8: Utility, full-pass rate, and acquisition cost on the Qwen 22-family, 155-case main cohort (Avg@3, family-macro). Gain and 95% CI refer to utility differences from No Skill.

Condition	Utility ↑	Gain (pp)	95% CI	Full-pass (%) ↑	Tokens/run (M)
No Skill	0.723	–	–	32.50	–
EXPERT AUTHORING	0.728	+0.46	$[-5.46, 6.56]$	40.44	4.95
POST-HOC TEACHING	0.748	+2.44	$[-3.36, 7.90]$	40.37	8.71
Adapted TRACE2SKILL	0.761	+3.76	$[-0.50, 8.96]$	39.11	–
Adapted SKILL-KD	0.751	+2.76	$[-1.59, 8.09]$	41.65	–
FIXED-WINDOW SSL ($g = 10$)	0.801	+7.75	$[3.53, 12.68]$	49.49	6.89
ISSUE-GUIDED SSL	**0.805**	**+8.18**	$[3.54, 13.57]$	**50.82**	5.72

Full-pass gains over No Skill are 16.99 percentage points for FIXED-WINDOW SSL (95% paired family-bootstrap CI: $[8.13, 26.86]$) and 18.33 points for ISSUE-GUIDED SSL ($[9.44, 28.12]$), using the same family resampling as the utility comparisons.

Fixed-Window SSL's paired gains are 7.29 points over Expert Authoring, 5.31 over Post-hoc Teaching, 3.99 over Trace2Skill, and 4.99 over adapted SKILL-KD. It improves on No Skill in 16 of the 22 families and uses 21% fewer acquisition tokens than Post-hoc Teaching. The 95% bootstrap confidence interval over task families for Fixed-Window SSL–Trace2Skill is $[1.45, 6.49]$ percentage points. The intervals for Post-hoc Teaching–Expert Authoring, Fixed-Window SSL–Post-hoc Teaching, and Fixed-Window SSL–Expert Authoring are respectively $[-1.57, 5.64]$, $[2.34, 9.09]$, and $[3.62, 11.44]$ points. The Fixed-Window SSL–adapted SKILL-KD interval is $[1.79, 8.48]$ points. All intervals resample the same paired family differences.

Table 9: Per-family Avg@3 scores on the Qwen main cohort (22 families, 155 cases), including adapted Trace2Skill and SKILL-KD. Only the last row is a macro-average across task families.

Family	No Skill	Expert Authoring	Post-hoc Teaching	Trace2Skill	SKILL-KD	Fixed-Window SSL	Issue-Guided SSL
ACL inheritance	0.933	0.728	0.858	0.885	0.933	0.933	0.881
API scheduler	0.767	0.730	0.745	0.897	0.921	0.791	0.830
Audio STFT peaks	0.563	0.631	0.652	0.608	0.664	0.643	0.656
Bank transaction filter	0.678	0.548	0.550	0.522	0.515	0.593	0.652
Cross-entropy method	0.553	0.556	0.642	0.617	0.556	0.667	0.656
DB WAL recovery	0.473	0.826	0.705	0.905	0.901	0.884	0.878
Deterministic tarball	0.678	0.691	0.677	0.706	0.682	0.728	0.693
Filter JS from HTML	0.366	0.350	0.403	0.363	0.391	0.433	0.440
Financial analysis	0.792	0.750	0.875	0.747	0.792	0.875	0.875
Financial documents	0.921	0.603	0.754	0.913	0.897	0.913	0.873
Flask	0.408	0.406	0.638	0.516	0.396	0.690	0.785
Git leak recovery	0.929	0.905	0.872	0.896	0.928	0.975	0.961
HTML finance verification	0.731	1.000	0.944	0.935	0.824	0.917	0.944
JQ processing	0.875	0.835	0.853	0.889	0.867	0.889	0.892
KV-store gRPC	0.607	0.635	0.630	0.586	0.584	0.625	0.635
ODE RK4	0.887	0.987	1.000	0.973	0.827	1.000	1.000
Office workflow	0.875	0.905	0.907	0.867	0.873	0.959	0.964
Organization JSON	0.778	1.000	1.000	0.875	1.000	1.000	1.000
PyPI server	0.544	0.578	0.564	0.536	0.533	0.583	0.589
SPARQL professors	0.706	0.680	0.690	0.713	0.672	0.680	0.705
SQLite truncation	0.915	0.747	0.569	0.856	0.830	0.915	0.875
XLSX structure	0.933	0.923	0.919	0.933	0.933	0.923	0.928
Macro average	0.723	0.728	0.748	0.761	0.751	0.801	**0.805**

