CAN ADAPTATION SUBSTITUTE FOR MODEL SCALE? AN EMPIRICAL STUDY OF LANGUAGE MODEL CAPABILITY GAPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Smaller language models offer a practical deployment option, and a range of adaptation methods can narrow their performance gaps relative to larger models. However, choosing a method for a particular workload requires more than knowing its best attainable score. We present an empirical study across 16 generalization and discovery benchmarks, 8B and 27B targets, and five strategies spanning prompting, harness optimization, and weight updates. We compare method outcomes and score margins, separating heterogeneous endpoint inventories from gains against corresponding unadapted controls. The evaluated methods exhibit both large performance differences and near ties, and their ordering can change with the target model. Behavioral analyses reveal benefits concentrated in particular task strata and changes accompanying reduced output failures. Cost decomposition relates method choice to quality requirements, adaptation expenditure, and reusable supervision. We propose an experience-guided mechanism that prioritizes feasible probes under a budget and selects by target validation. Replay of its validation gate on four additional benchmarks reveals both successful choices and ranking errors. Code and experimental results are available at `https://anonymous.4open.science/r/adaptation-study-57D6/`.

1 INTRODUCTION

Scaling language models has expanded their capabilities (Kaplan et al., 2020; Brown et al., 2020), but deploying a larger model also places demands on memory and serving resources (Kwon et al., 2023). Smaller models offer lower weight-memory requirements and local deployment options when data must remain on site. These benefits motivate our study of whether workload-specific evidence helps adaptation meet practical quality and cost requirements. For a particular workload, adaptation offers another investment: improve a smaller model through examples, an optimized harness, or weight updates. The practical question is therefore *which adaptation method should be used for which workload?* A strong result from one method on one benchmark does not determine its value on another workload or target model. Choosing an intervention requires evidence about relative performance, the task characteristics associated with its benefits, and its cost.

We conduct an empirical study to answer three key questions. **Q1: How do adaptation methods compare across workloads and target models?** The size of a performance difference matters alongside the ordering, including whether a simple method is already sufficient. **Q2: Which workload characteristics help explain their relative effectiveness?** Benchmark averages can conceal different constraints, execution failures, and task strata that affect the value of an intervention. **Q3: Which method meets the required quality at the lowest cost?** We compare adaptation expenditure at a required quality and examine how reusable supervision changes the additional investment.

Our evaluation spans 16 benchmarks: 13 generalization tasks and three discovery tasks. We study Qwen3-8B and Qwen3.6/3.8-27B targets (Qwen Team, 2025; 2026a;b), with Kimi K3 and DeepSeek-V4-Pro as large-model references (Moonshot AI, 2026; DeepSeek-AI, 2026). Five strategies provide distinct interventions: few-shot prompting supplies examples (Brown et al., 2020); SkillOpt optimizes the external harness (Yang et al., 2026); supervised fine-tuning and reinforcement learning update

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.

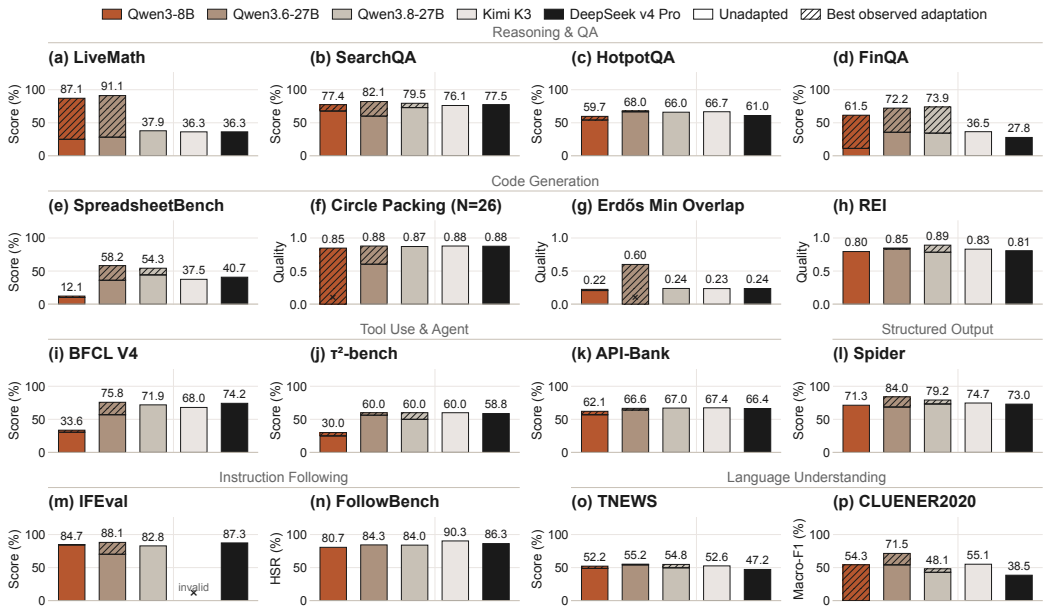


Figure 1: Initial performance and best observed outcomes across 16 benchmarks, with colors identifying models and hatched extensions showing best observed adaptation outcomes above their initial scores; large models are unadapted references. Each small-model–benchmark pair independently selects the best result from its baseline and available adaptation strategies, so selected strategies may differ across bars.

weights using examples and rewards (Hu et al., 2022; Shao et al., 2024); and test-time training adapts during solution search (Yuksekgonul et al., 2026). We evaluate applicable methods, separate heterogeneous endpoint inventories from matched-control gains, and distinguish adaptation expenditure from independent final evaluation.

Figure 1 summarizes the final performance outcomes across all 16 benchmarks, comparing unadapted models with their best observed outcomes across the evaluated adaptation strategies. For each model–benchmark pair, we independently select the highest score among the unadapted baseline and all available adaptation results in the plotted metric; the selected strategy can therefore differ across models and benchmarks. For example, the best observed adapted 27B results exceed the evaluated large-model references on SpreadsheetBench (Ma et al., 2024) and BFCL (Patil et al., 2025), whereas the best observed adapted 8B results remain substantially below them. These outcomes illustrate both the promise and limits of adaptation, but leave the practical selection problem open: how should a method be chosen before its test performance is known? We therefore examine the individual methods behind these best outcomes, the workload characteristics associated with their differences, and the cost of obtaining sufficient quality.

Our contributions are threefold:

- **A comparative study of adaptation methods across workloads.** Across 16 benchmarks, 8B and 27B targets, and five strategies, we analyze performance differences, the competitiveness of simple baselines, and changes in method ordering across target models. We distinguish observed endpoint comparisons from gains supported by matched controls (Section 3).
- **Evidence connecting method outcomes to workload structure and cost.** We identify contrasting adaptation responses across tasks and target models, and examine rule-level repairability, workload composition, and supervision availability as candidate factors for method selection. We decompose recorded costs to examine how quality requirements and reusable supervision affect adaptation expenditure (Sections 4–5).

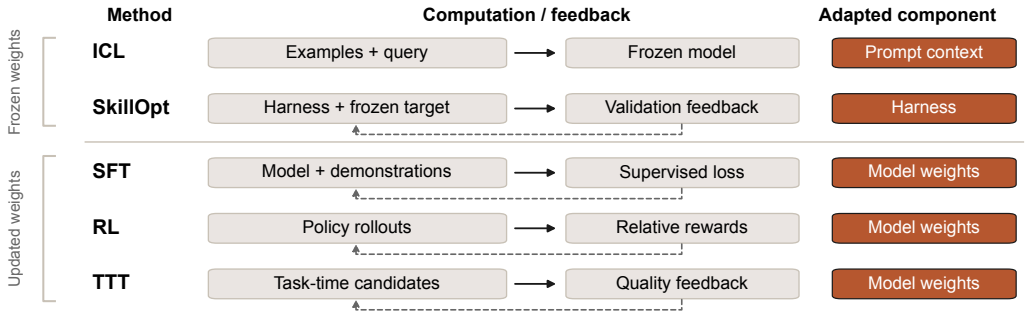


Figure 2: The five adaptation strategies grouped by whether model weights change; orange labels identify the adapted component. Solid arrows show forward computation and dashed arrows show optimization feedback in the evaluated implementations.

- **Experience-guided method selection.** We propose a policy for unseen workloads that filters methods, prioritizes budget-limited probes using compatible history and diagnostic similarity, and freezes a validation-selected candidate at a quality or budget threshold. We evaluate its validation gate by replay on four benchmarks outside the original 16, finding both successful choices and ranking errors (Section 6).

2 Study Design

2.1 Tasks and models

We evaluate 16 benchmarks across reasoning and question answering (LiveMath, SearchQA, HotpotQA, FinQA) (He et al., 2026; Dunn et al., 2017; Yang et al., 2018; Chen et al., 2021), code and structured generation (SpreadsheetBench, Spider) (Ma et al., 2024; Yu et al., 2018), tool use (BFCL, τ^2-bench, API-Bank) (Patil et al., 2025; Barres et al., 2025; Li et al., 2023b), instruction following (IFEval, FollowBench) (Zhou et al., 2023; Jiang et al., 2024), language understanding (TNEWS, CLUENER2020) (Xu et al., 2020a;b), and discovery: Circle Packing and Erdős Minimum Overlap (Yuksekgonul et al., 2026), plus the regular-expression inference (REI) task suite provided by RL-x-SkillOpt (stueam, 2026). Thirteen tasks measure held-out generalization; three measure discovery. Discovery separates feasibility (*Gate*) and normalized *Quality*; we do not average heterogeneous metrics. Evaluator profiles appear in Appendix A.

The target models are Qwen3-8B, Qwen3.6-27B, and Qwen3.8-27B (Qwen Team, 2025; 2026a;b). Version and inference configuration are part of model identity, so these comparisons do not isolate parameter count. Kimi K3 and DeepSeek v4 Pro provide unadapted large-model references (Moonshot AI, 2026; DeepSeek-AI, 2026), without an equivalent adaptation search.

2.2 Adaptation strategies

We evaluate five adaptation strategies: in-context learning (ICL), skill optimization (SkillOpt), supervised fine-tuning (SFT), reinforcement learning (RL), and test-time training (TTT) (Figure 2). ICL and SkillOpt leave model weights unchanged, whereas SFT, RL, and TTT modify them. The following descriptions introduce each approach and its implementation in our study.

In-context learning (ICL). ICL conditions a frozen language model on demonstrations included in its input prompt (Brown et al., 2020). These demonstrations may contain intermediate reasoning, as in chain-of-thought prompting (Wei et al., 2022b). Their number, selection, and ordering are central design choices in recent work (Agarwal et al., 2024; Bertsch et al., 2025; Wang et al., 2025; Akula et al., 2025; Chung et al., 2026). Our evaluated configuration uses three demonstrations drawn exclusively from the training split where available. The Circle Packing configuration with external reference examples is treated separately as a diagnostic.

Skill optimization (SkillOpt). Harness optimization adapts the instructions or execution procedures around a frozen model using task feedback. Related approaches include instruction optimization in OPRO (Yang et al., 2024) and pipeline optimization in DSPy (Khattab et al., 2023). Our implementation uses SkillOpt (Yang et al., 2026), which iteratively revises a reusable skill artifact and selects it using validation performance. DeepSeek v4 Pro is the fixed optimizer for the Q1 comparison; the API-Bank diagnostic in Q2 uses Kimi K3. The overview selects the best observed outcomes across optimizer settings.

Supervised fine-tuning (SFT). SFT adapts model weights to supervised training examples, as in instruction tuning (Wei et al., 2022a) and teacher-guided distillation (Hsieh et al., 2023). We obtain teacher outputs from Kimi K3 with low reasoning effort on training examples and fit low-rank adapters using LoRA (Hu et al., 2022) on Tinker (Thinking Machines Lab, 2026). Gains are measured against unadapted controls with the same backend and final-evaluation profile.

Reinforcement learning (RL). RL adapts a model's output distribution using reward feedback, which can represent human preferences (Ouyang et al., 2022) or task-specific correctness (Shao et al., 2024; DeepSeek-AI et al., 2025). We use group-relative policy optimization (GRPO) with LoRA. Each experimental recipe retains its specified reward function, checkpoint-selection rule, and evaluation scope.

Test-time training (TTT). TTT adapts model weights using feedback available at test time, rather than completing all weight updates before deployment (Sun et al., 2020; Hardt & Sun, 2024). We use TTT-Discover (Yuksekgonul et al., 2026), in which candidate evaluations provide feedback for adaptation during solution search. The search state, budget, and final evaluation procedure define each run. Evaluation by archive replay is reported separately from fresh sampling with a frozen checkpoint.

2.3 Evaluation protocol

We use training data for adaptation and, where specified, validation data for skill or checkpoint selection, then report held-out test performance. Generalization tasks use native benchmark metrics; discovery tasks measure feasibility and quality under the specified search budget and feedback access. We distinguish observed endpoints from gains against compatible unadapted controls. We distinguish adaptation expenditure from independent final evaluation, retaining reported charges, estimates, and bounds on bundled costs. These service expenses do not measure local hardware costs. Task-specific splits, selection rules, budgets, and exceptions appear in Appendix A.

