FROM PAST ATTEMPTS TO BETTER DECISIONS: GENERALIZING VERIFIED EXPERIENCE THROUGH PARAMETRIC MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

A language model's first attempt on a difficult reasoning problem is often wrong. Correct answers may emerge only after repeated sampling or revision. Yet the failed and successful attempts are then discarded, so the next problem starts from scratch and the same mistakes can recur. We ask which lessons from this trial-and-error history are useful, how to store them in model parameters, and whether they generalize to unseen problems. We introduce BVEd, a parametric experience memory for frozen solvers. Starting from repeated training attempts, BVEd removes final predictions and copied answer text, then uses a teacher to distill successes and failures into candidate lessons about useful reasoning steps, missed constraints, and recurring mistakes. Source-side screens measure whether lessons increase support for the known answer, limit answer steering without the problem, and detect copied answer cues; at most one training target is selected from each admitted source. The selected lessons train a helper to transform a new, label-free problem into fresh problem-specific guidance, which the unchanged solver uses to produce its answer. Across four reasoning benchmarks, BVEd yields absolute accuracy gains of 3.09–14.87%. Together, these results suggest that trial-and-error histories can supply useful supervision: BVEd extracts experience from past attempts, consolidates it into model parameters, and generates guidance for previously unseen problems. Our code is available at an anonymous link: https://anonymous.4open.science/r/ICLR_2027-92D5

1 INTRODUCTION

Experience accumulated across multiple attempts, from initial failures to eventual success, often reveals more than the final answer alone. Failed attempts can expose missed constraints, ineffective reasoning strategies, or recurring errors, while successful ones show what ultimately helped. Together, these attempts contain signals about both what should be done and what should be avoided. Yet much of this information remains tied to the problems on which it was observed, rather than being turned into supervision that can guide reasoning on new problems [32, 15, 23]. The challenge is to distill these histories into guidance for unseen problems, rather than preserve them only as source-specific trajectories.

The difficulty is that these histories are not uniformly useful. Failed attempts may mix useful signals with incorrect reasoning, while successful ones can depend on details specific to a single problem [17, 7]. Moreover, once the outcome is known, extracted guidance may appear useful simply because it reveals or strongly suggests the answer [16, 36]. The key question is therefore **what information from past attempts is actually worth learning**. Ideally, a useful lesson should provide additional information about the correct decision when the problem is given, while carrying little information about the answer on its own. This distinction is important if experience is to remain useful beyond the problem from which it was extracted.

We introduce **BVEd**, which turns past attempts into source-screened supervision for a separate helper model. A teacher first extracts short candidate lessons from sanitized successful and failed attempts, capturing useful reasoning steps, missed constraints, and recurring failure patterns. BVEd then measures whether a candidate increases the solver's support for the known answer on its source

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.

problem, while separately checking whether it can steer the solver toward an answer when the problem context is hidden. Candidates that directly copy answer content are also removed. The retained lessons are used to train a helper that generates new, problem-specific guidance for unseen problems, allowing useful experience from past attempts to be consolidated into model parameters.

Across four primary benchmark families, guidance generated by BVEd improves direct-answering performance by 3.09% to 14.87%. We further compare against generic and mismatched guidance to test whether the gains come from problem-relevant information rather than additional context alone, and evaluate the same helpers across multiple recipient solvers to examine how the effect varies across models. These experiments quantify system-level gains, test the importance of problem–guidance correspondence, and characterize variation across recipient solvers.

We make three contributions:

- We formulate experience memory as a learned mapping from problems to guidance, enabling patterns from past attempts to inform unseen problems without retrieving the original examples.
- We develop a procedure for extracting and screening candidate lessons before training, combining usefulness on the source problem with checks for context-free answer steering and copied answer text.
- We demonstrate repeated system-level gains over direct answering on four deductive-reasoning benchmarks, and use matched text controls, pretrained-helper comparisons, repair–harm analyses, and exploratory cross-recipient evaluations to characterize the observed effects.

2 Related Work

Inference-time reasoning and refinement. Language models can improve difficult decisions by allocating additional computation to reasoning, refinement, or reusable reasoning structures [28, 20, 35, 31, 12]. Recent work further studies whether stronger inference-time behaviors can be distilled or amortized into model parameters [11]. These approaches mainly focus on improving the solution to the current problem. However, the information produced across repeated attempts is often discarded once the problem is solved, and how such information can be reused beyond the instance on which it was generated remains underexplored.

Learning from reasoning and feedback. Intermediate reasoning traces and feedback have also been widely used as supervision. Prior work learns from self-generated rationales or teacher explanations [33, 10, 24], supervises intermediate reasoning steps [14, 13, 5], or uses verification and process-level signals to improve reasoning [4]. These methods typically rely on complete rationales, step-level annotations, or already constructed preference signals. Less attention has been paid to how useful supervision can be identified from a mixture of successful and failed attempts, especially when those attempts may also contain misleading or answer-specific information.

Experience and memory. A closely related line of work enables language models and agents to reuse past experience through reflection, episodic memory, skill libraries, or retrieved insights [34, 22]. More recent methods study experience replay, multi-level reflection, memory distillation, and more selective mechanisms for retaining and reusing experience [26, 8, 18, 6, 19, 1, 30, 2, 29, 27]. These approaches show that past interactions can improve future decisions, but most keep experience as explicit memory that is stored, retrieved, or updated at inference time. It remains less clear which parts of past experience are worth learning from, and whether useful information from those experiences can be internalized without relying on retrieval when solving a new problem.

3 Method

Figure 1 summarizes the three stages of BVEd. A frozen solver samples attempts on labeled training problems, and a frozen teacher turns their sanitized traces and outcome labels into candidate lessons. Source-side tests assess the candidates' usefulness and screen for answer cues. We then distill the

selected lessons into a separate helper through supervised fine-tuning on problem–lesson pairs, using each problem as the input and its selected lesson as the target. At inference, the helper generates problem-specific guidance from a new problem alone. This guidance is prepended to the same frozen solver's prompt before the solver makes its prediction.

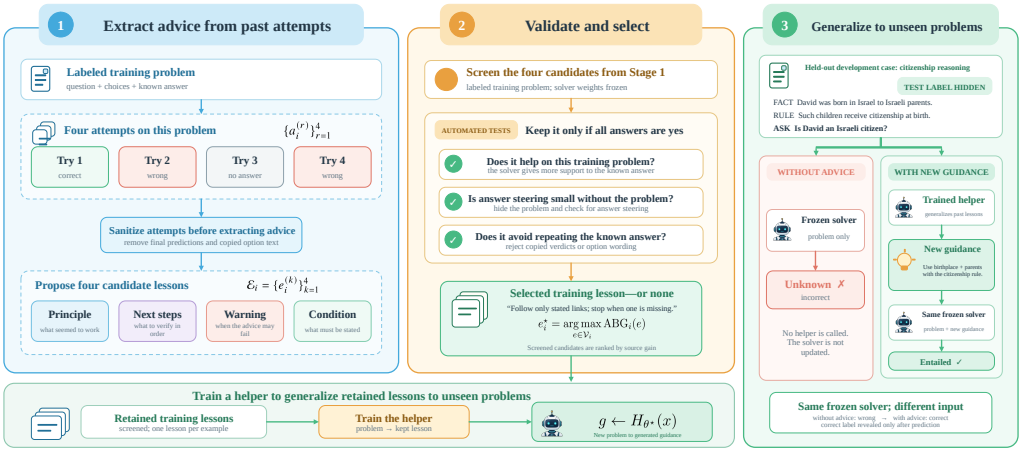

Figure 1: Overview of BVEd. (1) A frozen solver samples attempts on labeled source problems, and a frozen teacher uses their sanitized traces and outcomes to propose candidate lessons. (2) Source-side tests screen for improved support of the correct answer and for answer cues; the illustrated maximum-ABG selection yields problem–lesson pairs that train a helper. (3) For a new problem, the helper generates guidance that is prepended to the frozen solver's prompt before the solver makes its prediction. The right-hand outputs illustrate a recorded correction with abbreviated guidance, not a claim that the guidance is answer-independent.

