LEARNING TO USE WHAT MODELS KNOW: KNOWLEDGE RETROSPECT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models have made substantial progress on complex reasoning tasks, yet their performance remains less reliable when solving knowledge-intensive problems. In these settings, a model must not only access relevant facts, but also express them as intermediate premises during multi-step chain-of-thought reasoning. We study this gap through the lens of knowledge dispatch, focusing on whether accessible knowledge is actually invoked in the generated reasoning trajectory. To this end, we introduce a factorized diagnostic framework that separates isolated knowledge accessibility from in-chain knowledge use. Our analysis shows that knowledge points answered correctly in isolation are often not expressed during free-form reasoning. Further diagnostics indicate that these failures are more frequently associated with omission than incorrect invocation, and that the model can often use the omitted knowledge when it is explicitly provided. These findings suggest a knowledge dispatch bottleneck: relevant knowledge may be available to the model, but not reliably surfaced during generation. Motivated by this diagnosis, we propose **Knowledge Retrospect Learning** (KRL), an on-policy self-distillation framework that uses knowledge-augmented contexts during training to provide distributional supervision for problem-only reasoning. The student samples trajectories from the original problem context, while a knowledge-conditioned teacher retrospectively evaluates the same trajectories and transfers token-level signals to the student. Experiments on multiple knowledge-intensive reasoning benchmarks show that KRL improves reasoning accuracy, suggesting that training-time knowledge conditioning can help models surface relevant knowledge more reliably during generation.

1 INTRODUCTION

Recent advances in large language models, driven by chain-of-thought reasoning, test-time scaling, and reinforcement learning, have led to strong progress on complex logical and mathematical reasoning tasks (Wei et al., 2022; Wang et al., 2022; DeepSeek-AI, 2025; OpenAI, 2024). However, performance on knowledge-intensive reasoning remains more limited. Compared with mathematical reasoning, these tasks require the model not only to possess relevant facts, but also to bring them into the reasoning process at the right time and organize them into intermediate premises that later steps can use.

In particular, this issue is especially pronounced in long chain-of-thought (CoT) reasoning (Wei et al., 2022). As illustrated in Fig. 1, a model may correctly answer related facts in isolation, yet still fail to consistently integrate them into its multi-step reasoning. As a result, relevant knowledge is not always preserved as an intermediate premise, and the resulting trajectory may leave essential reasoning steps implicit. This indicates that the challenge is not limited to the final answer, but also concerns whether relevant knowledge can be sustained and utilized throughout generation. Since such failures are not always apparent from the final output alone, analyzing the reasoning trajectory becomes necessary (Wu et al., 2025; Arcuschin et al., 2025).

To better understand this breakdown, we construct a factorized diagnostic framework that separates knowledge accessibility from in-chain knowledge use. Concretely, we assess whether the model can access a required knowledge point under isolated conditions, and whether that same knowledge point is explicitly invoked during CoT reasoning. This decomposition allows us to examine the transition

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.

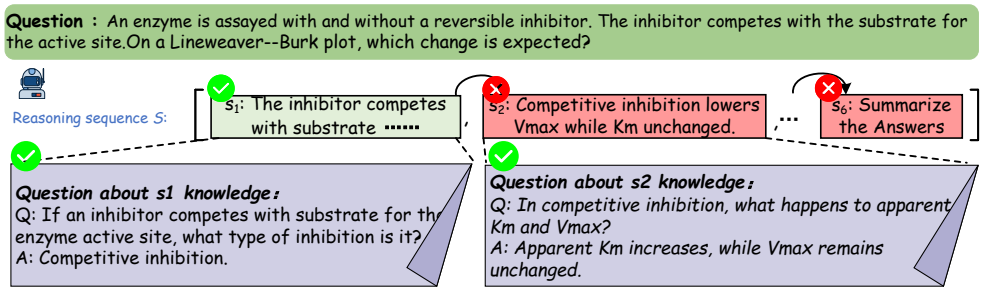


Figure 1: An illustrative example of inconsistent knowledge use in chain-of-thought reasoning. The model answers direct questions about the relevant knowledge correctly, but uses one of the same knowledge points incorrectly within the generated reasoning sequence.

from accessible knowledge to knowledge that is actually used, rather than relying on the final answer as the sole indicator of failure.