D.2 Repeat variation, non-ceiling cases, and sensitivity

Table 10 compares each case's Avg@3 score with its No-Skill score. Counts are independent of family or domain weighting. The ceiling subset contains the 25 cases with No-Skill score 1 across all repeats; the other 130 cases are non-ceiling. Fixed-Window SSL improves 55.4% of non-ceiling cases, and Issue-Guided SSL improves 54.6%.

Table 10: Case-level Avg@3 transfer directions relative to No Skill on the Qwen main cohort (22 families, 155 cases), with non-ceiling and ceiling subsets for Fixed-Window SSL and Issue-Guided SSL.

Method	Improved	Unchanged	Regressed
All cases ($n = 155$)			
Expert Authoring	58	56	41
Post-hoc Teaching	57	53	45
Adapted Trace2Skill	59	65	31
Fixed-Window SSL	72	59	24
Issue-Guided SSL	71	63	21
Adapted SKILL-KD	55	69	31
Non-ceiling ($n = 130$)			
Fixed-Window SSL	72	37	21
Issue-Guided SSL	71	43	16
Ceiling ($n = 25$)			
Fixed-Window SSL	0	22	3
Issue-Guided SSL	0	20	5

Every Fixed-Window SSL and Issue-Guided SSL repeat exceeds its corresponding No-Skill repeat. SD is the sample standard deviation across the 3 repeat-level scores. The final column averages the within-family sample SD across 22 families.

Table 11: Repeat variation on the Qwen main cohort (22 families, 155 cases): family-macro utility for each repeat, its standard deviation across repeats, and mean within-family variability.

Condition	Repeat 1	Repeat 2	Repeat 3	SD	Mean within-family SD
No Skill	0.719	0.730	0.721	0.006	0.036
FIXED-WINDOW SSL	0.772	0.799	0.831	0.030	0.031
ISSUE-GUIDED SSL	0.767	0.810	0.838	0.036	0.037
Adapted SKILL-KD	0.746	0.756	0.750	0.005	0.039

D.3 ADDITIONAL MODEL PAIRS

DeepSeek-V4-Flash-0731 / DeepSeek-V4-Pro-0813 and Kimi K2.5 / Kimi K3 independently acquire packages on a 10-family, 77-case lite subset of the main cohort: Audio STFT peaks (10 cases), Filter JS from HTML (10), Git leak recovery (6), HTML finance verification (6), JQ processing (7), KV-store gRPC (6), PyPI server (10), SPARQL professors (10), SQLite truncation (6), and XLSX structure (6). Each condition has 1 repeat, 3 acquisition rounds, and 77 held-out evaluations in fresh sessions with equal weighting across task families. Each model-pair experiment comprises 40 acquisition runs, 120 anchor evaluations by fresh solvers, 308 Skill-based held-out evaluations, and 77 No-Skill evaluations, and compares four acquisition protocols and No Skill.

On all three model pairs, the two interleaved protocols take the top two positions and Expert Authoring ranks third (Table 2(a)). Fixed-Window SSL exceeds Post-hoc Teaching by 4.88, 2.54, and 4.09 points on the shared families for Qwen, DeepSeek, and Kimi, respectively. Fixed-Window SSL has the highest mean utility in both additional pairs, gaining 3.08 and 8.24 points over the respective No-Skill solvers; Table 12 gives the complete comparison. Table 13 reports paired family bootstrap intervals for the headline contrasts, and Table 14 lists the per-family scores.

Table 12: Utility and gain over No Skill for DeepSeek and Kimi on the lite subset (10 families, 77 cases; single repeat).