3 Q1: How Do Methods Compare Across Workloads and Target Models?

Figure 3 compares the individual method outcomes behind Figure 1's best-result envelope. SFT uses Kimi K3 teacher demonstrations; SkillOpt uses DeepSeek v4 Pro as the fixed optimizer for this comparison. We examine the distribution of best-performing methods, their score margins, and the additional performance available from a broader method set. The counts below describe the currently observed endpoints, including the marked version and protocol differences, rather than isolated method effects. Comparisons use reported eligible outcomes; discovery Quality is analyzed separately from percentage scores. Matched controls appear in Appendix B.

Insight 1: The distribution of best-performing methods changes with the target model. For the 27B configurations, ICL or SkillOpt attains the highest reported score on 11 of the 13 generalization benchmarks; SFT leads on the other two. For 8B, the highest scores are distributed across ICL/SkillOpt on five tasks, SFT on six, and RL on two. Restricting the comparison to the 12 tasks with ICL, SkillOpt, and SFT results preserves this contrast: the better of ICL and SkillOpt matches or exceeds SFT on 10 tasks for 27B, compared with five for 8B. Thus, the observed concentration of best-performing methods depends on the target, rather than defining a single ordering that transfers unchanged between targets.

	Qwen3-8B					Qwen3.6-27B				
	ICL	SkillOpt	SFT	RL	TTT	ICL	SkillOpt	SFT	RL	TTT
LiveMath	51.6	51.6	**87.1**	—	·	61.3	57.3	**91.1**	—	·
SearchQA	71.5	73.6	**77.4**	58.0†	·	64.8	**82.1**	80.3	79.5†	·
HotpotQA	57.1	55.3	58.3	**59.7†**	·	64.1	67.6	**67.7**	65.7†	·
FinQA	26.3	46.3	42.2	**61.5**	·	53.7	**71.2**	45.7	34.2	·
SpreadsheetBench	10.4	**12.1**	10.4	—	·	46.4	**55.7**	33.9	—	·
BFCL V4	**32.0**	28.9	7.8	—	·	68.8	**75.8**	46.9	—	·
τ²-bench	12.5	25.0	**30.0**	—	·	**55.0**	50.0	12.5	—	·
API-Bank	53.9	61.0	**62.1**	43.6	·	65.6	**66.6**	65.5	48.6	·
Spider	58.8	63.7	**70.6**	65.3†	·	81.2	**82.9**	82.1	78.3†	·
IFEval	**84.7**	82.8	79.2	84.4	·	86.8	**88.1**	83.1	81.3	·
FollowBench	44.5	**48.9**	43.6	×	·	60.7	**67.8**	60.4	—	·
TNEWS	47.4	44.0	**52.2**	—	·	54.2	**55.2**	54.6	—	·
CLUENER2020	**54.3**	47.0	×	45.5†	·	69.4	**71.5**	×	48.1†	·
Circle Packing	×	**0.85**	·	·	0.82†	×	0.69	·	·	**0.88†**
Erdős overlap	0.00	0.17	·	·	**0.21†**	0.24	**0.60**	·	·	0.24†
REI	**0.57**	0.43	·	·	0.53	0.82	0.77	·	·	**0.85**

Figure 3: Performance comparison of five adaptation methods across 16 benchmarks and 8B/27B targets. Each cell reports the final task score after adaptation, not the gain over baseline: benchmark-specific percentages for generalization tasks and normalized Quality for discovery tasks. Orange outlines mark the highest reported score within each benchmark and target block, including ties. Hatching denotes Qwen3.8 RL, † protocol differences, and ·/—/× unassigned/missing/excluded entries.

Insight 2: Winning margins distinguish decisive advantages from near ties. On LiveMath, SFT leads the next highest available method by 35.49 pp for 8B (87.10 versus 51.61) and 29.84 pp for 27B (91.13 versus 61.29). In contrast, the 27B HotpotQA scores of SFT and SkillOpt differ by only 0.14 pp (67.71 versus 67.57), while the ICL, SkillOpt, and SFT scores on 27B API-Bank span just 1.15 pp (65.47–66.62). Counting each as a method win would conceal the substantial difference in performance at stake. These small point-estimate differences do not establish superiority or statistical equivalence without repeated evaluation.

Insight 3: A method with few wins can still provide a large, concentrated benefit. Relative to the better of ICL and SkillOpt, adding weight-update methods to the observed 27B candidate set raises the best score by 29.84 pp on LiveMath and only 0.14 pp on HotpotQA, with no increase on the other 11 generalization tasks. For 8B, the additional best-score margins are spread across more tasks, including 35.49 pp on LiveMath, 15.17 pp on FinQA, and 6.90 pp on Spider. A method's contribution to the available performance envelope therefore cannot be assessed from its win count alone. These retrospective maxima describe the reported candidates under their stated protocols; they are not outcomes of a prospective method selector.

Together, these results show where method choice matters and how much performance is at stake, but the rankings alone do not indicate which method to try on a new workload. We next examine rule-level repairability, workload composition, and supervision availability to identify measurable evidence beyond aggregate rankings.

4 Q2: Which Properties Can Guide Adaptation Choice?

We identify three candidate factors from output traces, subtask results, and teacher demonstrations: **(1) Error repairability**, the fraction of errors a targeted correction fixes relative to the successes it damages; **(2) Workload composition**, the mix of subtasks with different method advantages for a given target model; and **(3) Supervision availability**, the quantity and coverage of successful demonstrations. Figure 4 presents the evidence; measurement details appear in Appendix C.1. Section 6 proposes a selection mechanism and evaluates its validation gate.

Diagnostic 1: Repair coverage and regression risk indicate opportunities for targeted correction. A low baseline score indicates room for improvement but does not identify an effective intervention;

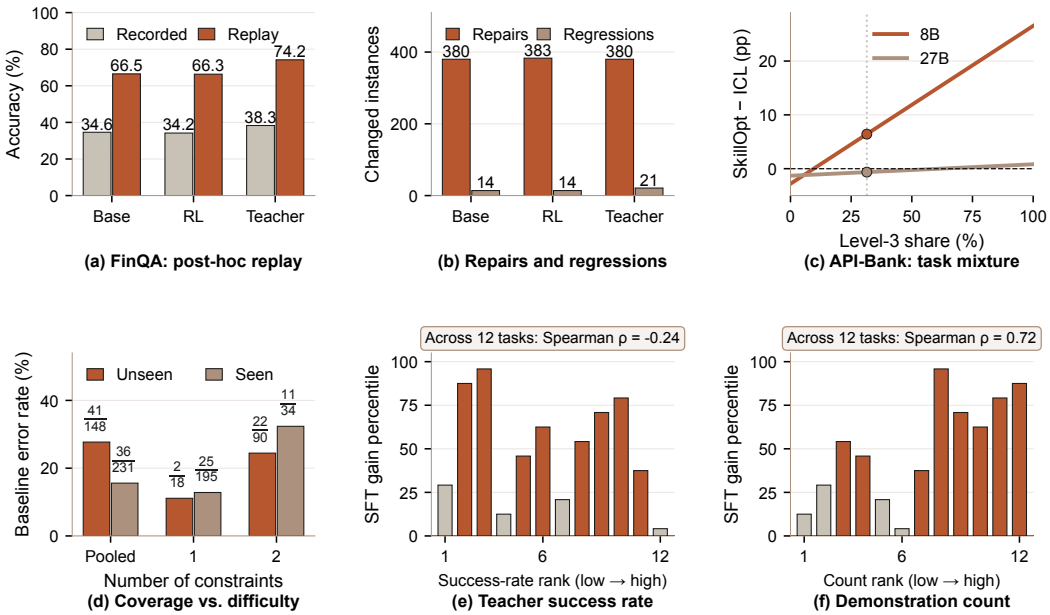


Figure 4: Evidence for three adaptation diagnostics: error repairability (a,b), workload composition (c), and supervision availability, with a coverage check (d–f). Panels (a,b) replay FinQA outputs from the 27B baseline, RL, and K3 teacher post hoc, (c) reweights API-Bank scores, (d) shows IFEval error rates with error/total counts, and (e,f) show percentile ranks of mean SFT gains across two targets, ordered by teacher success rate or demonstration count (gray: negative original gain); boxes report correlations across 12 tasks.

baseline score alone fails to improve leave-one-task-out SFT gain prediction over a constant for either target (Appendix C.1). We instead measure how many failures a concrete rule repairs and how many successes it damages. On FinQA, removing a terminal multiplication by 100 from eligible programs repairs 380/750 baseline errors while breaking 14/397 successes, raising execution accuracy from 34.61% to 66.52% (Figure 4(a)). Figure 4(b) separates repairs from damage: the same rule repairs 383 errors and loses 14 successes in RL outputs, and repairs 380 while losing 21 in K3 teacher outputs. The repeated pattern indicates a shared output-convention issue across all three sets. The rule selects outputs by program structure without reference answers, but was discovered post hoc; these are replay results, not held-out adaptation scores. Together, the gain in Figure 4(a) and limited damage in Figure 4(b) motivate testing a cheap correction before weight adaptation.

Diagnostic 2: Workload composition and target model jointly determine the observed method preference. An aggregate ranking can obscure opposing subtask responses. With Kimi K3 as optimizer, 8B SkillOpt exceeds ICL by 26.53 pp on API-Bank Level 3 but loses 1.50 and 6.67 pp on Levels 1 and 2. Holding within-level scores and the Level-1:Level-2 ratio fixed, reweighting these observations switches the preferred method at a Level-3 share of 9.6% for 8B, versus 61.6% for 27B (Figure 4(c)); dots mark the observed workload mix, while zero crossings mark changes in method preference. These are conditional crossings, not validated deployment thresholds. The relevant signal is therefore the subtask-specific method advantage weighted by the intended workload mix, measured for the target model being deployed.

Diagnostic 3: Successful demonstration count correlates more strongly with SFT gains than teacher success rate. We use Spearman's rank correlation ($\rho \in [-1, 1]$) to measure whether higher diagnostic values accompany larger SFT gains across tasks; positive values indicate agreement in ranking, not causation. Teacher success rate alone is a weak proxy for downstream learning value: across 12 tasks, its correlation with mean SFT gain is $\rho = -0.24$, compared with $\rho = 0.72$ for successful demonstration count (Figure 4(e) and Figure 4(f)). The count association remains after excluding LiveMath ($\rho = 0.73$), although training exposure and task properties remain confounded. Ranking the mean gains across tasks and ordering them by each indicator makes this contrast

visible: gray bars (negative original gains) remain dispersed in Figure 4(e), but all fall in the lower-count half in Figure 4(f). Figure 4(d) checks whether an apparently useful coverage signal survives difficulty controls: IFEval's higher baseline error rate for unseen constraint combinations (27.70% versus 15.58%) reverses within both the one- and two-constraint strata. This counterexample concerns baseline errors, not SFT gains; it shows why unadjusted coverage can mislead. Successful demonstration count is a stronger descriptive correlate of SFT gains than teacher success rate in this study; the association does not establish a causal effect or a validated selection rule.

Together, these diagnostics suggest which interventions merit testing for a workload. Once multiple strategies meet the required quality, the remaining choice is economic: how much additional investment does obtaining each adapted system require? Q3 separates this investment from final-test expenses and compares candidates at a required quality.

5 Q3: Which Adaptation Strategy Costs Less at a Required Quality?

We compare the additional investment required to obtain an adapted system:

$$C_{\text{adapt}} = C_{\text{new data}} + C_{\text{optimization}} + C_{\text{training}} + C_{\text{validation}}. \tag{1}$$

Optimization includes search and rollout generation; validation includes skill and checkpoint selection. Independent final-test execution is excluded from C_{adapt}. Figure 5 provides an overview of 8B performance against recorded experiment costs across 16 tasks, including final evaluation; we separate that expense for the adaptation-cost analysis below. The 27B overview is in Appendix D.2.

We use Codex (OpenAI, 2026) as the execution harness. Our experiments use paid online services: SFT and RL build on the official Tinker Cookbook, while TTT uses the reference TTT-Discover implementation on Tinker (Thinking Machines Lab, 2026; Yuksekgonul et al., 2026).[1] We estimate total service expenditure for the experimental campaign at approximately $5,000. This campaign-wide amount is distinct from the per-run cost accounting below, which retains estimates and accounting exclusions.

Bundled charges define an adaptation-cost interval: separately identified non-test expenses give the lower endpoint, and including the full bundled charge gives the upper endpoint. These intervals support the numerical comparisons below; Figure 5 retains recorded totals. ICL reuses fixed demonstrations and incurs zero new preparation API charge under Equation 1; its plotted cost includes evaluation. Figure 6 separates final evaluation from other recorded expenses.