3.1 Problem formulation and attempt histories

Let x_i denote the model-visible context and query for training problem i, including answer options if they are shown to the model. Let $\mathcal{Y}_i$ be its candidate answer set for scoring and $y_i^\star \in \mathcal{Y}_i$ its gold answer. The frozen solver S_ϕ produces K independent attempts $\{a_{i,k}\}_{k=1}^K$ on x_i; these are parallel samples, not revisions informed by earlier attempts. An answer parser compares each attempt's declared final answer with $y_i^\star$, labeling the attempt correct or incorrect when an answer can be parsed and invalid otherwise. The attempts and their outcome labels form the history $\mathcal{R}_i$.

This history places successful and unsuccessful attempts on the same problem in a common context. Correct attempts can expose useful inferences, whereas incorrect or invalid ones can reveal missing conditions, unsupported shortcuts, or failures to produce an admissible answer. The teacher uses this contrast to formulate guidance about how to reason, rather than a demonstration to copy. The solver remains frozen when scoring candidate lessons and making guided predictions; only the helper H_θ is trained.

3.2 Candidate lesson extraction

Before the teacher sees an attempt, we remove its declared final prediction and verbatim copies of complete answer options. The frozen teacher T receives the source context and query, the sanitized attempts, and their outcome categories, but not the answer-choice list or gold label. It proposes a set $\mathcal{E}_i$ of candidate lessons based on the observed approaches and their outcomes. Subsequent screening checks for answer cues that sanitization may leave behind.

The teacher can express the lesson as a reusable inference principle, an ordered sequence of checks, a warning against an observed shortcut, or a condition that must hold before a rule applies. Figure 1 depicts these forms as a principle, next steps, a warning, and a condition. They are alternative ways to use the same prior interaction history, including its observed failures. Sanitization alone does

not establish either usefulness or freedom from answer cues, so the candidates undergo source-side screening.

3.3 SOURCE SCREENING AND LESSON SELECTION

Candidate generation provides several possible interpretations of an attempt history; screening asks which of them helps the solver on the source problem without simply revealing an answer. Let $p_\phi(y \mid x, e)$ be the solver's relative score for answer y, obtained by length-normalizing answer likelihood and normalizing across the candidate answers (Appendix A). This normalization makes the comparison about support within the answer set, rather than about the raw likelihood of answer strings of different lengths. We measure the source utility of candidate $e \in \mathcal{E}_i$ by its answer-belief gain:

$$\mathrm{ABG}_i(e) = \log p_\phi(y_i^\star \mid x_i, e) - \log p_\phi(y_i^\star \mid x_i). \tag{1}$$

The two terms compare the candidate-normalized scores for the correct training answer on the same source problem, with and without the proposed lesson. Positive and negative ABG indicate increased and decreased relative support, respectively. ABG does not by itself establish the lesson's correctness or its effect on the solver's decoded prediction. The gold answer is used for this training-time scoring comparison and is not supplied to the helper at inference.

An increase in source support alone is insufficient: a lesson that simply points to an answer could also have positive ABG. To test for such context-independent steering, we compare two prompts that hide the source context and query but retain the answer set $\mathcal{Y}_i$. Using the candidate-normalized score defined above, the prompt with lesson e gives $p_\phi^{\mathrm{blind}}(\cdot \mid \mathcal{Y}_i, e)$; omitting e gives the reference distribution $p_\phi^{\mathrm{blind}}(\cdot \mid \mathcal{Y}_i)$. We compare them by Jensen–Shannon divergence:

$$b_i(e) = \mathrm{JSD}\big(p_\phi^{\mathrm{blind}}(\cdot \mid \mathcal{Y}_i, e), p_\phi^{\mathrm{blind}}(\cdot \mid \mathcal{Y}_i)\big)\,. \tag{2}$$

A small divergence indicates that the lesson-bearing prefix has little effect on answer preferences without the source premises; it does not establish that the lesson is harmless in context. A separate lexical screen $\ell_i(e)$ catches direct forms of disclosure, such as a copied choice or an explicit option or verdict cue, which a distributional comparison alone might miss. The source-utility test, blind comparison, and lexical check therefore address distinct failure modes. With τ_b as the blind-JSD cutoff and $\ell_i(e) = 0$ indicating no flagged cue, the admitted set is

$$\mathcal{V}_i = \{e \in \mathcal{E}_i : \mathrm{ABG}_i(e) > 0,\ b_i(e) \le \tau_b,\ \ell_i(e) = 0\}. \tag{3}$$

Problems with $\mathcal{V}_i = \varnothing$ contribute no training pair. For each admitted source, a selection policy chooses one lesson $e_i^\star$. Figure 1 depicts the maximum-ABG choice from $\mathcal{V}_i$; other target policies may choose from the candidates that pass the blind and lexical screens (Appendix D).

Illustrative FOLIO case. Figure 2 illustrates a FOLIO training question: Spain is in Europe and Dominique does not live in Europe, so she cannot live in Spain. Of four independent attempts, two overlook this implication and answer unknown; two use it and answer entailed. The teacher turns the sanitized attempts and their outcomes into four candidate lessons. After source-side utility and answer-cue screening, the highest-ABG eligible lesson expresses the missing inference as a counterexample check and becomes a helper-training target. The appendix gives the full record. This illustrates lesson construction, not its effect on a later test question.

3.4 PARAMETRIC DISTILLATION AND INFERENCE

Each source problem with a selected lesson contributes one supervised pair $(x_i, e_i^\star)$. We train H_θ by standard supervised fine-tuning to generate the lesson from the problem, masking the input prompt and applying the language-modeling loss to the lesson and assistant-termination tokens (Appendix A). The target is an edited summary of the attempts rather than a verbatim attempt transcript or a stand-alone gold label. The attempt history and outcome labels shape that target but are absent from the helper input. The helper must therefore learn from x_i when similar guidance may be useful, so it can guide a new problem without a new attempt history.

At inference, the trained helper generates candidate guidance for a new problem x. After removing detected explicit answer declarations and complete copies of answer choices and bounding its length,

we denote the resulting guidance by g. The frozen solver receives g as a reasoning-experience prefix to its ordinary problem prompt, with an instruction to apply it only when logically relevant, and makes a single prediction. Direct answering uses the same solver and problem prompt without this prefix (Figure 1, right). Neither model receives the held-out label; it is used only to score the prediction. Prompt and decoding settings are given in Section 4 and Appendix A.