	DeepSeek		Kimi	
Condition	Utility	Gain (pp)	Utility	Gain (pp)
No Skill	.754	–	.679	–
EXPERT AUTHORING	.766	+1.21	.738	+5.85
POST-HOC TEACHING	.759	+0.54	.721	+4.15
FIXED-WINDOW SSL	**.784**	**+3.08**	**.762**	**+8.24**
ISSUE-GUIDED SSL	.779	+2.50	.753	+7.36

Table 13: Headline contrasts on the shared lite subset (10 families, 77 cases): 95% paired family-bootstrap intervals for Table 2(a). Qwen averages 3 repeats; DeepSeek and Kimi use one.

Contrast	Qwen (shared)	DeepSeek	Kimi
FIXED-WINDOW SSL − No Skill	[+1.02, +8.03]	[−4.63, +13.03]	[+2.48, +15.62]
ISSUE-GUIDED SSL − FIXED-WINDOW SSL	[−0.84, +1.44]	[−5.34, +3.14]	[−4.72, +2.89]
FIXED-WINDOW SSL − EXPERT AUTHORING	[−0.83, +7.14]	[−1.93, +7.15]	[−2.33, +7.98]
EXPERT AUTHORING − POST-HOC TEACHING	[−1.28, +6.05]	[−0.89, +2.23]	[−1.13, +4.76]

Acquisition costs and review scheduling for these pairs appear in Appendix F.

Table 14: Per-family scores for DeepSeek and Kimi on the lite subset (10 families, 77 cases; single repeat). Qwen scores for these families appear in Table 9.

Family	No Skill	Expert Authoring	Post-hoc Teaching	Fixed-Window SSL	Issue-Guided SSL
DeepSeek					
Audio STFT peaks	.668	.657	.677	.637	.570
Filter JS from HTML	.315	.485	.452	.712	.529
Git leak recovery	.963	.891	.891	.912	.912
HTML finance verification	.833	1.000	1.000	1.000	1.000
JQ processing	.909	.891	.846	.817	.835
KV-store gRPC	.656	.635	.600	.668	.689
PyPI server	.558	.617	.592	.642	.642
SPARQL professors	.718	.652	.672	.652	.722
SQLite truncation	.972	.898	.931	.875	.944
XLSX structure	.944	.933	.931	.931	.944
Kimi					
Audio STFT peaks	.450	.635	.585	.660	.660
Filter JS from HTML	.248	.366	.314	.593	.456
Git leak recovery	.867	.891	.915	.891	.888
HTML finance verification	.833	1.000	.917	.889	1.000
JQ processing	.826	.817	.846	.891	.864
KV-store gRPC	.607	.635	.635	.635	.635
PyPI server	.542	.558	.600	.583	.633
SPARQL professors	.592	.647	.652	.677	.622
SQLite truncation	.907	.913	.812	.866	.838
XLSX structure	.921	.917	.933	.933	.933

E Teaching Process Analysis

E.1 Teaching-intervention taxonomy and dialogue turns

The Educator records intent labels when issuing each teaching tool call. These labels describe its intended change in learner state. A lesson may receive multiple labels; silence is a separate decision. The codebook distinguishes task knowledge, reasoning, evidence acquisition, diagnosis, calibration, and consolidation, following the emphasis on task, process, and self-regulation in feedback research and active construction during tutoring (Hattie & Timperley, 2007; Chi et al., 2001).

Table 15: Intent definitions and tag frequencies for Qwen Fixed-Window SSL ($g = 10$; 22 families, 66 acquisition runs). Opening lessons and follow-up turns use separate denominators of 254 and 91.

Intent	Operational target	Opening	Follow-up
Knowledge transfer	Facts, mechanisms, APIs, formats, boundaries, or expert experience	101 (39.8%)	19 (20.9%)
Reasoning guidance	Representation, decomposition, causal reasoning, comparison, generalization, or decisions	168 (66.1%)	43 (47.3%)
Exploration/verification	Search, experimental design, evidence collection, validation, or tool-use procedure	23 (9.1%)	21 (23.1%)
Error diagnosis/recovery	Attribution of an observed error or stuck state and a recovery direction	99 (39.0%)	46 (50.5%)
Epistemic calibration	Unsupported assumptions, over-generalization, evidence levels, or confidence	39 (15.4%)	11 (12.1%)
Experience consolidation	Retrievable triggers, rules, workflows, scripts, or references from local experience	104 (40.9%)	25 (27.5%)
Other	A target not captured by a combination of the six core labels	0	0

Reasoning is the most common opening intent; diagnosis and verification are more frequent in follow-ups.