Finding 1: Weight updates need not cost more than harness optimization. Figure 5 shows this directly on 8B Spider: SFT reaches 70.60% at a recorded total cost of $1.135, compared with SkillOpt's 63.70% at $1.876. SFT therefore costs 39.5% less and scores 6.90 percentage points higher. The ordering also holds after separating final evaluation: SFT adaptation costs are bounded by $0.774–$1.135, including teacher preparation, versus SkillOpt's $1.641. Even the SFT upper bound is 30.8% lower (Appendix D.2). With 197 accepted demonstrations and three LoRA epochs (75 steps), SFT provides a lower-cost, higher-scoring option than the evaluated SkillOpt configuration, which uses 16 search steps with target execution and validation. This result highlights small-data fine-tuning as a cost-competitive adaptation option: spending decisions should account for supervision volume and search budget, rather than treating weight updates as inherently more expensive.

Finding 2: Existing supervision can reverse the cost ordering at a quality target. Figure 6(c) shows how separating preparation expenses changes the comparison when identical teacher data already exist. For 8B API-Bank, the SFT breakdown in Table 11 assigns $2.385 to teacher data and $0.237 to the remaining recipe. From scratch, its adaptation cost is $2.385–$2.621, above SkillOpt's $0.890. If the identical teacher data already exist, SFT's additional adaptation expense is at most $0.237, reversing that ordering. SFT and SkillOpt score 62.13% and 60.98%, respectively, so both meet an illustrative 60% requirement; ICL at 53.92% does not. This reuse scenario is derived from the cost components, rather than plotted as a separate run in Figure 5. It shows why a spending decision should account for existing supervision as well as the required quality.

[1] Tinker provides hosted model training and sampling: `https://tinker-docs.thinkingmachines.ai/`. Base-model and harness inference primarily uses the OpenRouter API (`https://openrouter.ai/`); adapted checkpoints are sampled through Tinker.

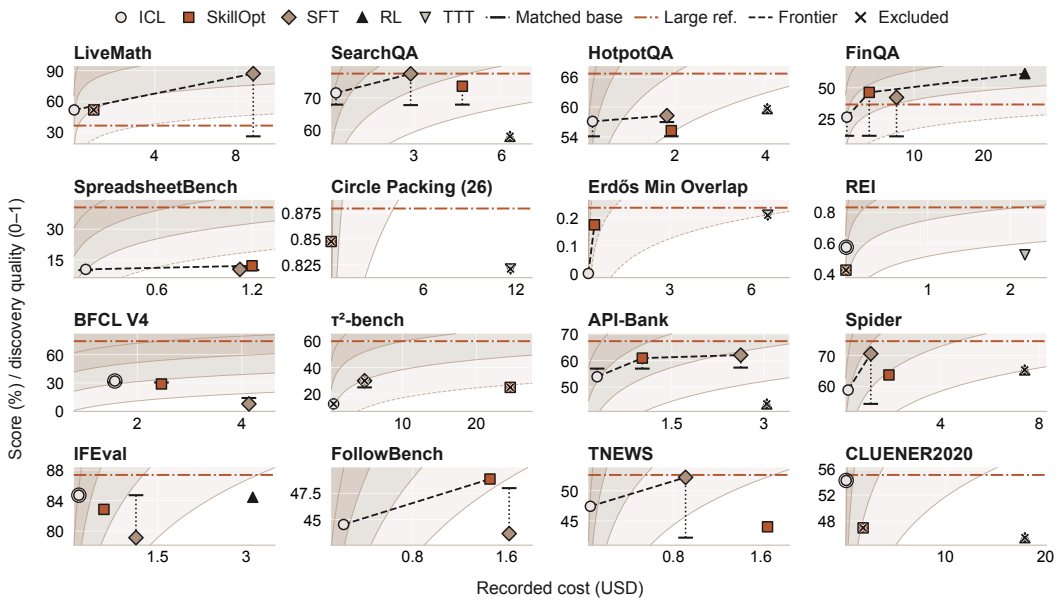


Figure 5: Qwen3-8B cost–performance observations. Dotted stems link matched-base ticks to adapted scores; ticks imply no baseline cost. Dashed lines mark eligible frontiers, crosses exclusions, and curves preferences. Score ranges vary by task (Appendix D.2).

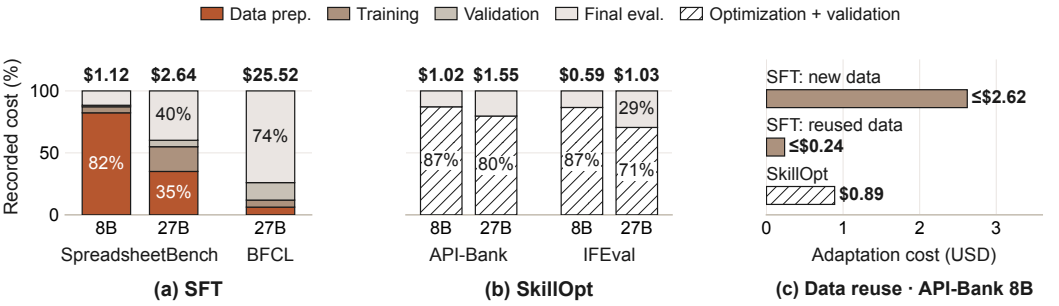
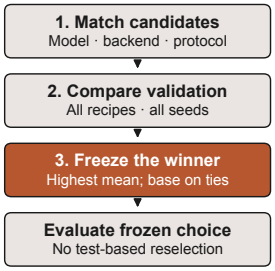


Figure 6: Cost composition for (a) SFT and (b) SkillOpt: bars show percentages, labels give total USD, and hatching combines optimization and validation. (c) Reusing identical teacher data reverses the API-Bank 8B cost ordering; SFT bars show adaptation-cost upper bounds, excluding the sunk data expense in the reuse scenario (Appendix D.3).

Both findings describe the evaluated recipes under recorded service prices and stage-wise accounting. Appendix D specifies the comparison scope and cost definitions. Section 6 incorporates these quality and budget constraints into method selection.

6 An Experience-Guided Selection Mechanism

Our method. We propose a sequential mechanism for an unseen workload w, given a target model f, an exploration budget B, and an optional quality requirement q. A baseline probe on permitted development examples yields a diagnostic profile z_w: observable error types, subtask proportions, and available supervision. These Q2 factors describe the workload without consulting test outcomes. Data access, feedback, and weight/harness access define the eligible recipe set $\mathcal{M}_w$; baseline retention remains an option.



(a) Selection gate

★ Best candidate Validation choice

		Base	ICL	SkillOpt	SFT	RL	Selected	Gap
BIRD	8B	37.0	36.6	34.2	44.2★	36.8	44.2	0.00
	27B	61.2	60.2	58.6	60.8	61.4★	61.4	0.00
MuSR	8B	56.2	59.4	68.4	62.9★	54.3	54.3	8.59
	27B	55.9	55.9	73.0	76.6★	57.0	76.6	0.00
CRUX-O	8B	51.2★	50.6	50.8	50.8	50.9	51.2	0.00
	27B	72.0	71.7	70.8	69.2	91.8★	91.8	0.00
IFBench†	8B	74.0	76.3	81.3★	73.3	73.3	81.3	0.00
	27B	82.0	77.3	84.7★	80.7	83.0	82.3‡	2.33

(b) Candidate accuracy (%) and selection gap (pp)

Figure 7: Orange outlines mark selections; stars mark eligible best scores; Gap is best minus selected accuracy (pp); gray scores are references excluded from selection and Gap. † IFBench uses cross-fitted validation; ‡ 27B mixes SkillOpt/RL (dashed outlines); its gap uses the best fixed recipe. Protocols: Appendix E.2.

Compatible historical records $\mathcal{H}$ prioritize which recipe to probe. For recipe m, we propose a similarity-weighted validation-gain score

$$p_w(m) = \frac{\sum_{h \in \mathcal{H}_m} \omega(z_w, z_h) \Delta_h(m)}{\sum_{h \in \mathcal{H}_m} \omega(z_w, z_h)}, \tag{2}$$

Here $\Delta_h(m)$ is seed-mean gain over the matching baseline; ω is Jaccard tag overlap. Unsupported recipes follow supported ones by quoted cost, with fixed-name ties.

Initialize $\mathcal{O}_0$ with baseline validation. At step t, probe the highest-priority untried recipe with $c_t + b_m \leq B$: c_t includes diagnostic and previous probe costs; b_m caps the next probe. Record validation gains, regressions, and actual costs. Failed probes consume budget but cannot enter the completed set $\mathcal{O}_t$. The quality gate chooses $\hat{m}_t = \arg\max_{m \in \mathcal{O}_t} \bar{v}_w(m)$ using seed-mean validation accuracy, with baseline-first then fixed-name ties. Stop when $\bar{v}_w(\hat{m}_t) \geq q$ or no further probe fits; without q, exhaust the available budget or recipes. Freeze the choice before independent evaluation (Figure 7(a)). The evaluation below isolates this final gate of the proposed policy.

Evaluation. We evaluate on **four new benchmarks unseen in our preceding 16-benchmark study**: BIRD (Li et al., 2023a), MuSR (Sprague et al., 2024), CRUXEval-O (Gu et al., 2024), and IFBench (Pyatkin et al., 2025). The gate uses target validation: seed-17 SFT/RL for BIRD/MuSR, three-seed means for CRUXEval-O, and five-fold cross-fitting for IFBench. Gray references are excluded from selection and Gap because validation or backend differs (Appendix E.2). The replay tests candidate choice, not exploration efficiency.

Results. Selection matches the eligible test-best on BIRD, CRUXEval-O, and MuSR 27B. MuSR 8B selects RL (54.30%) over SFT (62.89%), an 8.59-point gap; the SkillOpt reference (68.36%) exceeds both eligible candidates. IFBench 8B selects SkillOpt in all five folds (81.33%). The 27B gate selects SkillOpt in four folds and RL in one, scoring 82.33%, 2.33 points below fixed SkillOpt. Validation can misrank candidates; coverage and protocol bound these results.

7 Conclusion

In this paper, we explore whether adaptation can substitute for model scale and how to choose a strategy for a workload. Across five strategies, 16 benchmarks, and 8B/27B targets, adapted smaller models can compete with larger-model references on some workloads, but this substitution depends on the task and target. Trace analyses identify repairability, workload composition, and supervision availability as candidate diagnostics. Small-data SFT can compete with harness optimization on cost, and reusable supervision can reverse cost orderings. These findings motivate an experience-guided selection mechanism. Replaying its validation gate on four additional benchmarks yields both successful choices and ranking errors, underscoring the need to check historical guidance against target validation.

REFERENCES

Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. Many-shot in-context learning. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2404.11018.

Arjun R. Akula, Kazuma Hashimoto, Krishna Srinivasan, Aditi Chaudhary, Karthik Raman, and Michael Bendersky. Refract ICL: Rethinking example selection in the era of million-token models, 2025. URL https://arxiv.org/abs/2506.12346.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

Amanda Bertsch, Maor Ivgi, Emily Xiao, Uri Alon, Jonathan Berant, Matthew R. Gormley, and Graham Neubig. In-context learning with long-context models: An in-depth exploration. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2025. URL https://arxiv.org/abs/2405.00200.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2005.14165.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711, 2021. doi: 10.18653/v1/2021.emnlp-main.300. URL https://aclanthology.org/2021.emnlp-main.300/.

Tsz Ting Chung, Lemao Liu, Mo Yu, and Dit-Yan Yeung. Many-shot CoT-ICL: Making in-context learning truly learn. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2605.13511.

DeepSeek-AI. DeepSeek-V4-Pro: Official model card, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro. Accessed September 23, 2026.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948v1.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. SearchQA: A new Q&A dataset augmented with context from a search engine, 2017. URL https://arxiv.org/abs/1704.05179.

Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I. Wang. CRUXEval: A benchmark for code reasoning, understanding and execution. *arXiv preprint arXiv:2401.03065*, 2024. URL https://arxiv.org/abs/2401.03065.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.18466.

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

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023. URL https://arxiv.org/abs/2305.02301.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. FollowBench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 4667–4688, 2024. doi: 10.18653/v1/2024.acl-long.257. URL https://aclanthology.org/2024.acl-long.257/.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into self-improving pipelines, 2023. URL https://arxiv.org/abs/2310.03714.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023. URL https://arxiv.org/abs/2309.06180.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM already serve as a database interface? a BIg bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing Systems*, volume 36, 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/83fc8fab1710363050bbd1d4b8cc0021-Abstract-Datasets_and_Benchmarks.html.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-Bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3102–3116, 2023b. URL https://aclanthology.org/2023.emnlp-main.187/.

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, 2024. URL https://arxiv.org/abs/2406.14991.

Moonshot AI. Kimi K3: Official model card, 2026. URL https://huggingface.co/moonshotai/Kimi-K3. Accessed September 23, 2026.

OpenAI. Codex CLI. Official documentation, 2026. URL https://learn.chatgpt.com/docs/codex/cli. Accessed September 26, 2026.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2203.02155.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The Berkeley Function Calling Leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2507.02833.

Qwen Team. Qwen3-8B: Official model card, 2025. URL https://huggingface.co/Qwen/Qwen3-8B. Accessed September 23, 2026.