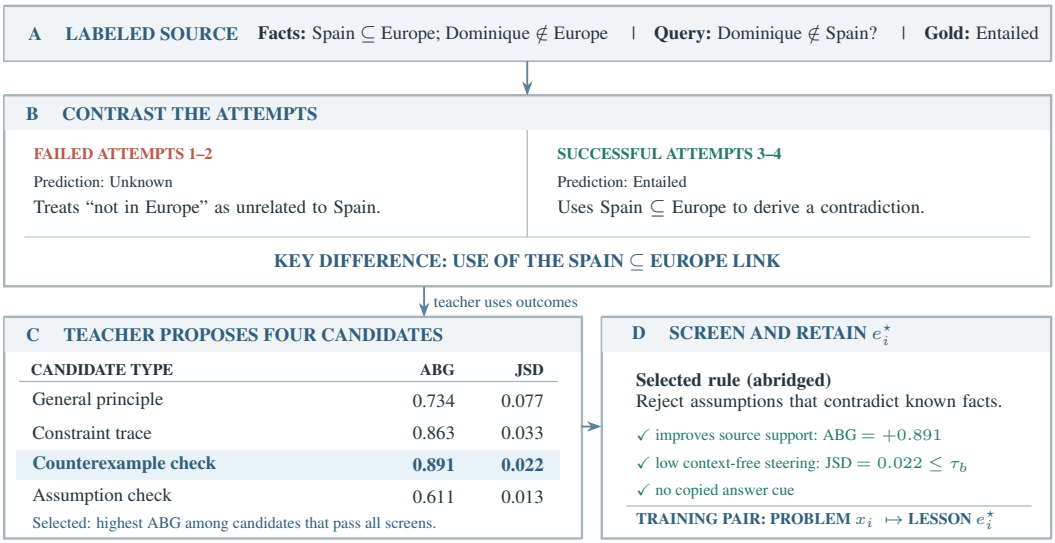

Figure 2: A FOLIO example of converting attempt outcomes into a screened lesson for helper training. Spain lies in Europe and Dominique does not live in Europe; the query asks whether she does not live in Spain. Two attempts overlook the Spain–Europe relation and answer unknown, whereas two use it and answer entailed. From the sanitized attempts and their outcomes, the teacher proposes four candidate lessons. Source-side utility and answer-cue checks identify eligible lessons; the highest-ABG candidate, shown here as a counterexample check, becomes the helper's supervised target.

3.5 Interpretation of screening scores

ABG and blind JSD quantify different effects of a candidate lesson. ABG measures the change in the correct answer's log score on its source problem. Blind JSD compares answer-score distributions with and without the lesson-bearing prefix after hiding the source context and query but retaining the answer options; a small value means that this prefix changes answer preferences little without the problem. The latter also has an information-theoretic interpretation: in an auxiliary experiment that chooses either blind prompt with equal probability and samples an answer from its candidate-normalized scores, JSD equals the mutual information between the prompt choice and the sampled answer (Appendix B).

4 Experiments

4.1 Experimental setup

Tasks. We evaluate BVEd on four deductive-reasoning benchmarks: FOLIO [9], which tests natural-language first-order inference; FLD×2 [21], a synthetic first-order deduction task; and the rule-based ProofWriter [25] and RuleTaker [3] benchmarks. Table 1 summarizes the task formats and evaluation sets. Each benchmark uses a dataset-specific helper. Training sources are disjoint from evaluation examples and benchmark groups, with no exact model-visible context in both. FOLIO and FLD×2 have separate development and final sets; the rule-based benchmarks use held-out audit panels for the main evaluation and some exploratory analyses. These checks exclude direct reuse of training items or contexts, but not semantically related templates or pretraining exposure.

Table 1: Deductive-reasoning benchmarks and primary evaluation sets. A frozen Qwen3-8B model serves as the recipient solver across all benchmarks; helper runs are independent fine-tunings on a fixed candidate pool. The benchmark-specific constructions are described in Appendix D.

Benchmark	Reasoning setting	Evaluation problems	Helper runs	Construction
FOLIO	Natural-language first-order logic	204	3	Unified
FLD×2	Synthetic first-order deduction	5000	3	Unified
ProofWriter	Synthetic rule deduction	10158	5	Safe-target variant
RuleTaker	Synthetic rule entailment	10068	5	Full-scale variant

Models and training. The helper is initialized from Qwen3-4B and fully fine-tuned as described in Section 3. The main recipient is a frozen Qwen3-8B model; separate frozen invocations of that model supply the training attempts, source-side scores, and teacher lessons. For FOLIO and FLD×2, development data fix the helper configuration before final evaluation. In the primary ProofWriter variant, each admitted source has a candidate passing all three screens, but the selected target is sampled from blind- and lexically safe candidates and can have nonpositive ABG. Optimization settings and the benchmark-specific selection policies are given in Appendices A and D.

Evaluation and uncertainty. Our primary metric is exact-match answer accuracy over the complete evaluation set: a prediction is correct only when its normalized answer matches the gold label, and invalid or unparseable outputs are counted as incorrect. To measure assistance, we rerun direct answering alongside each trained helper on the same questions and report the paired accuracy gain, defined as guided accuracy minus the corresponding direct accuracy.

In the main four-benchmark evaluation, the solver uses deterministic answer-only decoding with a 96-token limit; explicitly labeled controls and recipient analyses use the settings reported with those experiments. For FOLIO and FLD×2, the text controls are evaluated in separate paired runs from the primary evaluation. When results are aggregated over independently trained helpers, the reported $\pm$ value is the sample standard deviation across helper runs. It describes run-to-run variation in paired gains on a fixed candidate pool, including variation in the paired direct baseline. Follow-up uncertainty intervals use paired bootstrap resampling. Because FOLIO's 204 questions share 73 premise contexts, the decision-relevant FOLIO comparisons resample premise contexts rather than individual questions (Appendix F).

4.2 Baselines and controls

Direct answering prompts the frozen solver without helper output and serves as the reference for every paired comparison. To distinguish problem-specific assistance from the effect of adding text, we use two text controls: *generic text*, matched to the generated guidance in length, and *other-problem guidance*, assigned by a one-to-one permutation across different underlying premise sets or rule theories. The latter preserves actual helper outputs while breaking their correspondence to the current problem; on FOLIO, donor guidance must also come from a different premise context. Construction details are in Appendix A.

We include an *initial-helper* reference condition to examine whether the observed gains can be explained simply by introducing an additional capable model at inference time. This condition uses the same inference interface as BVEd, but replaces the fine-tuned helper with the pretrained Qwen3-4B initialization. Because the helper architecture, input format, and downstream solver interaction are kept unchanged, the comparison focuses on the additional benefit associated with training on the selected problem–lesson pairs. In this way, the initial-helper condition provides a direct reference for assessing the contribution of fine-tuning beyond the effect of merely augmenting the solver with a pretrained auxiliary model.

4.3 TRAINED-HELPER GUIDANCE IMPROVES DIRECT ANSWERING

Table 2: Accuracy of the frozen Qwen3-8B solver with and without BVEd guidance. Direct uses the problem prompt alone; BVEd adds trained-helper guidance. Direct, BVEd, and gain are means across paired runs on the same evaluation questions. Values are in %; ± denotes the sample SD of gains across helper runs.