E.2 REVIEW DECISIONS AND SUBSEQUENT STUDENT BEHAVIOR

We relate teaching and silence to the Student's next action and subsequent Skill revision. The analysis covers 66 FIXED-WINDOW SSL runs from the evaluation manifest, using the latest complete attempt for each family and repeat. They contain 630 reviews, 254 opening lessons, and 91 follow-ups. Regular ten-tool-call reviews include 226 teaching and 206 silent decisions; trajectory-end reviews include 28 and 170, respectively.

Observation windows. Each window begins when the Student receives a lesson or a silent review completes, and ends at the next review or round boundary. We use event order and tool-call IDs to identify actions begun within the window. Initial prompts delimit rounds; recovery messages remain within the current round.

Across 432 regular windows, we classify the first subsequent tool call as environment, write/edit, read/search, or absent. Trajectory-end reviews are analyzed separately: 24/28 lessons precede a tool call, while all 170 silent reviews end the round.

Skill revision over ten tool calls. The revision analysis uses 193 windows containing exactly ten calls begun after review, with no continuation of an earlier call. We compare the Skill versions at the start and end of each window. Both versions are observable in 112 windows; the other 81 lack a comparable pair. Among these 112 windows, the package changes in 67/69 teaching windows and 27/43 silent windows. A sensitivity analysis relaxes the ten-call restriction and includes 316 windows with comparable versions.

Estimation. These observational comparisons measure associations with teaching: review decisions respond to the Student's ongoing work. We standardize contrasts by acquisition round, using rounds with both comparison groups and weighting them by the eligible sample's round distribution. Table 16 reports the number of supported observations with known outcomes (n). Confidence intervals resample the 22 whole families, keeping each family's runs and reviews together and retaining the round weights. Pre-review comparisons use recorded tool composition, error flags, shell exit status, and package state. Tool-error and no-error groups contain 111 and 321 regular windows; the observable/unobservable package comparison is supported in round one.

Writing and revision after teaching. Following teaching, the Student's next tool call is 23.9 percentage points more likely to write or edit a file than following silence (95% CI: $[13.0, 36.1]$). Over the next ten tool calls, the Skill package is 38.6 points more likely to change ($[10.7, 57.8]$), after standardizing by round. A within-run check gives a mean package-change difference of 34.3 points over 18 runs containing both decision types.

Table 16: Round-standardized review-behavior contrasts for Qwen FIXED-WINDOW SSL ($g = 10$; 22 families, 66 acquisition runs). Counts give supported observation windows, and brackets give cluster-bootstrap confidence intervals over task families.

Outcome and comparison	n	Adjusted difference (pp)
Teaching: observed tool error vs no error	432	-0.4 [-9.2, $+8.1$]
Teaching: shell failure vs no failure	413	-8.5 [-20.8, $+2.6$]
Teaching: observable vs unobservable package	233	-31.5 [-41.7, -22.8]
Teaching: preceding package change vs none	130	-10.2 [-28.7, $+12.4$]
Teaching: environment-dominant vs other work	432	$+1.5$ [-8.2, $+10.1$]
Teaching: write-dominant vs other work	432	$+1.8$ [-21.4, $+25.2$]
Teaching: read-dominant vs other work	432	$+4.6$ [-10.2, $+19.4$]
Environment first: teaching vs silence	432	$+0.9$ [-11.5, $+13.0$]
Write/edit first: teaching vs silence	432	$+23.9$ [$+13.0$, $+36.1$]
Read/search first: teaching vs silence	432	-17.2 [-25.0, -8.7]
No subsequent tool: teaching vs silence	432	-7.7 [-10.9, -4.5]
Package change, ten tools: teaching vs silence	112	$+38.6$ [$+10.7$, $+57.8$]
Package change, all intervals: teaching vs silence	316	$+60.1$ [$+49.2$, $+70.6$]

E.3 Learning progress and review cost

Each of the 66 runs completes all 3 acquisition rounds, with practice continuing after anchor success. We link the resulting 198 fresh-solver anchor evaluations to package snapshots. The unit of analysis is a later round paired with its preceding anchor result, yielding 132 transitions with comparable snapshots.