Qwen Team. Qwen3.6-27B: Official model card, 2026a. URL https://huggingface.co/Qwen/Qwen3.6-27B. Accessed September 23, 2026.

Qwen Team. Qwen3.8-27B: Official model card, 2026b. URL https://huggingface.co/Qwen/Qwen3.8-27B. Accessed September 23, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jenyYQzue1.

stueam. RL-x-SkillOpt: Regular expression inference (REI) task definitions and data splits. GitHub software and data repository, 2026. URL https://github.com/stueam/RL-x-SkillOpt/tree/main/benchmarks/rei. Repository owner: stueam. Accessed September 26, 2026.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. URL https://proceedings.mlr.press/v119/sun20b.html.

Thinking Machines Lab. Tinker Cookbook: Post-training with Tinker. Software repository, 2026. URL https://github.com/thinking-machines-lab/tinker-cookbook. Accessed September 22, 2026.

Xubin Wang, Jianfei Wu, Yuan Yichen, Deyu Cai, Mingzhe Li, and Weijia Jia. Demonstration selection for in-context learning via reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/wang25dp.html.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022a. URL https://arxiv.org/abs/2109.01652.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022b. URL https://arxiv.org/abs/2201.11903.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. CLUE: A chinese language understanding evaluation benchmark.

In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4762–4772, 2020a. doi: 10.18653/v1/2020.coling-main.419. URL https://aclanthology.org/2020.coling-main.419/.

Liang Xu, Yu Tong, Qianqian Dong, Yixuan Liao, Cong Yu, Yin Tian, Weitang Liu, Lu Li, Caiquan Liu, and Xuanwei Zhang. CLUENER2020: Fine-grained named entity recognition dataset and benchmark for chinese, 2020b. URL https://arxiv.org/abs/2001.04351.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.03409.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. SkillOpt: Executive strategy for self-evolving agent skills, 2026. URL https://arxiv.org/abs/2605.23904.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3911–3921, 2018. doi: 10.18653/v1/D18-1425. URL https://aclanthology.org/D18-1425/.

Mert Yuksekgonul, Daniel Koceja, Xinhao Li, Federico Bianchi, Jed McCaleb, Xiaolong Wang, Jan Kautz, Yejin Choi, James Zou, Carlos Guestrin, and Yu Sun. Learning to discover at test time, 2026. URL https://arxiv.org/abs/2601.16175.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A Evaluation Protocols and Adaptation Configurations

This appendix records the evaluation restrictions needed to interpret the main figures. Appendices B–D provide supporting evidence for Q1–Q3, followed by the completed validation-gate replay and exact selection paths in Appendix E.

A.1 Task scope and comparability

A comparison is identified by task, exact model version, recipe, backend, evaluation scope, and metric. Table 1 records the restrictions that prevent some endpoints from serving as matched controls. The inventory contains 106 outcomes, 11 missing positions, and five exclusions across 122 assigned positions. Seven 27B RL entries use Qwen3.8; other 27B method entries use Qwen3.6. These are distinct model identities.

Table 1: Task-specific evaluation scope and key restrictions. Counts refer to scored test instances unless candidate counts are stated.

Task	Main scope	Interpretation restriction
LiveMath	124	SkillOpt uses a different final-answer wrapper from the API baseline and ICL.
SearchQA	1,400	Some RL evaluations use 1,000 instances.
HotpotQA	700	Some RL evaluations use 1,000 instances.
FinQA	1,147	Training access and backends vary; answer-only diagnostics are excluded.
SpreadsheetBench	280	One instance lacks test cases; 8B SFT has additional sampler timeouts.
BFCL V4	128	Research multi-turn-base split, not the official leaderboard; controls vary by backend.
τ^2-bench	40 per trial	Baseline averages four trials; adapted results use single trials with a fixed user simulator.
API-Bank	779	Research train/validation subsets; baseline and adaptation worker/retry settings differ.
IFEval	379	Strict prompt correctness; some API controls include rejection or output-budget effects.
FollowBench	300 / 575	HSR on 300 and verifier accuracy on 575 are separate protocols.
TNEWS	500	Baseline and adaptation samples share only 26 normalized titles.
CLUENER2020	300 / 1,000	Evaluation scopes differ; ambiguous SFT metric labels are excluded.
Spider	300 / 1,000	Behavioral comparisons use 1,000 instances and single-database execution.
Circle Packing	8 candidates	External-reference ICL is excluded; mathematical TTT uses archive replay.
Erdős overlap	8 candidates	Qwen3.6 baseline failed before responses; TTT uses archive replay.
REI	10 / 3 tasks	Task subsets and output budgets vary by recipe.

Discovery outcomes. Gate measures feasibility separately from best-valid Quality. Circle Packing Quality is the best valid sum of radii divided by three; Erdős Quality is $\max(0, 1 - C_5/0.5)$. REI aggregates per-task best quality over its stated task set. Its task definitions and train/validation/test splits are provided by the RL-x-SkillOpt repository (stueam, 2026); this citation identifies the software/data source, rather than a benchmark paper. No valid candidate makes best-valid Quality undefined, not zero. Archive replay evaluates retained search candidates and is distinguished from fresh checkpoint sampling. Scores with different metric semantics are not averaged together.

A.2 Adaptation configurations

ICL uses three training-only demonstrations where available; the external-reference Circle Packing diagnostic is excluded from method comparisons. SkillOpt uses DeepSeek v4 Pro in the method matrix and Kimi K3 in the API-Bank workload-composition diagnostic. The best-result overview

may select a different recorded optimizer, so it does not represent one common optimization recipe. SFT uses verified Kimi K3 training outputs and LoRA with rank 32, learning rate 10^{-4}, batch size eight, and three epochs. Table 2 records the differing supervision quantities. RL uses the stated task reward and checkpoint-selection procedure; its paired FinQA and IFEval evaluations are detailed in Appendix C. TTT-Discover is evaluated under the recorded search and terminal-evaluation contracts.

Table 2: SFT training-set sizes and optimization steps for each target model. Source counts teacher training prompts; accepted counts the examples retained for training. The CLUENER recipe is documented although its ambiguous endpoint labeling excludes it from the main gain analysis.

Task	Source	Accepted	Validation	Test	Steps
LiveMath	466	191	18	124	72
SearchQA	400	296	200	1,400	111
HotpotQA	200	163	100	700	63
FinQA	1,000	383	200	1,147	144
SpreadsheetBench	80	28	40	280	12
BFCL V4	40	17	32	128	9
τ^2-bench	54	34	20	40	15
API-Bank	223	100	111	779	39
IFEval	108	104	54	379	39
FollowBench	165	101	80	575	39
TNEWS	400	200	100	500	75
CLUENER2020	286	146	143	1,000	57
Spider	286	197	143	1,000	75

Accepted SFT training sets range from 17 to 383 examples. Shared hyperparameters therefore do not isolate task identity from supervision quantity or quality. Validation and test have separate roles; different output budgets and execution failures remain part of the reported recipes.

A.3 Evidence scope and limitations

Most recipes lack repeated training seeds, and some evaluations have small or dependent samples. Endpoint differences are descriptive unless supported by compatible controls and appropriate uncertainty estimates. Per-instance traces are incomplete across methods; aggregate counts cannot reconstruct repairs and regressions. Paired RL intervals capture item-sampling uncertainty for the observed checkpoints, not training variability. These same observations are reused across figures and tables, rather than constituting independent replications. Neither target-model contrasts nor backend comparisons isolate parameter count or a causal serving effect. Workload-level diagnostics characterize observed associations rather than establish predictive selection rules.

Source records retain model and protocol identity, metric, recipe, cost status, and trace provenance. Figure data and plotting scripts preserve these links; reproducing model executions additionally requires the original evaluators, configurations, and service access. Unknown or incompatible outcomes remain distinct from evaluated failures. Cost assumptions and selection-validation boundaries are specified in their respective appendices.

B Q1: Supporting Method Comparisons and Matched Controls

The main method matrix already reports the available endpoints. Here we retain the controls and within-method comparisons needed to interpret adaptation gains, without duplicating the matrix as bars or rankings.

B.1 Unadapted controls

The following comparisons have matching reported scope and metric, but do not independently establish identical example IDs, rendered inputs, or serving configurations. They are cross-configuration differences, not isolated provider effects.

Table 3: Unadapted controls on tasks with matching reported scope and metric. Entries show OpenRouter → Tinker (percentage scores). Matching scope and metric do not establish paired example identity or equivalence of the full execution configuration.

Task	Qwen3-8B	Qwen3.6-27B
LiveMath	25.00 → 25.81	28.23 → 31.45
SearchQA	67.86 → 67.71	60.07 → 57.57
HotpotQA	54.14 → 57.00	66.71 → 66.71
FinQA	11.25 → 10.81	35.66 → 34.96
SpreadsheetBench	10.00 → 11.07	36.07 → 38.93
BFCL V4	30.47 → 14.06	57.03 → 48.44
API-Bank	57.00 → 57.38	64.06 → 64.18
IFEval	84.43 → 84.70	70.18 → 87.60

On BFCL, the 27B-minus-8B difference changes from +26.56 pp to +34.38 pp between these configurations, using the corrected Tinker tool-template controls. On IFEval, the 27B SFT endpoint of 83.11% is −4.49 pp relative to its training-backend control, rather than +12.93 pp relative to the API reference. These contrasts motivate recipe-specific controls rather than a common correction across tasks.

B.2 SFT RESPONSES UNDER MATCHED CONTROLS

The evaluated SFT recipe improves the same eight tasks and declines on the same four for both targets. Table 4 supplies control scores, endpoints, and gains; it is a descriptive pattern, not a claim of significance or invariance to other training recipes.

Table 4: SFT compared with its same-backend unadapted control. Both models have the same observed gain sign on all 12 included tasks. Differences use the displayed rounded scores and can differ by 0.01 pp from values computed before rounding. No statistical significance is implied.

	Qwen3-8B			Qwen3.6-27B		
Task	Control	SFT	Δ pp	Control	SFT	Δ pp
LiveMath	25.81	87.10	+61.29	31.45	91.13	+59.68
SearchQA	67.71	77.43	+9.72	57.57	80.29	+22.72
HotpotQA	57.00	58.29	+1.29	66.71	67.71	+1.00
FinQA	10.81	42.20	+31.39	34.96	45.68	+10.72
SpreadsheetBench	11.07	10.36	−0.71	38.93	33.93	−5.00
BFCL V4	14.06	7.81	−6.25	48.44	46.88	−1.56
τ^2-bench	25.00	30.00	+5.00	10.00	12.50	+2.50
API-Bank	57.38	62.13	+4.75	64.18	65.47	+1.29
Spider	54.30	70.60	+16.30	81.30	82.10	+0.80
IFEval	84.70	79.16	−5.54	87.60	83.11	−4.49
FollowBench	48.00	43.65	−4.35	63.30	60.35	−2.95
TNEWS	42.20	52.20	+10.00	54.00	54.60	+0.60

SpreadsheetBench's 8B adapted evaluation includes 36 sampler timeouts, versus one in its control. Excluding that task and the small τ^2-bench trial leaves ten matching gain signs. Task-specific training quantities (Table 2) and runtime failures limit a causal explanation of the cross-task pattern.

B.3 ALIGNED QWEN3.8 RL EVALUATIONS

The current method matrix uses aligned Qwen3.8-27B RL reevaluations: FinQA changes from 34.61% to 34.18%, and IFEval from 79.68% to 81.27%. They replace earlier incompatible endpoints rather than maximize over scoring procedures. FinQA's answer-only diagnostic is excluded from its program-based comparison. FollowBench's 300-instance HSR evaluation is separate from its 575-instance verifier evaluation. The paired sample counts and uncertainty behind the RL differences follow in Appendix C.

B.4 COMPLETE ENDPOINT INVENTORY

The following tables disclose the plotted scores rather than rounding them into visual bins. Percentage scores and discovery Quality (0–1) retain their separate scales. Q3, Q3.6, and Q3.8 denote the respective Qwen versions; K3 and DS-v4 denote Kimi K3 and DeepSeek v4 Pro. These are endpoint inventories, not paired adaptation effects. Optimizer suffixes report actual provenance: K3 or DS denotes the SkillOpt optimizer, not the target model. SFT uses K3.

Table 5: Figure 1 numeric inventory. Best is the plotted best observed endpoint, including baseline retention. A dash denotes no reported valid score; it is not zero.