After conducting comprehensive analyses, we find that knowledge accessible in isolation does not translate reliably into explicit use in the reasoning chain. The gap between accessibility and usage is substantial, and it does not disappear simply because the model answers the knowledge point correctly in isolation. More importantly, when the relevant knowledge is explicitly provided in context, the model is able to make better use of it during reasoning. These results suggest that the main difficulty lies less in whether the knowledge is present and more in whether it is effectively dispatched during free-form generation. We refer to this phenomenon as the *knowledge dispatch bottleneck*.

Motivated by this observation, we propose **Knowledge Retrospect Learning** (KRL), a training approach designed to internalize knowledge dispatch behavior into model parameters. The central idea is to use knowledge-conditioned contexts as privileged supervision for trajectories generated from problem-only prompts, so that comparable behavior can be reproduced without external prompting. To this end, KRL uses on-policy self-distillation, following prior work on knowledge distillation for language models (Hinton et al., 2015; Agarwal et al., 2024; Gu et al., 2024), to supervise student trajectories and encourage more reliable knowledge use during complex reasoning.

We evaluate KRL on multiple knowledge-intensive reasoning benchmarks. The results show that KRL improves reasoning accuracy without requiring external knowledge prompts at inference time. Ablation studies further indicate that the improvement stems from task-aligned knowledge conditioning rather than from simply increasing context length or adding unrelated information.

Our main contributions are summarized as follows:

- We identify the *knowledge dispatch bottleneck* in CoT reasoning, showing that knowledge accessible in isolation is not always expressed in the generated reasoning chain.
- Through factorized empirical analysis, we characterize the failure mode and show that the main issue is the mismatch between knowledge accessibility and knowledge use during free-form generation.
- We propose Knowledge Retrospect Learning (KRL), an on-policy self-distillation method that internalizes knowledge-dispatch behavior into model training.

2 Disentangling Knowledge Accessibility and In-Chain Knowledge Use

Reasoning models still exhibit substantial performance gaps on knowledge-intensive tasks, but the source of this behavior is not immediately clear. Poor performance may reflect insufficient access to required knowledge, incorrect use of that knowledge, or a mismatch between what the model can answer under isolated knowledge-point assessment and what it expresses during long-form reasoning.

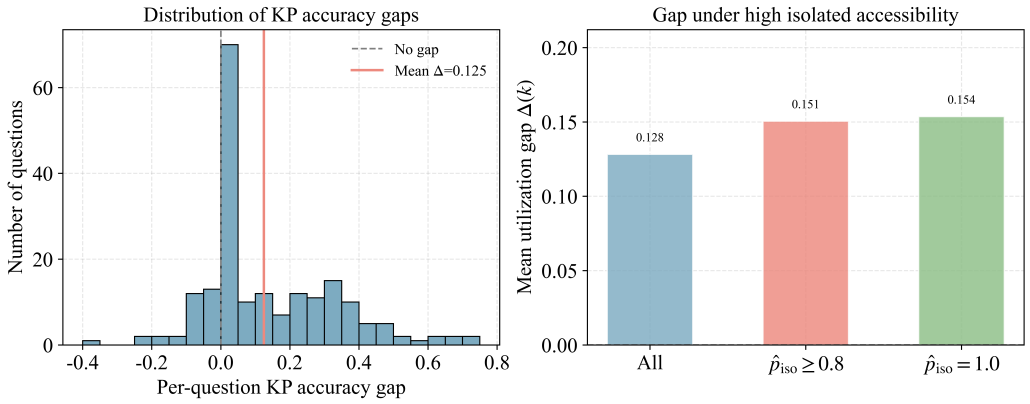


Figure 2: Knowledge utilization gap between isolated assessment and in-chain knowledge use. Left: distribution of per-question knowledge-point accuracy gaps. Right: the single-knowledge-point gap remains positive even when restricting to knowledge points with high isolated accessibility.