Benchmark	Problems	Seeds	Direct	BVEd	Gain (%)
FOLIO	204	3	55.23	**70.10**	+14.87±0.28
FLD×2	5000	3	51.95	**62.55**	+10.60±0.20
ProofWriter	10158	5	27.33	**34.93**	+7.61±0.19
RuleTaker	10068	5	60.98	**64.07**	+3.09±0.09

Table 2 shows positive paired gains on all four benchmarks, ranging from 3.09% to 14.87%. The improvements are largest on FOLIO and FLD×2, while ProofWriter and RuleTaker exhibit smaller but still consistent gains. Importantly, the magnitude of improvement is not simply determined by the solver's direct-answering accuracy: ProofWriter has the lowest direct accuracy, yet it does not receive the largest benefit from guidance. This suggests that baseline difficulty alone does not explain how much assistance the helper provides. Across repeated helper runs, every trained helper improves over its paired direct baseline, and the gains show limited variation within each fixed candidate pool. The gains persist across the evaluated fine-tuning seeds under a fixed pretrained helper initialization and candidate pool. At the same time, the benchmark-specific helper constructions and source-selection procedures differ, so these results should be interpreted as evidence that BVEd provides reliable assistance under the evaluated protocols, rather than as evidence that any particular benchmark property determines the size of the gain.

Table 3: Problem-matched guidance versus text controls. Generic text is length-matched to the helper output; other-problem guidance reassigns helper outputs to different problems. Entries are paired accuracy gains over direct answering in the same run (%); ± denotes the sample SD across helper runs. FOLIO and FLD×2 use separate control runs, so their BVEd gains differ slightly from Table 2.

Benchmark	Problems	BVEd gain	Generic-text gain	Other-problem gain
FOLIO	204	**+14.54±0.28**	-1.31±0.28	-1.14±0.57
FLD×2	5000	**+10.56±0.19**	+2.17±0.16	+0.86±0.28
ProofWriter	10158	**+7.61±0.19**	-7.28±0.05	-0.65±0.45
RuleTaker	10068	**+3.09±0.09**	-0.42±0.05	+0.20±0.10

Table 3 tests whether added text produces the same gains after its relation to the current problem is removed. Length-matched generic text provides no consistent benefit and sometimes reduces accuracy. Reassigning genuine helper outputs to other problems preserves the form of the guidance but reduces the gains to between -1.14% and $+0.86\%$. Problem–guidance correspondence therefore matters in these evaluations. The generic-text control shows that the gains are not explained by simply adding a similarly sized prefix, while the other-problem control shows that preserving the form of genuine guidance is insufficient once its match to the current problem is broken. These controls therefore support the importance of problem-matched guidance, while leaving the precise source of that useful correspondence to the analyses that follow.

4.4 COMPARISON WITH A PRETRAINED HELPER

We further compare the trained helper with its pretrained initialization to assess whether experience training can provide value beyond simply introducing a capable second model at inference time. This control keeps the helper architecture and inference interface fixed, while changing whether the helper has been fine-tuned on the selected problem–lesson pairs. In a post hoc FOLIO evaluation with a Qwen3-4B recipient, learned guidance improves direct answering by $8.99 \pm 0.57\%$ and exceeds the

initial helper by 7.03% on average. Thus, in this setting, the benefit of the trained helper cannot be attributed solely to the presence of an additional pretrained model.

We further examine the reliability of this learned-over-initial difference using premise-cluster bootstrap resampling. The 95% interval for learned-minus-initial assistance has a positive lower bound for each of the three helper runs (Appendix F), indicating that the observed margin is consistent across the independently trained helpers under this evaluation protocol. Taken together, these results provide evidence that experience training can contribute additional value beyond a pretrained helper for this recipient setting. Additional recipient comparisons are reported in the appendix.

4.5 Repair–Harm Analysis of Guidance

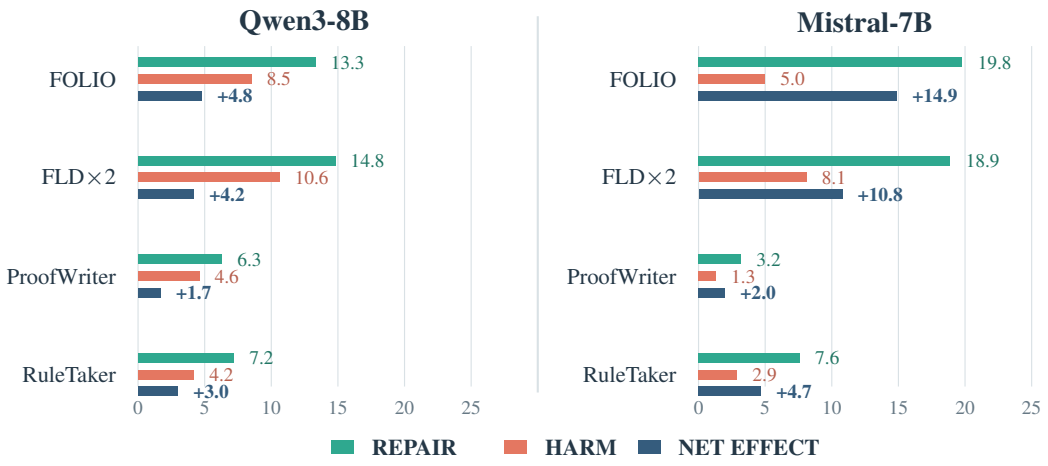

Figure 3: How guidance changes answers in a separate two-recipient development audit of Qwen3-8B and Mistral-7B with deterministic 96-token answer-only decoding. REPAIR and HARM are percentages of all evaluated questions that change from wrong to correct and correct to wrong, respectively; NET EFFECT is their difference. Each bar averages three fixed helper artifacts. Panel sizes are 215 (FOLIO), 5,000 (FLD×2), 10,158 (ProofWriter), and 10,068 (RuleTaker). The helper cohort and run protocol differ from the primary evaluation and the separate backbone audit in Table 4; values should be compared within this figure. Values are rounded independently.

Net accuracy can conceal opposing changes in individual decisions. Figure 3 therefore separates corrections of initially wrong answers from disruptions of initially correct ones in a distinct development audit with Qwen3-8B and Mistral-7B recipients. This decomposition makes it possible to examine whether the overall gain is driven primarily by repairing previous errors or is offset by substantial changes to answers that were already correct. For every displayed solver–benchmark pair, corrections exceed disruptions, yielding a positive net effect in each case, although the balance between the two varies across recipients and benchmarks. The magnitude of both repair and harm also differs across settings, indicating that guidance does not affect all recipient models or tasks in the same way. The examples in Appendix C illustrate both omitted rule chains that can be recovered with guidance and unsupported shortcuts that may lead the solver away from an initially correct decision. Together, these observations show that guidance changes the solver's decision behavior by correcting more errors than it introduces in the evaluated settings.

4.6 ASSISTANCE VARIES ACROSS RECIPIENT SOLVERS

Table 4: Exploratory five-recipient backbone audit with deterministic 96-token answer-only decoding. Entries are mean paired BVEd-minus-direct accuracy differences (%) over three auxiliary helper artifacts per benchmark (± sample SD), distinct from the five primary ProofWriter and RuleTaker runs. Development-panel sizes are 215 (FOLIO), 5,000 (FLD×2), 10,158 (ProofWriter), and 10,068 (RuleTaker).