Task	Target	Initial	Best	Selected
LiveMath	Q3-8B	25.00	87.10	SFT
LiveMath	Q3.6-27B	28.23	91.13	SFT
LiveMath	Q3.8-27B	37.90	37.90	Base
LiveMath	K3	36.29	36.29	Base
LiveMath	DS-v4	36.29	36.29	Base
SearchQA	Q3-8B	67.86	77.43	SFT
SearchQA	Q3.6-27B	60.07	82.14	SkillOpt-DS
SearchQA	Q3.8-27B	73.00	79.50	RL
SearchQA	K3	76.14	76.14	Base
SearchQA	DS-v4	77.50	77.50	Base
HotpotQA	Q3-8B	54.14	59.70	RL
HotpotQA	Q3.6-27B	66.71	68.00	SkillOpt-K3
HotpotQA	Q3.8-27B	66.00	66.00	Base
HotpotQA	K3	66.71	66.71	Base
HotpotQA	DS-v4	61.00	61.00	Base
FinQA	Q3-8B	11.25	61.46	RL
FinQA	Q3.6-27B	35.66	72.19	SkillOpt-K3
FinQA	Q3.8-27B	34.35	73.93	SkillOpt-DS
FinQA	K3	36.53	36.53	Base
FinQA	DS-v4	27.81	27.81	Base
SpreadsheetBench	Q3-8B	10.00	12.14	SkillOpt-DS
SpreadsheetBench	Q3.6-27B	36.07	58.21	SkillOpt-K3
SpreadsheetBench	Q3.8-27B	44.29	54.29	SkillOpt-DS
SpreadsheetBench	K3	37.50	37.50	Base
SpreadsheetBench	DS-v4	40.71	40.71	Base
Circle Packing N=26	Q3-8B	–	0.8471	SkillOpt-DS
Circle Packing N=26	Q3.6-27B	0.6035	0.8786	TTT
Circle Packing N=26	Q3.8-27B	0.8713	0.8713	Base
Circle Packing N=26	K3	0.8787	0.8787	Base
Circle Packing N=26	DS-v4	0.8757	0.8757	Base
Erdős Min Overlap	Q3-8B	0.2053	0.2231	SkillOpt-K3
Erdős Min Overlap	Q3.6-27B	–	0.6000	SkillOpt-DS
Erdős Min Overlap	Q3.8-27B	0.2366	0.2366	Base
Erdős Min Overlap	K3	0.2347	0.2347	Base
Erdős Min Overlap	DS-v4	0.2358	0.2358	Base
REI	Q3-8B	0.7951	0.7951	Base
REI	Q3.6-27B	0.8326	0.8479	TTT
REI	Q3.8-27B	0.7846	0.8917	SkillOpt-DS
REI	K3	0.8311	0.8311	Base
REI	DS-v4	0.8076	0.8076	Base
BFCL V4	Q3-8B	30.47	33.59	SkillOpt-K3
BFCL V4	Q3.6-27B	57.03	75.78	SkillOpt-DS
BFCL V4	Q3.8-27B	71.88	71.88	Base
BFCL V4	K3	67.97	67.97	Base
BFCL V4	DS-v4	74.22	74.22	Base
τ^2-bench	Q3-8B	25.00	30.00	SkillOpt-K3
τ^2-bench	Q3.6-27B	56.25	60.00	SkillOpt-K3
τ^2-bench	Q3.8-27B	50.00	60.00	SkillOpt-DS
τ^2-bench	K3	60.00	60.00	Base
τ^2-bench	DS-v4	58.75	58.75	Base
API-Bank	Q3-8B	57.00	62.13	SFT

Task	Target	Initial	Best	Selected
API-Bank	Q3.6-27B	64.06	66.62	SkillOpt-DS
API-Bank	Q3.8-27B	67.01	67.01	Base
API-Bank	K3	67.39	67.39	Base
API-Bank	DS-v4	66.37	66.37	Base
Spider	Q3-8B	71.33	71.33	Base
Spider	Q3.6-27B	68.67	84.00	SkillOpt-K3
Spider	Q3.8-27B	73.30	79.20	SkillOpt-DS
Spider	K3	74.67	74.67	Base
Spider	DS-v4	73.00	73.00	Base
IFEval	Q3-8B	84.43	84.70	ICL
IFEval	Q3.6-27B	70.18	88.13	SkillOpt-DS
IFEval	Q3.8-27B	82.85	82.85	Base
IFEval	K3	–	–	–
IFEval	DS-v4	87.34	87.34	Base
FollowBench	Q3-8B	80.67	80.67	Base
FollowBench	Q3.6-27B	84.33	84.33	Base
FollowBench	Q3.8-27B	84.00	84.00	Base
FollowBench	K3	90.33	90.33	Base
FollowBench	DS-v4	86.33	86.33	Base
TNEWS	Q3-8B	48.80	52.20	SFT
TNEWS	Q3.6-27B	54.20	55.20	SkillOpt-DS
TNEWS	Q3.8-27B	49.80	54.80	SkillOpt-DS
TNEWS	K3	52.60	52.60	Base
TNEWS	DS-v4	47.20	47.20	Base
CLUENER2020	Q3-8B	0.00	54.26	ICL
CLUENER2020	Q3.6-27B	54.25	71.45	SkillOpt-DS
CLUENER2020	Q3.8-27B	42.90	48.15	RL
CLUENER2020	K3	55.11	55.11	Base
CLUENER2020	DS-v4	38.51	38.51	Base

Table 6: 8B method endpoints behind Figure 3. SkillOpt uses the recorded DeepSeek optimizer. †: Qwen3.8-27B supplement; other 27B cells use Qwen3.6. –: no plotted value; Excl.: incompatible/diagnostic entry. These states are not zero scores.

Task	ICL	SkillOpt	SFT	RL	TTT
LiveMath	51.61	51.61	87.10	–	–
SearchQA	71.50	73.57	77.43	58.00	–
HotpotQA	57.14	55.29	58.29	59.70	–
FinQA	26.33	46.29	42.20	61.46	–
SpreadsheetBench	10.36	12.14	10.36	–	–
Circle Packing N=26	Excl.	0.8471	–	–	0.8208
Erdős Min Overlap	0.0000	0.1746	–	–	0.2091
REI	0.5744	0.4270	–	–	0.5255
BFCL V4	32.03	28.91	7.81	–	–
τ^2-bench	12.50	25.00	30.00	–	–
API-Bank	53.92	60.98	62.13	43.65	–
Spider	58.80	63.70	70.60	65.33	–
IFEval	84.70	82.85	79.16	84.43	–
FollowBench	44.52	48.87	43.65	Excl.	–
TNEWS	47.40	44.00	52.20	–	–
CLUENER2020	54.26	46.96	Excl.	45.45	–

Table 7: 27B method endpoints behind Figure 3. SkillOpt uses the recorded DeepSeek optimizer. †: Qwen3.8-27B supplement; other 27B cells use Qwen3.6. –: no plotted value; Excl.: incompatible/diagnostic entry. These states are not zero scores.

Task	ICL	SkillOpt	SFT	RL	TTT
LiveMath	61.29	57.26	91.13	–	–
SearchQA	64.79	82.14	80.29	79.50†	–
HotpotQA	64.14	67.57	67.71	65.70†	–
FinQA	53.71	71.23	45.68	34.18†	–
SpreadsheetBench	46.43	55.71	33.93	–	–
Circle Packing N=26	Excl.	0.6933	–	–	0.8786
Erdős Min Overlap	0.2368	0.6000	–	–	0.2367
REI	0.8169	0.7712	–	–	0.8479
BFCL V4	68.75	75.78	46.88	–	–
τ^2-bench	55.00	50.00	12.50	–	–
API-Bank	65.60	66.62	65.47	48.65†	–
Spider	81.20	82.90	82.10	78.33†	–
IFEval	86.81	88.13	83.11	81.27†	–
FollowBench	60.70	67.83	60.35	–	–
TNEWS	54.20	55.20	54.60	–	–
CLUENER2020	69.42	71.45	Excl.	48.15†	–

C Q2: Diagnostic Evidence and Robustness Checks

This appendix follows the three diagnostics in Figure 4: error repairability, workload composition, and supervision availability. It specifies their measurements and interpretation, then reports paired RL outcomes supporting the corrected endpoints in Appendix B.3.

C.1 Diagnostic definitions and data scope

Error repairability: Figure 4(a–b). Figure 4(a) and Figure 4(b) replay FinQA outputs from Qwen3.8-27B baseline and RL test runs ($n = 1{,}147$ each), and K3 teacher training outputs ($n = 1{,}000$). A fixed AST rule removes a final multiplication of the preceding result by 100, in either operand order, from programs with at least two operations. The original evaluator is unchanged and reproduces recorded correctness before replay. Baseline repairs/regressions are 380/14; RL, 383/14; teacher, 380/21. Baseline net gain is 31.91 pp, with a company–year cluster-bootstrap interval of $[28.35, 35.50]$ pp across 278 clusters. This interval conditions on the chosen rule and does not account for post-hoc rule discovery. Replay scores are not official adaptation results or evidence that SkillOpt discovers the rule.

Workload composition: Figure 4(c). Figure 4(c) uses K3-optimized SkillOpt versus training-only three-shot ICL on 779 shared scoring positions with the same parser and API correctness function. For Levels 1/2/3, correct counts are 301/56/113 (8B) and 301/74/131 (Qwen3.6-27B), versus ICL counts 307/65/48 and 307/75/129, respectively; denominators are 399/135/245. For Level-3 share p, the plotted difference is $(1-p)\Delta_{1,2} + p\Delta_3$, preserving the observed Level-1:Level-2 ratio. Dots mark the observed mixture. Marginal counts do not provide paired uncertainty or establish performance after distribution shift.

Supervision availability: Figure 4(d–f). Figure 4(d) compares Qwen3.8-27B IFEval baseline errors by whether the exact multiset of constraint types occurs among successful K3 demonstrations. Pooled unseen/seen error counts are 41/148 and 36/231; for one constraint, 2/18 and 25/195; for two, 22/90 and 11/34. The three-constraint stratum has only two seen examples and is omitted from the stratified bars. All 25 atomic constraint types are covered, making atomic coverage uninformative here. These baseline associations do not estimate SFT effects.

Figure 4(e) and Figure 4(f) use 12 task-level observations, averaging the two targets' SFT gains against their own released Tinker controls; CLUENER is excluded because its reported metrics

are incompatible. Teacher counts denote successful outputs, not total attempts; API-Bank pools success across four attempts. We have teacher traces and released SFT summaries, but not local per-instance SFT outputs. For Figure 4(e) and Figure 4(f), we convert mean gains to percentile ranks $100(r - 0.5)/12$, where r is the ascending gain rank (averaged for ties). Both panels use the same transformation; their horizontal order follows the respective supervision indicator. Gray bars retain the sign of the original negative gain. This displays ordering, not gain magnitude, and does not force monotonicity. Spearman correlations are exploratory. In leave-one-task-out linear prediction, accepted count yields mean absolute error 10.41 pp versus 10.92 for a training-fold median predictor (13.04 for a mean predictor); teacher success yields 13.59 pp. Headroom-based prediction also fails to beat a constant for either target. Count, coverage, and success rates are not independently randomized; no causal training-data claim follows. Prospective selection must compute diagnostics on permitted development data, keeping final-test outcomes unavailable.

C.2 Paired RL outcomes and uncertainty

We align Qwen3.8-27B baseline and RL outputs by unique instance identifier, verify matching questions, and use hard correctness. IFEval strata use the number of strict instruction checks; FinQA strata count reference-program operations, pooling three or more. These properties do not depend on adapted outputs. Repair denotes incorrect-to-correct and regression denotes correct-to-incorrect, so the gain equals $100(R - D)/n$ percentage points.

Task	Stratum	n	Repairs	Regressions	Δ pp	95% interval
IFEval	1 constraint	213	4	2	$+0.94$	$[-1.41, 3.29]$
IFEval	2 constraints	124	4	3	$+0.81$	$[-3.23, 4.84]$
IFEval	3 constraints	42	3	0	$+7.14$	$[0.00, 16.67]$
IFEval	All	379	11	5	$+1.58$	$[-0.53, 3.69]$
FinQA	1 operation	654	4	3	$+0.15$	$[-0.61, 0.92]$
FinQA	2 operations	409	1	5	-0.98	$[-2.20, 0.24]$
FinQA	3+ operations	84	0	2	-2.38	$[-5.95, 0.00]$
FinQA	All	1147	5	10	-0.44	$[-1.13, 0.17]$

Table 8: Paired RL transitions and gains. Stratum rows partition each task; the overall row summarizes the same instances rather than new observations.

These are exploratory post-hoc strata. Percentile intervals use 20,000 paired bootstrap draws (seed 20260924), resampling outcome differences within each stratum. They are pointwise, without multiple-comparison adjustment, and do not represent variability across training seeds. All stratum intervals include or touch zero. FinQA program-valid outputs decrease from 1,025 to 1,014 among 1,147 instances; validity is not correctness. No paired interval is inferred from the API-Bank marginal counts. The retained analysis artifacts preserve source hashes and exact counts. These analyses locate observed changes without establishing a universal task-feature-to-method rule.

C.3 Task-level supervision observations

Table 9 supplies the values behind Figure 4(e–f). Mean gain averages the two matched-control SFT gains in Table 4. API-Bank's 100 accepted examples pool four attempts, so its success rate is not a single-attempt teacher accuracy. Training quantities also appear in Table 2; this table adds the precise diagnostic input and response pairing.

Table 9: Task-level inputs to the supervision analysis. Success is a percentage; mean SFT gain is in percentage points.