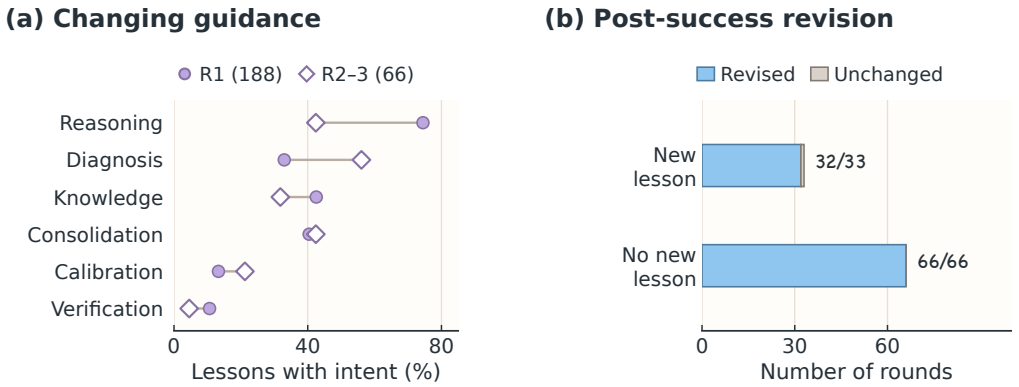


Figure 6: **Teaching emphasis and development after anchor success.** Qwen Fixed-Window SSL ($g = 10$), 22 families and 66 acquisition runs. (a) Educator-reported intents in first-round versus later lessons; labels may overlap. (b) Package revisions in rounds following a perfect anchor score, grouped by whether a new lesson arrives.

Teaching across rounds. Figure 4a uses regular-review lesson counts of 170/233, 42/112, and 14/87. Including trajectory-end reviews gives 188/299, 47/178, and 19/153. Utility on anchor tests by fresh solvers is 0.824, 0.986, and 0.990, with 38/66, 61/66, and 63/66 perfect results. Figure 6a compares 188 first-round lessons with 66 later lessons; reasoning tags occur in 140 versus 28 and diagnosis tags in 62 versus 37.

Development after anchor success. Of the 132 later-round transitions, 99 follow a perfect anchor (38 second rounds and 61 third rounds), covering 61 runs and all 22 families. Packages change in 98/ 99: 32/33 rounds with a new lesson and 66/66 without one. Those 33 rounds contain 35 lessons. Across all later rounds, the 76 rounds without a new lesson all revise their package. Prior lessons and anchor feedback remain available.

Review cost before follow-up. Silent reviews account for 59.7% of review opportunities and 72.1% of opening-review tokens. Figure 4c includes cached input and excludes follow-up dialogue. Each round's teaching and silent token totals are divided by its total review count, so stacked heights give mean cost per review. Round totals are 22.3M, 49.9M, and 60.5M tokens; silent shares are 40.4%, 69.8%, and 85.8%. Average token use per review rises from 74.7K to 395.7K tokens per review across acquisition rounds, a 5.3-fold increase. Appendix F reports schedule-level costs including follow-up dialogue.

The corresponding teaching sessions and training-iteration evaluations are released under `traj/qwen-22/fixed-window-ssl/`; each run's `meta.json` lists its session inventory.

F Review Scheduling and Cost Allocation

F.1 Schedules and role-level cost

All four Qwen schedules use 22 families, 3 acquisition rounds, and 3 repeats (66 runs each). The ten-tool-call condition is the main Fixed-Window SSL experiment. Table 17 reports utility, cost, and teaching activity (lessons exclude follow-ups), with mean acquisition tokens split into Student-author, Educator, and anchor-evaluation work (Figure 7).

Table 17: Utility, acquisition cost, reviews, and opening lessons for the four Qwen schedules on the main cohort (22 families, 155 cases; 3 repeats and 66 acquisition runs per schedule).

Teaching schedule	Utility ↑	Tokens/run (M) ↓	Reviews	Lessons
Every 5 tool calls	0.776	9.56	1126	403
Every 10 tool calls	0.801	6.89	630	254
Every 20 tool calls	0.776	6.61	375	169
Issue transitions (Issue-Guided SSL)	**0.805**	**5.72**	**329**	**135**

Issue-Guided SSL reaches 0.805 utility versus 0.801 for Fixed-Window SSL at $g = 10$, while reducing Educator tokens by 61.7% and total acquisition tokens by 16.9% (6.89M to 5.72M per run). Teaching efficiency is 2.75 times that of Fixed-Window SSL at $g = 10$. The fixed $g = 5$ and $g = 20$ schedules reach 0.776 and 0.776, respectively.