Solver	FOLIO	FLD×2	ProofWriter	RuleTaker
Qwen3-1.7B	+9.30±0.47	+7.43±0.31	+1.81±0.08	+3.38±0.07
Phi-3.5-mini (3.8B)	+5.58±0.81	+9.54±0.28	+4.05±0.05	+4.56±0.22
Granite-3.3-2B	+10.39±0.71	+4.64±0.33	+0.47±0.02	**+4.72±0.19**
Mistral-7B-v0.3	**+17.36±1.76**	**+12.81±0.40**	+1.97±0.07	+4.69±0.13
Qwen2.5-14B	-2.02±0.97	+4.13±0.18	**+4.16±0.11**	+2.88±0.21

Table 4 evaluates five additional frozen solver models on development panels from the same four benchmark families. Within each benchmark, the same three dataset-specific helper artifacts are reused across solver models, allowing the effect of changing the recipient to be examined while keeping the guidance source fixed. Positive mean gains are observed in 95% of the evaluated solver–benchmark combinations, with improvements ranging from $+0.47\%$ to $+17.36\%$. The only negative mean occurs for Qwen2.5-14B on FOLIO ($-2.02 \pm 0.97\%$). These results indicate that the learned guidance remains broadly compatible with different solver architectures and scales under the tested development settings.

The magnitude of assistance nevertheless varies substantially across both solver models and benchmarks. The gains are not monotonic in model size: larger models do not consistently benefit more from the same helper outputs. Mistral-7B, for example, obtains double-digit gains on FOLIO and FLD×2 but considerably smaller improvements on ProofWriter and RuleTaker, while the other solvers exhibit different benchmark-specific profiles. The negative FOLIO result for Qwen2.5-14B further shows that guidance that is effective for several solver models need not improve every model on every task. This variation suggests that the effect of guidance depends on the interaction between the supplied information and the recipient solver's existing reasoning behavior, rather than on model scale alone.

Taken together, the backbone audit extends the evaluation beyond the main Qwen3-8B recipient and shows that the observed assistance is not confined to a single solver. At the same time, the variation across solver–benchmark pairs motivates recipient-aware evaluation, since the same helper guidance can produce different levels of benefit depending on the model that receives it.

5 CONCLUSION

We introduced BVEd, which extracts concise lessons from successful and failed solver attempts, evaluates them with source-side criteria for usefulness and answer cues, and distills selected lessons into a problem-conditioned helper. This allows information revealed by past successes and failures to guide a new problem without retrieving the original attempts or modifying the frozen solver. Across FOLIO, FLD×2, ProofWriter, and RuleTaker, helper-generated guidance improves the solver's accuracy over direct answering by 3.09%–14.87%. Generic text and guidance from unrelated problems do not produce comparable gains, suggesting that the improvements depend on guidance relevant to the current problem rather than on extra text alone. In separate evaluations with Qwen3-8B and Mistral-7B, guidance turns more wrong answers into correct ones than correct answers into wrong ones. Together, these results provide evidence that experience is more than the final answer: useful reasoning steps, missed constraints, and recurring failure patterns distributed across multiple attempts can be consolidated into guidance that improves later decisions.

6 AI USE STATEMENT

Generative AI tools were used for language polishing, including improving the grammar, clarity, and readability of the manuscript. They were not used to generate research ideas, formulate the methodology, or draw scientific conclusions. All AI-assisted edits were carefully reviewed and, where necessary, revised by the authors. The authors take full responsibility for the accuracy, originality, and final content of the manuscript.

7 ETHICS STATEMENT

This work investigates experience-based guidance for language-model reasoning on existing deductive-reasoning benchmarks. We describe the use of benchmark data, the construction of model-generated supervision, and the evaluation procedures to support transparency and responsible interpretation of the findings.

8 REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe the method and experimental setup. Appendices A and D provide details of candidate generation, screening, helper training, and benchmark-specific evaluation protocols. Appendix F describes the FOLIO uncertainty analysis. An anonymous code repository is linked in the abstract to support inspection and reproduction of the implementation.

REFERENCES

[1] Zouying Cao, Jiaji Deng, Li Yu, Weikang Zhou, Zhaoyang Liu, Bolin Ding, and Hai Zhao. Remember me, refine me: A dynamic procedural memory framework for experience-driven agent evolution. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 16803–16822, 2026.

[2] Zihao Cheng, Zeming Liu, Yingyu Shan, Xinyi Wang, Xiangrong Zhu, Yunpu Ma, Hongru Wang, Yuhang Guo, Wei Lin, and Yunhong Wang. Mem2evolve: Towards self-evolving agents via co-evolutionary capability expansion and experience distillation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20784–20831, 2026.

[3] Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. *arXiv preprint arXiv:2002.05867*, 2020.

[4] Mengyi Deng, Xin Li, Duyi Pan, Zilin Wang, Zhiwei Li, Zhijiang Guo, and Wei Wang. Reskill: Explicit failure attribution and structured repair for interactive language agents. *arXiv preprint arXiv:2609.15684*, 2026.

[5] Yuchen Deng, Shichen Fan, Naibo Wang, Xinkui Zhao, and See Kiong Ng. Agentpro: Enhancing llm agents with automated process supervision. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9992–10017, 2025.

[6] Yubin Ge, Salvatore Romeo, Jason Cai, Monica Sunkara, and Yi Zhang. Samule: Self-learning agents enhanced by multi-level reflection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 16602–16621, 2025.

[7] Yifu Guo, Jiaye Lin, Huacan Wang, Yuzhen Han, Sen Hu, Ziyi Ni, Licheng Wang, and Mingguang Chen. Se-agent: Self-evolution trajectory optimization in multi-step reasoning with llm-based agents. *Advances in Neural Information Processing Systems*, 38:116314–116341, 2026.

[8] Priyanshu Gupta, Shashank Kirtania, Ananya Singha, Sumit Gulwani, Arjun Radhakrishna, Gustavo Soares, and Sherry Shi. Metareflection: Learning instructions for language agents using past reflections. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8369–8385, 2024.

[9] Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, et al. Folio: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22017–22031, 2024.

[10] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex 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*, pp. 8003–8017, 2023.

[11] Audrey Huang, Adam Block, Dylan Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. In *International Conference on Learning Representations*, volume 2025, pp. 76687–76739, 2025.

[12] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International conference on learning representations*, volume 2024, pp. 32808–32824, 2024.

[13] Zeyu Jia, Alexander Rakhlin, and Tengyang Xie. Do we need to verify step by step? rethinking process supervision from a theoretical perspective. *arXiv preprint arXiv:2502.10581*, 2025.

[14] Dongwei Jiang, Guoxuan Wang, Yining Lu, Andrew Wang, Jingyu Zhang, Chuyu Liu, Benjamin Van Durme, and Daniel Khashabi. Rationalyst: Pre-training process-supervision for improving reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26547–26566, 2025.

[15] Hyunseok Lee, Seunghyuk Oh, Jaehyung Kim, Jinwoo Shin, and Jihoon Tack. Revise: Learning to refine at test-time via intrinsic self-verification. *arXiv preprint arXiv:2502.14565*, 2025.

[16] Jungseob Lee, Seungyoon Lee, Suhyune Son, Dongyub Jude Lee, Sungbin Han, Sugyeong Eo, and Heuiseok Lim. Answer-conditioned chains of thought degrade verifiable-reasoning distillation in large language models. *arXiv preprint arXiv:2607.14552*, 2026.

[17] Nayoung Lee, Ziyang Cai, Avi Schwarzschild, Kangwook Lee, and Dimitris Papailiopoulos. Self-improving transformers overcome easy-to-hard and length generalization challenges. *arXiv preprint arXiv:2502.01612*, 2025.