Task	Source	Accepted	Success (%)	Mean gain
LiveMath	466	191	40.99	60.485
SearchQA	400	296	74.00	16.220
HotpotQA	200	163	81.50	1.145
FinQA	1000	383	38.30	21.055
SpreadsheetBench	80	28	35.00	-2.855

Task	Source	Accepted	Success (%)	Mean gain
BFCL V4	40	17	42.50	-3.905
τ^2-bench	54	34	62.96	3.750
API-Bank	223	100	44.84	3.020
IFEval	108	104	96.30	-5.015
FollowBench	165	101	61.21	-3.650
TNEWS	400	200	50.00	5.300
Spider	286	197	68.88	8.550

Source count is the number of source prompts; Log(1+count) transforms accepted count. Type shift is the total-variation distance between task-type distributions before and after demonstration filtering. LM/BFCL/SSB denote LiveMath, BFCL, and SpreadsheetBench. These auxiliary predictors and subset checks are exploratory; the table reports association with mean SFT gain.

Table 10: Exploratory supervision associations and leave-one-task-out (LOO) prediction errors. Permutation p-values are unadjusted; MAE is in percentage points. Subset identifiers describe sensitivity exclusions, not independent experiments.

Subset	Predictor	n	ρ	p	LOO MAE
All 12	Success rate	12	-0.238	0.461	13.59
All 12	Accepted count	12	0.720	0.013	10.41
All 12	Log(1+count)	12	0.720	0.013	10.99
All 12	Source count	12	0.834	0.001	13.20
All 12	Type shift (TV)	12	-0.070	0.830	12.42
No LiveMath	Success rate	11	-0.091	0.792	8.02
No LiveMath	Accepted count	11	0.727	0.014	3.66
No LiveMath	Log(1+count)	11	0.727	0.014	5.77
No LiveMath	Source count	11	0.793	0.005	4.76
No LiveMath	Type shift (TV)	11	-0.064	0.850	7.20
No API-Bank	Success rate	11	-0.273	0.419	14.27
No API-Bank	Accepted count	11	0.755	0.010	11.38
No API-Bank	Log(1+count)	11	0.755	0.010	11.58
No API-Bank	Source count	11	0.829	0.003	14.34
No API-Bank	Type shift (TV)	11	-0.032	0.931	13.55
No LM/BFCL/SSB	Success rate	9	-0.433	0.249	7.66
No LM/BFCL/SSB	Accepted count	9	0.700	0.043	4.51
No LM/BFCL/SSB	Log(1+count)	9	0.700	0.043	7.76
No LM/BFCL/SSB	Source count	9	0.803	0.012	5.53
No LM/BFCL/SSB	Type shift (TV)	9	-0.217	0.578	7.33

D Q3: Adaptation Costs and Accounting Evidence

D.1 Accounting scope

Figures 5 and 8 show recorded experiment costs, including final evaluation. The component inventory retains all 104 observations. The numerical adaptation-cost comparisons instead follow Equation 1, excluding independent final evaluation. SFT charges full teacher construction to each independently reproduced target. Joint campaigns charge shared data once, and existing-data scenarios exclude its sunk preparation cost. Diagnostic base controls are not charged.

Costs include catalog-valued provider usage, reconstructed ledgers, no-cache training upper estimates, and subscription API equivalents; they are not uniformly exact bills. The corrected BFCL total replaces the old erroneous final test while retaining training and validation expenses. Updated FinQA/IFEval 27B RL endpoints have no complete adaptation costs. Unreported costs are never treated as zero.

D.2 Adaptation-only costs and unresolved allocations

For each recorded component, a separately identified final-test charge is removed; a component containing only preparation, optimization, training, or validation is retained. If a combined charge includes final testing, its non-test contribution lies between zero and the entire recorded component. Adding known non-test costs yields the reported adaptation-cost interval. Optimizer-model charges are retained when available, but target-model charges spanning rollouts, validation, and final testing remain unresolved. Source rounding differences of at most \$0.000001 are tracked separately. These bounds hold under the reported valuation and scope; they are not bounds on unknown real invoices, nor statistical error bars.

The overview plots retain all 16 tasks and their recorded method candidates. SkillOpt uses the recorded DeepSeek optimizer and SFT uses K3 teacher data. Black crosses identify records excluded from frontier construction because of incompatible model versions, accounting scopes, or cost bounds. Among eligible records within each task, a point is non-dominated if no other point has lower or equal recorded cost and higher or equal score, with at least one strict inequality. Dashed lines connect these points in cost order; a ring marks a single-point frontier. Connections do not imply that intermediate outcomes are attainable. We replace the common initial-score line with method-specific unadapted controls from the audited SFT and matched context-method comparisons. A short horizontal tick marks each control, with a dotted vertical stem to its adapted endpoint. Both are placed at the adapted run's cost only for pairing; no baseline cost is implied. Upward/downward stems indicate gains/losses. Controls are omitted where a compatible pair has not been established, including RL/TTT; omission does not imply zero gain. Controls do not enter the frontier. Large-model horizontal lines remain descriptive references, not matched controls, and have no assigned cost coordinate.

Curved backgrounds illustrate the preference function $U(s, c) = s - 0.25 \log_{10}(c/c_{\min})$, where s is the score on a 0–1 scale and $c_{\min}$ is the lowest displayed task cost. Contour levels are scaled separately within each panel. They illustrate a preference for higher quality and lower cost, not a fitted response curve or a measured frontier. The frontier concerns the eligible recorded candidates, not an exhaustive optimum or a guarantee about actual invoices.

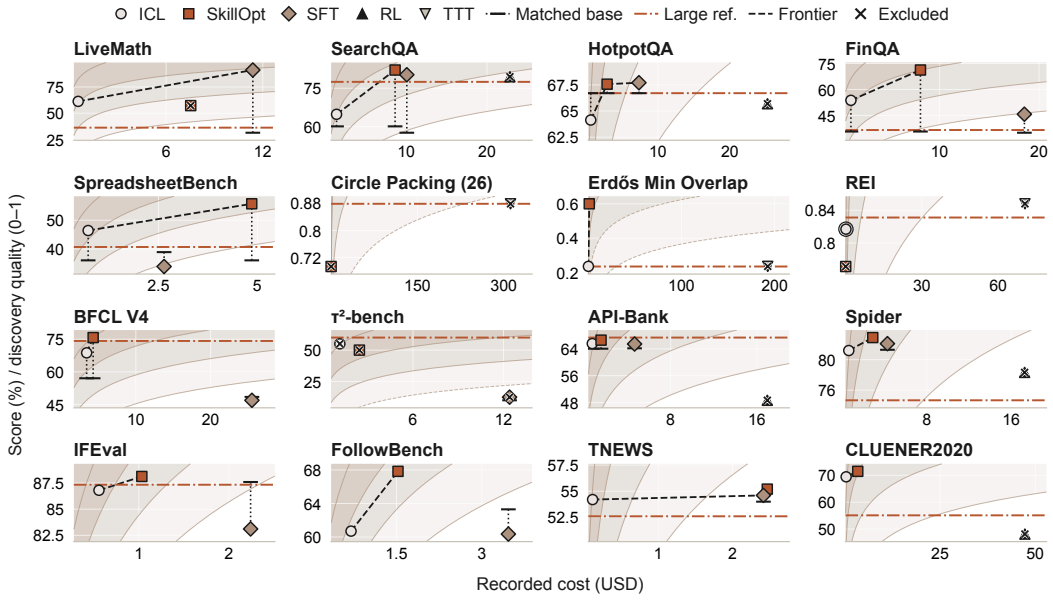


Figure 8: Recorded experiment cost and performance for the 27B target across 16 tasks, including final evaluation with panel-specific score ranges, with the same visual conventions as Figure 5. Black crosses denote records excluded from the frontier, including incompatible model versions or cost scopes.

For 8B Spider, SFT has \$0.774450 teacher preparation and a combined \$0.360768 training/validation/test charge. Its adaptation interval is therefore [\$0.774450, \$1.135218]. SkillOpt

has $1.641296 optimization/validation and $0.234232 final evaluation, leaving $1.641296 as adaptation expenditure. SFT's worst-case recorded allocation is 30.83% cheaper. This comparison does not require recovering SFT's unknown final-test share or interpolating a cost–performance frontier. For API-Bank 8B, the analogous SFT interval is [$2.384859, $2.621458]; when identical teacher data already exist, it becomes [$0, $0.236599], compared with SkillOpt's $0.890141. The zero endpoint is an accounting lower bound, not a claim that training is free. Reusing a skill would require its own scenario.

D.3 Stage-wise cost accounting

We distinguish five monetary stages. *Data preparation* covers teacher demonstrations and generated demonstration pools, including recorded unsuccessful attempts and teacher judging. *Optimization* covers strategy search and rollout generation; *training* covers weight updates. *Validation* covers checkpoint or skill selection, and *final evaluation* covers the frozen candidate's test execution. The final stage is an evaluation expense, not adaptation. Human curation, local hardware, and unrecorded charges are outside this service-cost accounting. Missing amounts are not zero.

The available reports permit the following decompositions of the 104 original experiment-cost observations:

- ICL (30 observations): final evaluation, plus separately reported demonstration generation on the two Erdős runs.
- SFT (24): teacher preparation plus recipe cost on all runs. Only SpreadsheetBench 8B/27B and BFCL 27B separately report training, validation, and final evaluation; the other 21 recipe charges remain combined.
- SkillOpt (32): ten runs on API-Bank, IFEval, TNEWS, Spider, and Erdős separate optimization plus validation from final evaluation. Four additional actor-wise breakdowns distinguish optimizer calls from target calls, but cannot separate target rollouts, validation, and test. Circle Packing and REI report final evaluation only; their missing prior optimization costs prevent an adaptation-cost comparison.
- RL (12) and TTT (6): recorded totals do not support a complete stage-wise dollar allocation. We retain combined charges, existing scope exclusions, and cost bounds.

Figure 6(a) shows all three fully itemized SFT observations; panel (b) illustrates the separately recorded final-evaluation charges on API-Bank and IFEval SkillOpt runs. Panel (c) compares API-Bank 8B adaptation expenditure with new versus reused teacher data. SFT bars are upper bounds retaining the full pooled recipe charge; SkillOpt uses its identified optimization/validation charge. SkillOpt's optimization and validation expenses remain one hatched component. The plotted components reconcile with recorded totals to within $0.000001 of source rounding; no residual is interpreted as a newly measured stage.

Across the 12 SFT tasks in the component inventory, teacher preparation exceeds half of 8B total cost on 11 tasks (median share 81.46%); for 27B the corresponding values are two tasks and 31.95%. These are within-run shares, not shares of a pooled campaign bill. For 8B API-Bank, the decomposition is $2.384859 teacher plus $0.236599 recipe. If the identical teacher data already exist, only the latter is a new expense; this conditional scenario leaves the measured score unchanged and does not assert transfer to a new workload. Reusing an existing skill would likewise require a separate incremental-cost comparison. BFCL 27B comprises $1.586712 teacher, $1.449114 training, $3.600974 validation, and $18.887475 final evaluation; its high total therefore primarily reflects test execution.

D.4 Complete recorded-cost inventory

Table 12 reports every plotted costed endpoint in Figures 5 and 8. Total includes final evaluation; Adapt. is the non-test allocation interval under the recorded accounting, not a confidence interval or provider invoice. Values are rounded to four decimal places here; the accompanying machine-readable disclosure retains full precision and every recorded component. Est.: estimated/mixed valuation; Rep.: reported; Bound: one-sided cost bound. Y/N indicates eligibility for the descriptive frontier, not superiority. A dash in Adapt. means that a complete usable allocation is not established; no zero is imputed.

Task	Teacher ($)	8B Recipe ($)	8B Teacher share	27B Recipe ($)	27B Teacher share
LiveMath	8.561	0.297	96.7%	2.824	75.2%
SearchQA	2.037	0.849	70.6%	8.024	20.2%
HotpotQA	1.181	0.641	64.8%	6.189	16.0%
FinQA	6.210	1.283	82.9%	12.318	33.5%
SpreadsheetBench	0.925	0.198	82.3%	1.717	35.0%
BFCL	1.587	2.550	38.4%	23.938	6.2%
τ^2-bench	2.804	2.062	57.6%	9.596	22.6%
API-Bank	2.385	0.237	91.0%	2.271	51.2%
Spider	0.774	0.361	68.2%	3.498	18.1%
IFEval	0.952	0.183	83.9%	1.290	42.5%
FollowBench	1.390	0.233	85.7%	2.076	40.1%
TNEWS	0.734	0.177	80.6%	1.681	30.4%

Table 11: SFT cost components for all 12 tasks in the original cost inventory. Recipe combines training, validation, and final evaluation; teacher is charged in full to each independently reproduced target, but only once in a joint campaign. Shares use teacher plus recipe, not the recipe alone. Values are recorded estimates, including no-cache upper estimates and API-equivalent teacher costs.

Table 12: Complete plotted cost ledger (USD). Point costs, bounds, and allocation intervals retain their distinct meanings.