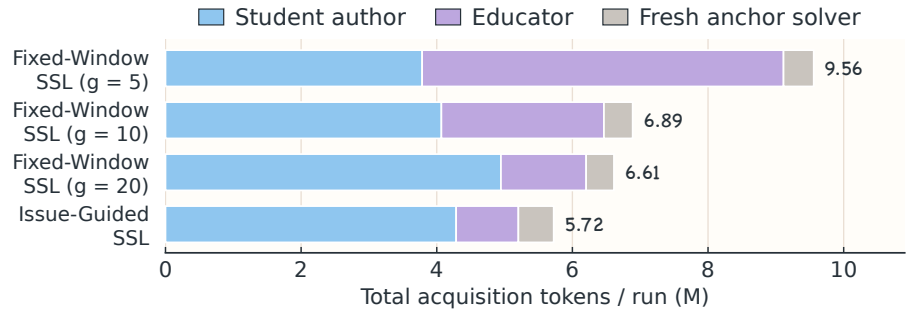


Figure 7: **Acquisition-cost allocation across the four teaching schedules.** Qwen, 22 families and 66 acquisition runs per schedule. Stacked bars show mean Student-author, Educator, and anchor-evaluation tokens per run, with the total labeled for each schedule.

Fixed-Window SSL uses 158.2M Educator tokens across 630 reviews; Issue-Guided SSL uses 60.6M across 329. The 47.8% reduction in reviews combines with a 26.7% reduction in tokens per review (251.1K to 184.1K), yielding a 61.7% reduction in Educator tokens. These per-review costs include follow-up dialogue. Issue-Guided SSL combines issue-triggered review with local context and its follow-up policy; the protocol settings are detailed in Appendix C.2.

Table 18 puts reductions in Educator tokens and changed learner work on the same scale, in millions of tokens per run. Positive net savings mean fewer acquisition tokens within the recorded scope. Student tokens fall 16.1% on Kimi but rise 39.2% on DeepSeek, where the additional Student work exceeds the Educator saving. These outcomes distinguish the consistent reduction in external supervision from the model-dependent change in total acquisition cost.

On DeepSeek, 52.8M additional Student tokens arise mainly from more calls (890 to 1,216, +36.6%), concentrated in the first two rounds. `sqlite-db-truncate` adds 23.8M total acquisition tokens; the other families save 12.8M, leaving a net increase of 11.0M. The 41.8M Educator saving combines fewer reviews (195 to 69) with smaller ones. Fixed-Window SSL has 25 Educator calls re-reading over 500K cached tokens (maximum 799K); its top 15% of calls consume 35% of Educator tokens. Issue-Guided SSL has no calls above 500K cached tokens.

Table 18: Role-level token changes and net acquisition savings for Issue-Guided SSL relative to Fixed-Window SSL, in millions of tokens per run. Qwen uses the 22 main-cohort families (3 repeats); DeepSeek and Kimi use the 10 shared families (one repeat). DeepSeek and Kimi rows cover the persistent author and Educator sessions.

Pair	Runs	Educator saved	Author added	Anchor added	Net saved
Qwen	66	1.479	0.218	0.097	1.164
DeepSeek	10	4.175	5.276	–	−1.101
Kimi	10	3.250	−0.870	–	4.120

Table 19: Per-run acquisition tokens by session role for Fixed-Window SSL and Issue-Guided SSL for DeepSeek and Kimi on the 10-family lite subset (single repeat), covering the persistent author (Student) and Educator sessions.