[18] Yitao Liu, Chenglei Si, Karthik R Narasimhan, and Shunyu Yao. Contextual experience replay for self-improvement of language agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14179–14198, 2025.

[19] Wenquan Ma, Jiayan Nan, and Wenlong Wu. What deserves memory: Adaptive memory distillation for llm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34789–34812, 2026.

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

[21] Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. Enhancing reasoning capabilities of llms via principled synthetic logic corpus. *Advances in Neural Information Processing Systems*, 37:73572–73604, 2024.

[22] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

[23] Vishnu Sarukkai, Zhiqiang Xie, and Kayvon Fatahalian. Self-generated in-context examples improve llm agents for sequential decision-making tasks. *Advances in Neural Information Processing Systems*, 38:64392–64425, 2026.

[24] Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. Distilling reasoning capabilities into smaller language models. *arXiv preprint arXiv:2212.00193*, 2022.

[25] Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, 2021.

[26] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023. *URL https://arxiv. org/abs/2305.16291*, 2(11), 2023.

[27] Tong Wang, Pei Xu, Shiyue Cao, Likun Yang, Daipeng Li, Jianbin Jiao, and Kaiqi Huang. Samem: State-aware memory as a fine-grained memory for llm agents in decision-making. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14691–14710, 2026.

[28] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[29] Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts llm agents: An empirical study of experience-following behavior. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 623–645, 2026.

[30] Cheng Yang, Xuemeng Yang, Licheng Wen, Daocheng Fu, Jianbiao Mei, Rong Wu, Pinlong Cai, Yufan Shen, Nianchen Deng, Jia Xu, et al. Towards self-evolving agents: Enabling autonomy through interactive experience refinement. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 30424–30451, 2026.

[31] Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E Gonzalez, and Bin Cui. Buffer of thoughts: Thought-augmented reasoning with large language models. *Advances in Neural Information Processing Systems*, 37:113519–113544, 2024.

[32] Ruimeng Ye, Zihan Wang, Yang Xiao, Zinan Ling, Manling Li, and Bo Hui. Weak-to-strong generalization with failure trajectories. In *International Conference on Learning Representations*, volume 2026, pp. 150477–150496, 2026.

[33] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

[34] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

[35] Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Denny Zhou, Swaroop Mishra, Huaixiu S Zheng, et al. Self-discover: Large language models self-compose reasoning structures. *Advances in Neural Information Processing Systems*, 37:126032–126058, 2024.

[36] Ruiyang Zhou, Shuozhe Li, Amy Zhang, and Liu Leqi. Expo: Unlocking hard reasoning with self-explanation-guided reinforcement learning. *Advances in Neural Information Processing Systems*, 38:90989–91016, 2026.

A TRAINING AND SCREENING DETAILS

Candidate generation. For each training source $(x_i, y_i^\star)$, the frozen solver samples independent attempts. The benchmark-specific parser extracts a candidate answer when possible; exact-match comparison with $y_i^\star$ then categorizes each attempt as correct, incorrect, or invalid. Declared final predictions and verbatim answer choices are removed before the frozen teacher proposes candidate lessons. The teacher receives the source context and query, sanitized attempts, and outcome categories; the answer-choice list and gold label are omitted from its prompt. Table 5 collects the generation, screening, and training settings.

Table 5: Hyperparameters for lesson construction, helper training, and inference. The unified tasks are FOLIO and FLD×2; the rule tasks are ProofWriter and RuleTaker. Source selection and repetitions are given in Appendix D.

Stage	Setting	Value
Attempts	Model	Frozen Qwen3-8B solver
	Samples per source	4 independent attempts
	Sampling	Temperature 0.7; top-p 0.9
	Output limit	192 tokens
Teacher	Model	Frozen Qwen3-8B teacher
	Lessons per source	4 candidates
	Sampling	Temperature 0.6; top-p 0.9
	Output limits	192 tokens; 90 words
Screening	Blind-JSD cutoff τ_b	0.0864 nats
Helper SFT	Model	Qwen3-4B; all parameters trainable
	Optimizer	8-bit AdamW
	Epochs	3
	Learning rate	10^{-6}
	Weight decay	0.1
	Adam betas	$(0.9, 0.95)$
	Warmup	3% linear
	Gradient clipping	1.0
	Sequence cap, unified tasks	4,096 tokens
	Sequence cap, rule tasks	2,048 tokens
	Microbatch / accumulation	1 / 16 steps
Inference	Helper output limit	192 tokens
	Solver output	Deterministic, answer-only; 96 tokens

Scoring and screening. For candidate answer y with L_y tokens, the scorer uses

$$s_\phi(y \mid x, e) = \frac{1}{L_y} \sum_{t=1}^{L_y} \log P_\phi(y_t \mid x, e, y_{<t}), \qquad p_\phi(y \mid x, e) = \frac{\exp s_\phi(y \mid x, e)}{\sum_{z \in \mathcal{Y}} \exp s_\phi(z \mid x, e)}. \tag{4}$$

This is a candidate-normalized score, not a calibrated probability over unrestricted outputs. Source ABG measures the change in log score for the training answer. For blind JSD, both prompts state that the original question and premises are hidden, list the complete answer options in their original order, and request only the complete text of the best option. The answer set and decoding request are identical; the treatment adds a lesson-bearing prefix that is absent from the reference prompt.

The treatment serialization is: *"Potentially useful reasoning experience:"*, a newline, the candidate lesson, two newlines, *"The original question and premises are hidden."*, two newlines, one line per answer in the form *"Option j: [complete answer text]"*, two newlines, and *"Return only the complete text of the best option."* The reference serialization omits only the first heading, lesson, and following blank line.

Before admission, text is case-folded and reduced to word tokens. The lexical screen rejects a candidate if it contains a complete normalized answer choice; explicitly selects a numbered or lettered

option; names an ordinal choice such as "the second answer"; states "the answer is" followed by a complete choice; or assigns a verdict such as *entailed*, *contradicted*, *true*, *false*, or *unknown* to the current query. A candidate enters $\mathcal{V}_i$ only if $\mathrm{ABG}_i(e) > 0$, $b_i(e) \le \tau_b$, and the lexical screen is negative. FFOLIO, FLD×2, and RuleTaker choose a fully screened target from this set. For ProofWriter, a source is admitted only if at least one candidate passes all three screens; its target is then selected from blind- and lexically safe candidates and may have nonpositive ABG (Appendix D). The blind-JSD cutoff in Table 5 is operational rather than optimized. These checks screen source usefulness and context-free answer influence, not the truth of every lesson or transfer. Accordingly, source admission requires a candidate that passes all three screens; this does not imply that every selected ProofWriter target passes the source-utility screen.

Helper training and selection. Within each microbatch, causal cross-entropy is averaged over all non-masked lesson and assistant-termination tokens; prompt and padding tokens carry ignore labels. Gradients are then averaged over the actual number of microbatches in each accumulation window, including the final partial window. In the unified fraction sweep, retained sources are ranked by source ABG and the chosen top fraction is capped at 4,096; the training order is then shuffled from the run seed. Development selection evaluates retained-source fractions $\{1, 0.5, 0.25\}$ and guidance limits $\{90, 45\}$ under a prespecified random seed, maximizing

$$\min\{A_{\mathrm{guide}} - A_{\mathrm{direct}},\ A_{\mathrm{guide}} - \max(A_{\mathrm{generic}}, A_{\mathrm{shuffled}})\}. \tag{5}$$

Here A_c is development accuracy under condition c on the same panel. The selected configuration is fixed before evaluation and retrained with three random seeds on the same candidate pool and pretrained helper base. These runs do not vary the pretrained initialization or repeat candidate construction. Supervised fine-tuning matches the construction because each admitted source has one target. Rejected candidates are not treated as coherent preference negatives because rejection may arise from answer cues, nonpositive source utility, or the selector's within-source choice.