Task	Target	Method	Total	Adapt.	Status	Eligible
LiveMath	Q3-8B	ICL	0.0295	0.0000	Est.	Y
LiveMath	Q3-8B	SkillOpt	0.9951	0.0000–0.9951	Est.	N
LiveMath	Q3-8B	SFT	8.8577	8.5610–8.8577	Est.	Y
SearchQA	Q3-8B	ICL	0.3335	0.0000	Rep.	Y
SearchQA	Q3-8B	SkillOpt	4.6516	0.0000–4.6516	Rep.	Y
SearchQA	Q3-8B	SFT	2.8854	2.0366–2.8854	Est.	Y
SearchQA	Q3-8B	RL	6.2801	0.0000–6.2801	Est.	N
HotpotQA	Q3-8B	ICL	0.1865	0.0000	Rep.	Y
HotpotQA	Q3-8B	SkillOpt	1.9193	0.0000–1.9193	Est.	Y
HotpotQA	Q3-8B	SFT	1.8223	1.1809–1.8223	Est.	Y
HotpotQA	Q3-8B	RL	4.0318	0.0000–4.0318	Est.	N
FinQA	Q3-8B	ICL	0.3742	0.0000	Rep.	Y
FinQA	Q3-8B	SkillOpt	3.5844	0.0000–3.5844	Rep.	Y
FinQA	Q3-8B	SFT	7.4934	6.2100–7.4934	Est.	Y
FinQA	Q3-8B	RL	25.8760	0.0000–25.8760	Est.	Y
SpreadsheetBench	Q3-8B	ICL	0.1495	0.0000	Est.	Y
SpreadsheetBench	Q3-8B	SkillOpt	1.1995	0.4997–1.1995	Est.	Y
SpreadsheetBench	Q3-8B	SFT	1.1229	0.9930	Est.	Y
Circle Packing N=26	Q3-8B	SkillOpt	0.0072	–	Rep.	N
Circle Packing N=26	Q3-8B	TTT	11.6462	0.0000–11.6462	Est.	N
Erdős Min Overlap	Q3-8B	ICL	0.0034	0.0016	Rep.	Y
Erdős Min Overlap	Q3-8B	SkillOpt	0.2261	0.2224	Est.	Y
Erdős Min Overlap	Q3-8B	TTT	6.5929	0.0000–6.5929	Est.	N
REI	Q3-8B	ICL	0.0133	0.0000	Rep.	Y
REI	Q3-8B	SkillOpt	0.0102	–	Rep.	N
REI	Q3-8B	TTT	2.1743	0.0000–2.1743	Est.	Y
BFCL V4	Q3-8B	ICL	1.5675	0.0000	Rep.	Y
BFCL V4	Q3-8B	SkillOpt	2.4612	0.0000–2.4612	Rep.	Y
BFCL V4	Q3-8B	SFT	4.1367	1.5867–4.1367	Est.	Y
τ^2-bench	Q3-8B	ICL	0.6816	0.0000	Rep.	N
τ^2-bench	Q3-8B	SkillOpt	24.5650	0.0000–24.5650	Rep.	N
τ^2-bench	Q3-8B	SFT	4.8662	2.8043–4.8662	Est.	N
API-Bank	Q3-8B	ICL	0.2859	0.0000	Rep.	Y
API-Bank	Q3-8B	SkillOpt	1.0210	0.8901	Est.	Y
API-Bank	Q3-8B	SFT	2.6215	2.3849–2.6215	Est.	Y
API-Bank	Q3-8B	RL	3.0451	0.0000–3.0451	Est.	N
Spider	Q3-8B	ICL	0.2102	0.0000	Rep.	Y
Spider	Q3-8B	SkillOpt	1.8755	1.6413	Est.	Y

Task	Target	Method	Total	Adapt.	Status	Eligible
Spider	Q3-8B	SFT	1.1352	0.7744–1.1352	Est.	Y
Spider	Q3-8B	RL	7.4113	0.0000–7.4113	Est.	N
IFEval	Q3-8B	ICL	0.1680	0.0000	Rep.	Y
IFEval	Q3-8B	SkillOpt	0.5925	0.5134	Est.	Y
IFEval	Q3-8B	SFT	1.1344	0.9517–1.1344	Est.	Y
IFEval	Q3-8B	RL	3.1096	0.0000–3.1096	Est.	Y
FollowBench	Q3-8B	ICL	0.2128	0.0000	Rep.	Y
FollowBench	Q3-8B	SkillOpt	1.4590	0.0000–1.4590	Rep.	Y
FollowBench	Q3-8B	SFT	1.6225	1.3900–1.6225	Est.	Y
TNEWS	Q3-8B	ICL	0.0378	0.0000	Rep.	Y
TNEWS	Q3-8B	SkillOpt	1.6615	1.6024	Rep.	Y
TNEWS	Q3-8B	SFT	0.9103	0.7336–0.9103	Est.	Y
CLUENER2020	Q3-8B	ICL	0.1207	0.0000	Rep.	Y
CLUENER2020	Q3-8B	SkillOpt	1.8099	–	Bound	N
CLUENER2020	Q3-8B	RL	17.9338	0.0000–17.9338	Est.	N
LiveMath	Q3.6-27B	ICL	0.5366	0.0000	Est.	Y
LiveMath	Q3.6-27B	SkillOpt	7.5255	0.0000–7.5255	Est.	N
LiveMath	Q3.6-27B	SFT	11.3847	8.5610–11.3847	Est.	Y
SearchQA	Q3.6-27B	ICL	1.3538	0.0000	Rep.	Y
SearchQA	Q3.6-27B	SkillOpt	8.6091	0.0000–8.6091	Rep.	Y
SearchQA	Q3.6-27B	SFT	10.0610	2.0366–10.0610	Est.	Y
SearchQA	Q3.8-27B	RL	22.8421	0.0000–22.8421	Est.	N
HotpotQA	Q3.6-27B	ICL	0.6715	0.0000	Rep.	Y
HotpotQA	Q3.6-27B	SkillOpt	2.9515	0.0000–2.9515	Est.	Y
HotpotQA	Q3.6-27B	SFT	7.3696	1.1809–7.3696	Est.	Y
HotpotQA	Q3.8-27B	RL	25.2298	0.0000–25.2298	Est.	N
FinQA	Q3.6-27B	ICL	1.0803	0.0000	Rep.	Y
FinQA	Q3.6-27B	SkillOpt	8.0752	0.0000–8.0752	Rep.	Y
FinQA	Q3.6-27B	SFT	18.5282	6.2100–18.5282	Est.	Y
SpreadsheetBench	Q3.6-27B	ICL	0.7223	0.0000	Est.	Y
SpreadsheetBench	Q3.6-27B	SkillOpt	4.8576	0.6531–4.8576	Est.	Y
SpreadsheetBench	Q3.6-27B	SFT	2.6417	1.5898	Est.	Y
Circle Packing N=26	Q3.6-27B	SkillOpt	0.0530	–	Rep.	N
Circle Packing N=26	Q3.6-27B	TTT	311.6268	–	Bound	N
Erdős Min Overlap	Q3.6-27B	ICL	0.1933	0.1501	Rep.	Y
Erdős Min Overlap	Q3.6-27B	SkillOpt	1.1093	1.0650	Est.	Y
Erdős Min Overlap	Q3.6-27B	TTT	192.5641	–	Bound	N
REI	Q3.6-27B	ICL	0.2187	0.0000	Rep.	Y
REI	Q3.6-27B	SkillOpt	0.1283	–	Rep.	N
REI	Q3.6-27B	TTT	70.4712	–	Bound	N
BFCL V4	Q3.6-27B	ICL	3.4206	0.0000	Rep.	Y
BFCL V4	Q3.6-27B	SkillOpt	4.2889	0.0000–4.2889	Rep.	Y
BFCL V4	Q3.6-27B	SFT	25.5243	6.6368	Est.	Y
τ^2-bench	Q3.6-27B	ICL	1.1595	0.0000	Rep.	N
τ^2-bench	Q3.6-27B	SkillOpt	2.4552	0.0000–2.4552	Rep.	N
τ^2-bench	Q3.6-27B	SFT	12.4006	2.8043–12.4006	Est.	N
API-Bank	Q3.6-27B	ICL	0.7331	0.0000	Rep.	Y
API-Bank	Q3.6-27B	SkillOpt	1.5480	1.2340	Rep.	Y
API-Bank	Q3.6-27B	SFT	4.6558	2.3849–4.6558	Est.	Y
API-Bank	Q3.8-27B	RL	16.9608	0.0000–16.9608	Est.	N
Spider	Q3.6-27B	ICL	0.6119	0.0000	Rep.	Y
Spider	Q3.6-27B	SkillOpt	2.8824	2.2699	Est.	Y
Spider	Q3.6-27B	SFT	4.2723	0.7744–4.2723	Est.	Y
Spider	Q3.8-27B	RL	17.2626	0.0000–17.2626	Rep.	N
IFEval	Q3.6-27B	ICL	0.5572	0.0000	Rep.	Y
IFEval	Q3.6-27B	SkillOpt	1.0335	0.7289	Rep.	Y
IFEval	Q3.6-27B	SFT	2.2416	0.9517–2.2416	Est.	Y
FollowBench	Q3.6-27B	ICL	0.7016	0.0000	Rep.	Y
FollowBench	Q3.6-27B	SkillOpt	1.5265	0.0000–1.5265	Rep.	Y
FollowBench	Q3.6-27B	SFT	3.4663	1.3900–3.4663	Est.	Y
TNEWS	Q3.6-27B	ICL	0.1150	0.0000	Rep.	Y
TNEWS	Q3.6-27B	SkillOpt	2.4666	2.3066	Rep.	Y

Task	Target	Method	Total	Adapt.	Status	Eligible
TNEWS	Q3.6-27B	SFT	2.4150	0.7336–2.4150	Est.	Y
CLUENER2020	Q3.6-27B	ICL	0.3136	0.0000	Rep.	Y
CLUENER2020	Q3.6-27B	SkillOpt	3.3658	1.2173–3.3658	Rep.	Y
CLUENER2020	Q3.8-27B	RL	47.2699	0.0000–47.2699	Rep.	N

Unavailable or excluded costs. The machine-readable companion retains records without usable costs and all frontier exclusion reasons, including incompatible evaluation scopes, model versions, final-evaluation-only charges, and bounded expenditure. Controls are separate score references and never receive the adapted run's cost as their own expenditure.

E Validation-Gated Method Selection

E.1 Completed candidate-library replay

We implement a quality-only gate that accepts validation scores and an exact workload/model/backend contract, averages complete recipe seeds, and selects the largest mean. Exact ties favor the unadapted baseline, then a fixed lexicographic method order. Test outcomes are loaded only to score the chosen candidate. This is a retrospective analysis designed after the result inventory was available, not a preregistered policy trial. The proposed sequential layer specifies admissibility, probe priority, and cost feasibility; the replay does not measure its exploration efficiency or the benefit of diagnostic retrieval.

Proposed sequential layer versus evaluated gate. Equation 2 specifies a proposed probe-ordering rule, not the existing replay's scoring function. Diagnostic tags and their extraction must be fixed before target adaptation. Jaccard similarity is zero for empty or disjoint tag sets; only positive-weight compatible records enter the ratio. Within each recipe, first average repeated source seeds within their workload, then weight workload means. Unsupported recipes follow supported recipes by quoted probe cost; priority ties use cost then fixed method name. Baseline validation initializes the observed set and its cost enters the ledger. The executor must enforce each quoted probe cap and record failed attempts; if no affordable untried recipe remains, the best observed validation candidate is frozen. The full-library replay bypasses this ordering and its budget stopping rule, and BIRD/MuSR use the restricted published candidate set specified below rather than an unobserved baseline-validation value.

Executable decision rule. Let $\mathcal{M}_t$ be the completed recipes admitted for a target contract, V_t its selection examples, and $\mathcal{S}_m$ the required seeds. We compute

$$\bar{v}_t(m) = \frac{1}{|\mathcal{S}_m|} \sum_{s \in \mathcal{S}_m} \frac{1}{|V_t|} \sum_{i \in V_t} a_{m,s,i}, \qquad \hat{m}_t = \arg\max_{m \in \mathcal{M}_t} \bar{v}_t(m), \tag{3}$$

where $a_{m,s,i}$ is the benchmark's correctness indicator. Reported aggregate scores supply the same mean for the report-only BIRD/MuSR replay.

1. **Lock the contract and roles.** Fix the workload, exact model, backend, scorer, recipe identity, seed set, and validation IDs. Final-test scores are not gate inputs. For IFBench, lock all five folds before evaluating any fold; the other 80 IDs are its selection input.
2. **Admit complete, compatible recipes.** Require the requested seed set (17 for BIRD/MuSR; 17/43/101 for CRUXEval-O/IFBench). A baseline may use its one recorded run. Do not pool model versions or backends; keep original and DeepSeek SFT as distinct recipes. If no candidate remains, return no recommendation instead of filling a score.
3. **Compare all admitted candidates.** Take the arithmetic mean across seeds; never pick a seed using validation or test scores. This is a full-library comparison, with no history-based pruning or early stopping. A two-point validation lead is used as observed, not treated as proof of statistical superiority; no significance threshold was tuned in this replay.