Pair	Schedule	Student (M/run)	Educator (M/run)	Total (M/run)
DeepSeek	FIXED-WINDOW SSL	13.46	6.55	20.01
	ISSUE-GUIDED SSL	18.74	2.37	21.11
Kimi	FIXED-WINDOW SSL	5.41	4.20	9.61
	ISSUE-GUIDED SSL	4.54	0.95	5.49

F.2 ISSUE LIFECYCLES

Table 20 distinguishes persistent nodes from repeated transition events. A closure ends a review episode and need not be the issue's last appearance. Most closures record Educator silence; verification and bounded-scope dispositions bypass another review in 9 events. All 28 reopen events lead to another lesson. The Flask sequence in Appendix G.2 illustrates this reopening behavior at the event level.

Table 20: Qwen Issue-Guided SSL lifecycle accounting across 22 families and 66 acquisition runs, listing persistent issues, reopen events, closure dispositions, and issue status at run end.

Quantity	Count or summary
Persistent issue nodes	177
Issues/run: mean; median; range	2.68; 3; 1–6
Nodes reopened at least once	23/177 (13.0%)
Reopen events	28
Reopen events followed by a new lesson	28/28
Review closures: Educator silence	194
Verification-backed closures	7
Bounded-scope closures	2
Evidence-backed rejection closures	0
Nodes closed by run end	175
Nodes still pending at run end	2

In an API-scheduling no-lesson example, prior anchor feedback shows the solver searching the wrong script location and reimplementing the scheduler. The Student adds execution and Quick Start guidance to `SKILL.md`; review closes as `teacher-silence`.

G CASE EVIDENCE AND FROZEN-PACKAGE USE

G.1 TESTING AND ADAPTING EDUCATOR GUIDANCE

The HTML-finance case comes from Qwen FIXED-WINDOW SSL, repeat 1, in the main 22-family cohort. It follows a teaching proposal through Student execution, package revision, and fresh use (Table 3).

From a proposal to a checked procedure. The Educator recommends deriving reported financial values from other entries and distinguishing agreement, disagreement, and the absence of an applicable rule. It also suggests generic arithmetic search. The Student implements and runs this strategy on the anchor. The resulting verifier flags 4 metrics and selects line 140, although the incorrect entry is the gross margin at line 210. Inspecting the output, the Student identifies missing financial derivations and replaces generic arithmetic with relations appropriate to each metric, including reverse derivations. Running the revised verifier on the same report leaves only line 210 flagged: the reported gross margin is 60.8%, while $520.4/842.0 \times 100$ rounds to 61.8%. The contrast records two executions during acquisition.

Retained knowledge and fresh use. The final `SKILL.md` retains three outcomes—confirmed, contradicted, and unchecked—and instructs solvers to use both forward and reverse financial relations. It bundles the revised `verify.py`. On held-out `v14`, a fresh solver reads both resources and checks relations in a cash-flow statement. Operating, investing, and financing subtotals sum to 8,000, whereas the reported net increase is 6,270. The closing cash balance independently agrees with 8,000. The solver submits `4206,6270`, identifying the erroneous line and its reported value, and passes 4/4 checks.

Source records. The released run is `traj/qwen-22/fixed-window-ssl/r1/html-finance-verify/`. Within it, `skill-formation/student-author.jsonl` contains the lesson and two executions (records 20 and 42–44); `heldout/v14.jsonl` contains package reads, accounting checks, and submission (records 4, 10, 12, and 14). Scores for all six held-out cases are in `meta.json`. Record numbers refer to the JSONL `i` field.

G.2 FLASK ISSUE REOPENING: EVENT-LEVEL EVIDENCE

In a Qwen Issue-Guided SSL Flask run, `issue-1` receives a lesson, revision, and no-lesson closure during the first round. Fresh-solver anchor feedback with 6/9 checks and missing-table errors reopens it in the next round. The renewed lesson identifies an import-bound database path: a changed working directory can redirect the service to the wrong database. The Student reproduces the mismatch, resolves paths at request time, and reports 14/14 local checks before verification-backed closure. Separately, fresh-solver anchor utility is 0.667, 1.000, and 1.000 across rounds. The released run is `traj/qwen-22/issue-guided-ssl/r3/flask/`. In `skill-formation/student-author.jsonl`, records 50 and 52 show the anchor feedback and renewed guidance; records 80–82 show verification and the revision addressing the issue.

H LIMITATIONS

Our study evaluates complete Skill-learning protocols under a shared single-anchor setting. Further controlled comparisons could isolate how much the Educator, review timing, and issue structure each contribute to transfer performance and teaching cost.