At test time, the helper receives the current problem and generates guidance under the length limits in Table 5. A lexical postprocessor removes complete answer-choice copies and detected option-selection or final-answer patterns; the resulting g is then supplied to the frozen, answer-only solver.

Text controls. The generic control repeats the benchmark-neutral instruction "Work methodically identify the task list each explicit condition trace only supported steps consider plausible alternatives verify the conclusion against all provided information avoid unstated assumptions and check the final response carefully." It is truncated to match the generated guidance in word count for the primary runs and in solver-token count for the separate FOLIO and FLD×2 control runs. The other-problem control permutes the saved helper outputs one-to-one while requiring different underlying groups: a FOLIO story, an FLD×2 fact set, or a ProofWriter/RuleTaker rule theory. Thus the donor text is genuine helper output but does not come from the recipient problem's group.

B Information-theoretic interpretation of the screening scores

Fix a source i, candidate lesson e, and answer set $\mathcal{Y}_i$. Equation (1) is the log ratio of the correct answer's candidate-normalized scores with and without e; it is not generally pointwise conditional mutual information because the no-lesson score is not defined by marginalizing over random lessons. For the blind comparison, let $Q_0(y) = p_\phi^{\mathrm{blind}}(y \mid \mathcal{Y}_i)$ and $Q_1(y) = p_\phi^{\mathrm{blind}}(y \mid \mathcal{Y}_i, e)$ be distributions on the same candidate set under the complete reference and treatment prompts. In an auxiliary experiment, draw $Z \sim \mathrm{Bernoulli}(1/2)$ to select the prompt and $A \mid Z = z \sim Q_z$ to sample an answer from its candidate-normalized scores. Since $\Pr(A = y) = M(y) := [Q_0(y) + Q_1(y)]/2$,

$$I(Z; A) = \tfrac{1}{2} D_{\mathrm{KL}}(Q_0 \| M) + \tfrac{1}{2} D_{\mathrm{KL}}(Q_1 \| M) = \mathrm{JSD}(Q_0, Q_1) = b_i(e). \tag{6}$$

This identity concerns the constructed blind score-sampling experiment, not the solver's decoded answer or information about the gold label; it does not establish freedom from answer cues or transfer to unseen problems.

Table 7: Decision-level effects on the four primary benchmarks. Repair denotes a transition from an incorrect direct prediction to a correct assisted prediction; harm denotes the converse transition. Both are unconditional percentages of the evaluation panel, so gain equals repair minus harm up to rounding. FOLIO and FLD×2 average three helper seeds; ProofWriter and RuleTaker average five helper seeds under their specified protocols. Generic and other-problem controls follow the corresponding benchmark protocol.

Task	Guidance	Repair %	Harm %	Gain %
FOLIO	BVEd	**18.46**	3.92	**+14.54±0.28**
FOLIO	Generic text	2.29	3.59	-1.31±0.28
FOLIO	Other problem	4.90	6.05	-1.14±0.57
FLD×2	BVEd	**19.09**	8.53	**+10.56±0.19**
FLD×2	Generic text	5.48	3.31	+2.17±0.16
FLD×2	Other problem	8.51	7.65	+0.86±0.28
ProofWriter	BVEd	**13.99**	6.38	**+7.61±0.19**
ProofWriter	Generic text	1.13	8.41	-7.28±0.05
ProofWriter	Other problem	7.10	7.75	-0.65±0.45
RuleTaker	BVEd	**7.38**	4.30	**+3.09±0.09**
RuleTaker	Generic text	1.86	2.28	-0.42±0.05
RuleTaker	Other problem	3.42	3.21	+0.20±0.10

C MECHANISM CONTROLS AND CONCRETE CASES

Table 6: Behavioral attribution hierarchy. Each layer constrains plausible explanations without establishing internal faithfulness.

Attribution target	Comparison	Observation	Interpretation
Text presence	Generic text and guidance from another problem	Neither matches the BVEd gains.	Problem alignment contributes to efficacy.
Training contribution	Initial vs. trained helper on post hoc FOLIO / Qwen3-4B	The trained helper exceeds the initial helper in this setting.	Evidence for an incremental training benefit in this task–recipient setting only.
Decision transitions	Repair/harm counts and saved cases	Guidance corrects more errors than it introduces for every audited pair.	The guidance channel is beneficial on average but imperfect.

Observed development cases. The following cases come from saved development predictions, not reconstructed training-source selection chains. They illustrate how the helper's generated text can affect a particular decision; no candidate-level ABG or blind-JSD score is available for these test-time outputs.

FOLIO: a repair and a harm. For the citizenship question shown in Figure 1 (saved example `folio::633`), the direct solver answered *unknown*, whereas the guided solver answered the gold label *entailed*. The displayed guidance is abbreviated: the saved helper output also stated the inferred citizenship conclusion. This case therefore documents a correction, not answer-cue independence. It follows the benchmark's intended reading of "citizenship at birth" as Israeli citizenship; the case illustrates recorded model behavior rather than independently validating that semantic inference. In a separate case, the direct solver correctly answered *unknown* when asked whether the band White Town was successful. The helper treated a song's chart performance as evidence of the band's success, an inference not licensed by the premises, and its saved output explicitly said that the conclusion was supported. The guided solver then answered *entailed*. This case illustrates an error that includes a direct verdict cue.

Table 8: A recorded FLD×2 development repair (example `001450`). The statement, hypothesis, and guidance are quoted from saved server outputs; the guidance itself reveals the contradiction, so the case illustrates a decision change rather than answer-cue independence.

Source fact	"Everything does unbridle 1960s." (fact 4)
Hypothesis	"The fact that everything unbridles 1960s does not hold."
Logical relation	The hypothesis negates fact 4: fact 4 states P, while the hypothesis states $\neg P$.
Helper output	"Fact4 states that everything does unbridle 1960s. The hypothesis claims that the fact that everything unbridles 1960s does not hold, which directly contradicts Fact4."
Solver result	Direct: *unknown* (incorrect). With guidance: *disproved* (correct).

The helper output also elicits *disproved* when evaluated without the premises, so this example does not establish that generated guidance is answer-independent. Candidate-level ABG and blind-JSD scores are unavailable for this test-time output. These saved cases show corrections and errors under guidance; they do not establish that the solver executes the generated text faithfully.

D BENCHMARK-SPECIFIC PROTOCOLS

Paired-run accounting. The retained seed-level summaries imply FOLIO direct correct counts of $112, 113, 113$ and guided counts of $143, 143, 143$ on the same 204 questions; FLD×2 gives direct counts $2597, 2595, 2601$ and guided counts $3118, 3123, 3142$ on the same 5,000 questions. Subtracting within run reproduces Table 2: FOLIO $14.87 \pm 0.28\%$ and FLD×2 $10.60 \pm 0.20\%$.