4. **Resolve ties and freeze the recipe.** Compare means rounded to 12 decimal places, prefer Base on a tie, then sort tied method names lexicographically. Save all seed scores, the sorted ranking, exclusions, the top-two margin, and the selected recipe. Retain all required recipe seeds for evaluation rather than deploying its best observed seed.
5. **Evaluate without revision.** Read only the frozen recipe's held-out outcomes to compute its score and gain over the matching base. Other candidates' test scores may diagnose regret afterward, but never revise the recommendation. In IFBench, aggregate each example only in its held-out fold; this does not construct an ensemble.

These steps specify the evaluated quality gate. The proposed sequential layer uses historical evidence to order probes and specifies budget constraints, but neither mechanism supplies a hidden score bonus or eligibility shortcut in the reported replay. Q2 diagnostics are not converted into fitted selection weights by this experiment.

BIRD Mini-Dev and MuSR. The source report publishes seed-17 Tinker validation pairs (SFT/RL): BIRD 8B 60/51, BIRD 27B 67/70, MuSR 8B 61/63, and MuSR 27B 72/51 percent. The target gate therefore chooses SFT, RL, RL, and SFT, respectively. Only these two training recipes enter each comparison; OpenRouter ICL and SkillOpt have a different execution contract and are excluded. Tinker base scores are evaluation references, not selectable candidates, because their selector-validation scores are not in the published report. The published source-only policy transfers the other benchmark's SFT/RL ranking and was frozen before its original test execution. Its test scores for the four pairs are 36.80, 60.80, 62.89, and 57.03 percent. Our target-validation replay uses the same seed-17 candidate matrix, including subsequent diagnostic completions, and must not be relabeled as that original prospective run. Additional training seeds are not pooled into this replay because their validation rankings are not published. These results are report-based; per-item traces are outside the committed snapshot.

CRUXEval-O. Baseline, ICL, original SFT, RL, and SkillOpt share the target's frozen Tinker contract. Adapted recipes average seeds 17, 43, and 101; the baseline has one run and is not treated as three independent measurements. Validation uses 100 examples and final evaluation uses 400 disjoint research test programs. The gate selects baseline for 8B (a 52% validation tie with SFT) and RL for 27B (95.33% validation). Their final scores are 51.25% and 91.75%; the latter is 19.75 points above the 72.00% base score. These are recorded candidate outcomes, not official full-800 benchmark scores or evidence that the selector caused RL's improvement.

IFBench. The candidate library also contains the separate DeepSeek SFT recipe. We partition the 100 selector-validation IDs into five folds by sorting their SHA-256 hashes with a fixed salt and assigning alternating ranks modulo five. Each fold selects on the other 80 examples using three-seed mean accuracy, then evaluates on its 20 held-out examples. All methods share identical folds. Adaptation used the separate training and optimizer-development roles; no held-out fold labels enter its own gate decision. The 8B gate selects SkillOpt in all five folds; 27B selects SkillOpt in four and RL in one. Aggregating held-out predictions gives 81.33% and 82.33%, against base scores of 74.00% and 82.00%. Folds and seeds are dependent, so no independent-fold confidence interval or significance claim is made. This analysis evaluates validation generalization, not the official 300-item final test.

E.2 Exact decision paths for the four workloads

All scores below are percentages. Bold validation entries identify the chosen recipe. Held-out scores are appended only after the validation decision; they do not determine the bold entries. No Q2 feature weight, source-task rank, or monetary cost enters these quality-only decisions.

BIRD Mini-Dev: compare two compatible training recipes. The 8B path is SFT 60.00 versus RL 51.00 $\rightarrow$ SFT; the 27B path is SFT 67.00 versus RL 70.00 $\rightarrow$ RL. Each decision uses 100 target-validation examples and seed 17, followed by the 500-item test. The validation margins are 9.00 and 3.00 points. The prior source-only policy had instead chosen RL and SFT, respectively; that historical choice does not prune this replay.

Task	Target	Base	ICL	SkillOpt	SFT	RL	Gate
BIRD Mini-Dev	8B	37.00	–	–	44.20	36.80	44.20
BIRD Mini-Dev	3.6-27B	61.20	–	–	60.80	61.40	61.40
MuSR	8B	56.25	–	–	62.89	54.30	54.30
MuSR	3.6-27B	55.86	–	–	76.56	57.03	76.56
CRUXEval-O	8B	51.25	50.58	50.83	50.83	50.92	51.25
CRUXEval-O	3.8-27B	72.00	71.67	70.83	69.25	91.75	91.75
IFBench†	8B	74.00	76.33	81.33	73.33	73.33	81.33
IFBench†	3.8-27B	82.00	77.33	84.67	80.67	83.00	82.33

Table 13: Recorded accuracy (%) and the validation-gate replay. BIRD/MuSR use seed 17 and Tinker SFT/RL only; CRUXEval-O uses three-seed recipe means. † IFBench uses held-out validation folds, not final test; its fixed-method columns use the same 100 validation items. Original SFT is shown for IFBench; DeepSeek SFT is an additional candidate (71.67% / 79.67%). Dashes indicate candidates outside the replay contract.

Target	Candidate	Validation	Test	Decision
8B	SFT	**60.00**	44.20	Selected
8B	RL	51.00	36.80	Not selected
3.6-27B	SFT	67.00	60.80	Not selected
3.6-27B	RL	**70.00**	61.40	Selected

Table 14: BIRD Mini-Dev seed-17 selection inputs and subsequent test outcomes. ICL/SkillOpt use a different backend profile; the baseline has no published selection score. They therefore do not enter this two-candidate gate.

MuSR: compare two compatible training recipes. The 8B path is SFT 61.00 versus RL 63.00 $\rightarrow$ RL; the 27B path is SFT 72.00 versus RL 51.00 $\rightarrow$ SFT. Each uses 100 validation examples and seed 17, followed by the 256-item test. The 8B gate follows its 2.00-point validation advantage even though SFT later scores better on test; the method is not changed after seeing that reversal. The 27B validation margin is 21.00 points.

Target	Candidate	Validation	Test	Decision
8B	SFT	61.00	62.89	Not selected
8B	RL	**63.00**	54.30	Selected
3.6-27B	SFT	**72.00**	76.56	Selected
3.6-27B	RL	51.00	57.03	Not selected

Table 15: MuSR seed-17 selection inputs and subsequent test outcomes. ICL/SkillOpt use a different backend profile; the baseline has no published selection score. They therefore do not enter this two-candidate gate.

CRUXEval-O: average recipes, then resolve the baseline tie. Each adapted recipe contributes all three validation seeds (17, 43, 101), not its best seed. For 8B, Base and SFT-original both average 52.00%; the explicit baseline-first tie rule selects Base. For 27B, RL averages $(91 + 98 + 97)/3 = 95.33\%$, ahead of the next recipe, ICL at 77.67%. The chosen methods are then evaluated on the 400-program test. The baseline is one recorded run; its score is not replicated into artificial seed observations.

IFBench: make five separate decisions without using held-out fold labels. For each target, sort the 100 IDs by SHA-256 of `section6-crossfit-v1:` concatenated with the ID. Fold k contains sorted positions congruent to k modulo five (zero-based). For each fold, average each recipe over the other 80 examples and its three seeds; compare these means; freeze one recipe; then score the 20 held-out examples. All six recipes share the same folds. SFT-original and SFT-DS remain separate candidates because their supervision differs. The table reports every fitting mean, including losing candidates.

Target	Candidate	Val. 17/43/101	Mean val.	Test mean	Pick
8B	Base	52/–/–	**52.00**	51.25	Yes
8B	ICL	50/55/49	51.33	50.58	–
8B	SkillOpt	50/50/50	50.00	50.83	–
8B	SFT-original	52/52/52	52.00	50.83	–
8B	RL	51/51/48	50.00	50.92	–
3.8-27B	Base	75/–/–	75.00	72.00	–
3.8-27B	ICL	81/77/75	77.67	71.67	–
3.8-27B	SkillOpt	76/75/77	76.00	70.83	–
3.8-27B	SFT-original	72/74/73	73.00	69.25	–
3.8-27B	RL	91/98/97	**95.33**	91.75	Yes

Table 16: CRUXEval-O: all eligible validation inputs and held-out recipe means. Base wins the 8B tie by rule, not by consulting its test performance.

Candidate / outcome	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Base	71.25	76.25	75.00	71.25	76.25
ICL	72.92	77.08	76.67	75.83	79.17
SkillOpt	**77.92**	**82.92**	**83.75**	**79.17**	**82.92**
SFT-original	71.25	75.00	73.33	71.25	75.83
SFT-DS	66.25	74.58	72.08	72.50	72.92
RL	70.00	75.83	74.17	71.25	75.42
Frozen choice	SkillOpt	SkillOpt	SkillOpt	SkillOpt	SkillOpt
Held-out score	95.00	75.00	71.67	90.00	75.00
Held-out base	85.00	65.00	70.00	85.00	65.00

Table 17: IFBench 8B: fit on 80 examples, evaluate on 20 per fold. Candidate cells are three-seed means (Base uses one run); bold cells are the frozen validation winners. Bottom rows are held-out outcomes, not selection inputs.

Candidate / outcome	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Base	81.25	82.50	82.50	82.50	81.25
ICL	75.83	78.33	77.50	79.17	75.83
SkillOpt	**83.33**	**85.83**	83.75	**85.00**	**85.42**
SFT-original	79.58	83.33	81.67	79.58	79.17
SFT-DS	78.75	80.00	78.75	78.75	82.08
RL	81.25	83.33	**84.58**	83.75	82.08
Frozen choice	SkillOpt	SkillOpt	RL	SkillOpt	SkillOpt
Held-out score	90.00	80.00	76.67	83.33	81.67
Held-out base	85.00	80.00	80.00	80.00	85.00

Table 18: IFBench 3.8-27B: fit on 80 examples, evaluate on 20 per fold. Candidate cells are three-seed means (Base uses one run); bold cells are the frozen validation winners. Bottom rows are held-out outcomes, not selection inputs.

The 8B path selects SkillOpt five times, yielding $(95 + 75 + 71.67 + 90 + 75)/5 \approx 81.33\%$. The 27B path selects SkillOpt in folds 1, 2, 4, and 5; fold 3 selects RL because 84.58% exceeds the 83.75% of SkillOpt. Its held-out score is 82.33%. These aggregated outcomes are cross-fitted validation scores, not the official final test; ‡ in Figure 7 denotes fold-specific recipe choices, not a new ensemble.

Interpretation and reproduction. Figure 7(b) additionally shows gray BIRD/MuSR reference scores: Base is the matching Tinker test control without published selection-validation scores; ICL and SkillOpt are the OpenRouter test runs. They are displayed for context and excluded from selection, best-candidate stars, and Gap. The separate IFBench DeepSeek SFT recipe remains in the replay and detailed tables but its column is omitted from the compact figure; it is neither selected nor the best fixed candidate for either target. Figure 7(b) reports the best eligible candidate score minus the selected score. For IFBench, the reference is the best fixed recipe across the same 100 validation examples, not an oracle that changes methods in each held-out fold. Its 27B gap is 2.33

points, computed before rounding the displayed means. The four workloads have different candidate coverage, model generations, and evaluation roles; we do not pool them into a single average gain or win-rate. The full-library gate includes all available compatible candidates and cannot demonstrate that historical ordering saves probes. The recorded MuSR 8B error is a concrete counterexample to guaranteed validation transfer. Source hashes, fold membership, candidate scores, and frozen decisions are saved with the replay output. The historical-priority rule and Q2 diagnostics are proposed inputs to sequential exploration; this replay evaluates the final validation gate, not their incremental benefit.

E.3 Reference scores excluded from selection

The gray references in Figure 7(b) are listed below for completeness. Base is the Tinker test control without selector-validation scores; ICL and SkillOpt use OpenRouter. None enters selection or Gap.

Table 19: Gray reference scores (percent) in the selection figure. These supplement, rather than enlarge, the eligible candidate library.

Task	Target	Method	Backend	Score
BIRD Mini-Dev	Q3-8B	Baseline	Tinker	37.00
BIRD Mini-Dev	Q3-8B	3-shot ICL	OpenRouter	36.60
BIRD Mini-Dev	Q3-8B	SkillOpt	OpenRouter	34.20
BIRD Mini-Dev	Q3.6-27B	Baseline	Tinker	61.20
BIRD Mini-Dev	Q3.6-27B	3-shot ICL	OpenRouter	60.20
BIRD Mini-Dev	Q3.6-27B	SkillOpt	OpenRouter	58.60
MuSR	Q3-8B	Baseline	Tinker	56.25
MuSR	Q3-8B	3-shot ICL	OpenRouter	59.38
MuSR	Q3-8B	SkillOpt	OpenRouter	68.36
MuSR	Q3.6-27B	Baseline	Tinker	55.86
MuSR	Q3.6-27B	3-shot ICL	OpenRouter	55.86
MuSR	Q3.6-27B	SkillOpt	OpenRouter	73.05