This appendix specifies the benchmark-specific helper constructions. All four benchmarks evaluate a frozen Qwen3-8B recipient with 96-token answer-only decoding at temperature zero, exact-match scoring, and invalid outputs counted as incorrect. FOLIO and FLD×2 use the unified maximum-ABG construction with three independently trained helpers on fixed candidate pools. ProofWriter first admits sources with at least one candidate passing all three screens, then selects uniformly among that source's blind- and lexically safe candidates, including candidates with nonpositive ABG; its five runs retrain the same target set. RuleTaker instead selects the maximum-ABG lesson among candidates passing all three screens, uses the full selected source set without a fraction sweep, and retrains that fixed set with five seeds. Results are interpreted within each benchmark because these constructions differ.

The unified campaign uses 789 training sources, 215 development questions, and 204 final questions for FOLIO; 636 sources pass screening and all 636 train the selected configuration. FLD×2 uses 4,096 training sources, 5,000 development questions, and 5,000 final questions; 2,028 sources pass screening and 507 train the selected configuration. Each development/final pair is disjoint despite FLD×2 having equal-sized sets. The ProofWriter and RuleTaker main runs instead evaluate on their held-out audit panels of 10,158 and 10,068 questions, respectively; the additional recipient and decision-change analyses also use those audit panels. The shared panels make those analyses exploratory rather than independent replications of the primary result.

E DIRECT-REASONING AND TRAINING CONTROLS

Table 9: FOLIO evaluation on smaller solvers using dataset-specific Qwen3-4B helpers without target-solver training. Initial helper denotes the original pretrained instruction model. Positive intervals count seeds whose premise-cluster-paired 95% lower bound for BVEd minus the initial helper is above zero (10,000 draws over 73 contexts). These are post hoc comparisons, and decoding protocols differ by row.

Solver	Solver protocol	Δ Direct (%)	Δ Initial (%)	Positive intervals	Min. valid %
Qwen3-4B	Answer-only / 96	+8.99±0.57	+7.03	3/3	95.59
Qwen3-1.7B	Visible CoT / 512	+13.73±0.00	+3.10	0/3	98.53
Phi-3.5-mini (3.8B)	Visible CoT / 512	+8.66±1.42	+5.07	1/3	99.51

The answer-only Qwen3-4B row provides the largest observed learned-over-pretrained margin, whereas the visible-reasoning rows primarily measure assistance compatibility.

F FOLIO PREMISE-CLUSTER UNCERTAINTY

FOLIO contains 204 test questions over 73 distinct premise contexts. The analysis therefore draws 73 premise clusters with replacement 10,000 times and recomputes the question-weighted paired accuracy difference in each replicate. Exact visible premise contexts define the clusters; all questions remain included, and predictions are held fixed. Table 10 reports the two comparisons for which this correction is decision-relevant: the three primary helpers and the three Qwen3-4B learned-versus-pretrained comparisons.

Table 10: Premise-cluster bootstrap for the key FOLIO comparisons. Paired differences are reported in %, with 95% intervals over 10,000 cluster resamples.

Recipient	Helper seed	Δ Direct [95% CI]	Δ Initial [95% CI]
Qwen3-8B	1	+15.20 [+9.00, +21.53]	–
Qwen3-8B	2	+14.71 [+8.81, +21.05]	–
Qwen3-8B	3	+14.71 [+8.87, +20.69]	–
Qwen3-4B	1	+8.33 [+0.95, +15.81]	+6.37 [+0.51, +12.56]
Qwen3-4B	2	+9.31 [+2.06, +16.59]	+7.35 [+1.60, +13.33]
Qwen3-4B	3	+9.31 [+2.05, +16.67]	+7.35 [+1.55, +13.40]

All six direct-assistance intervals remain positive after clustering, as do the three Qwen3-4B learned-versus-pretrained intervals. Dashes mark comparisons not evaluated under the corresponding protocol. This analysis corrects the sampling unit for these comparisons; it does not measure variability from candidate construction or complete retraining of the helper.

G LIMITATIONS AND REPRODUCIBILITY NOTES

Training labels characterize training attempts and score source candidates; they never enter helper inputs, solver prompts, or test-time selection. Sanitization and blind-bias screening limit answer steering but do not prove non-leakage. Because the helper receives the problem itself, answer-recoverable guidance alone does not imply access to the evaluation label.

G.1 A FOLIO TRAINING-TO-TEST ILLUSTRATION

Figure 2 shows a training source and a schematic one-line paraphrase, not the verbatim supervised target. A distinct held-out question and saved predictions illustrate the subsequent system behavior, without attributing that outcome to this single training lesson.

Training source and attempts. Spain is in Europe; Dominique does not live in Europe. The source asks whether she does not live in Spain (gold: *entailed*). Four independent solver attempts were sampled: two answered *unknown*, one treating non-European residence as unrelated to Spain; two answered *entailed* using Spain's inclusion in Europe. Other premises do not affect this inference.

Candidate generation and selection. The teacher saw the source question, sanitized attempts, and success/failure categories, but not the gold answer. Its four lessons (general principle, constraint trace, counterexample check, assumption check) had respective source ABG scores of 0.734, 0.863, 0.891, and 0.611, and blind-JSD scores of 0.077, 0.033, 0.022, and 0.013. The complete selected counterexample target reads: "Successful route: Dominique not living in Europe, and Spain being in Europe, implies she cannot live in Spain. Failure pitfall: Confusing non-European residence with non-Spanish residence. Countermodel: If Dominique lived in Spain, she would live in Europe, contradicting her non-European residence." Its unrounded source ABG was 0.890842 and blind JSD was 0.022285; the lexical screen did not flag it, and the lesson was retained as a supervised target.

Held-out question and generated guidance. The later FOLIO question has a different context and asks whether "No one playing for Nautico is Brazilian." One premise states that Ailton Silva is a Brazilian footballer who plays for Nautico. He is therefore a counterexample to the proposed universal claim, whose gold label is *contradicted*. The trained helper, given this new problem rather than the Dominique lesson or its attempt history, generated the following saved postprocessed guidance g, which was supplied to the solver:

> "Ailton Silva is identified as a Brazilian footballer (Premise 3), and he plays for Nautico (Premise 3). Therefore, the conclusion that 'No one playing for Nautico is Brazilian' is contradicted by the fact that at least one player (Ailton Silva) for Nautico is Brazilian."

The saved postprocessing record marks no lexical removal for this guidance: it selects no option and copies no complete answer choice, but states an answer-equivalent conclusion. The case records a repair under the actual solver input, not answer-cue independence or faithful reproduction of a source lesson.

Saved solver predictions. For this same held-out question, the frozen solver answered *unknown* without guidance and *contradicted* with the trained helper's guidance. In the corresponding paired controls, generic text and guidance reassigned from another problem both led to *unknown*:

Solver input	Saved prediction	Correct?
Problem only	Unknown	No
Problem + trained-helper guidance	Contradicted	Yes
Problem + generic text	Unknown	No
Problem + other-problem guidance	Unknown	No

The training and test questions belong to disjoint groups and share no visible context. This record shows a source lesson admitted for training and a later decision repaired by generated guidance. Because the helper is trained on the full set of selected source lessons, this example should be interpreted as a system-level illustration rather than as evidence that the lesson alone caused the repair. The aggregate paired evaluations provide the benchmark-level evidence for BVEd's effect.