To investigate this phenomenon, we decouple knowledge accessibility from knowledge use in generated reasoning chains. This factorized diagnostic allows us to examine whether required knowledge is available under isolated assessment, whether it appears in the model's reasoning trajectory, and what form the discrepancy takes when the two evaluations diverge.

2.1 Does Reasoning Degrade Knowledge Utilization?

Knowledge-intensive reasoning entangles knowledge availability with the model's ability to use that knowledge during generation. To separate these factors, we first examine whether isolated knowledge accessibility is preserved when the same knowledge is required inside a generated reasoning chain.

Setting. To examine this gap at the knowledge-point level, we sample 150 questions from GPQA-Diamond and construct a factorized diagnostic framework with two aligned assessment paths. For each question q, we extract a set of atomic knowledge points required to solve the question, denoted by $\mathcal{K}(q)$. Extraction is performed by a multi-stage LLM-based pipeline; the full diagnostic protocol is provided in Appendix A.

The first path, **Isolated Assessment**, measures isolated accessibility. For each $k \in \mathcal{K}(q)$, we convert the knowledge point into a standalone query, implemented as a multiple-choice knowledge-point assessment that targets the fact expressed by k outside the original reasoning context. We estimate $\hat{p}_{\text{iso}}(k)$ as the empirical accuracy of the model on this standalone query. This estimates whether the model can access the required knowledge when the knowledge demand is made explicit.

The second path, **In-Chain Assessment**, measures in-chain knowledge use, defined as whether the same knowledge point is correctly used in the generated explicit CoT. For the original question q, we sample model-generated CoT trajectories and judge, for each trajectory, whether k appears as a correct premise in the reasoning process. We estimate $\hat{p}_{\text{chain}}(k)$ as the fraction of sampled trajectories in which k is judged as correctly used. This estimates whether the knowledge is realized during open-ended reasoning.

We define the knowledge utilization gap as

$$\Delta(k) = \hat{p}_{\text{iso}}(k) - \hat{p}_{\text{chain}}(k).$$

Because the two assessment paths differ in format, $\Delta(k)$ should be interpreted as a behavioral gap between isolated accessibility and explicit use during free-form generation. The isolated path primarily probes recognition under an explicit standalone knowledge query, whereas the in-chain path requires the model to retrieve, verbalize, and integrate the same knowledge point within an open-ended reasoning trajectory. In Fig. 2, we report both a question-level summary and a knowledge-point-level analysis: the former shows how the gap is distributed across problem instances, while the latter examines whether the gap persists among knowledge points with high isolated accessibility.

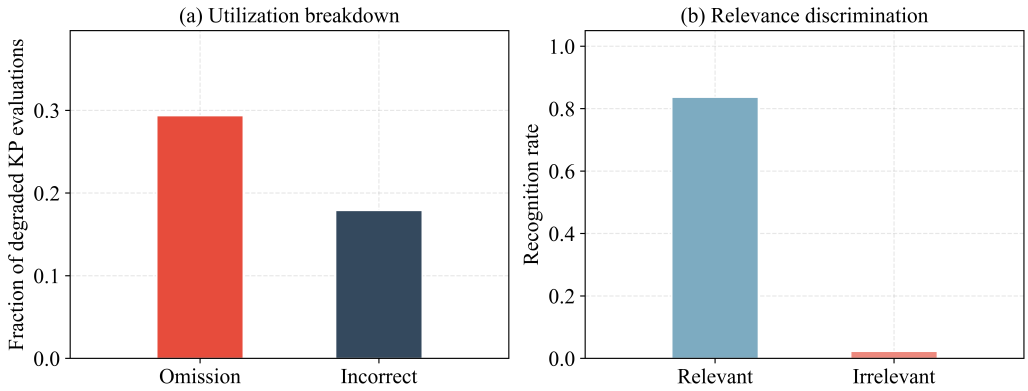


Figure 3: Failure-mode and relevance diagnostics. (a) Among degraded knowledge-point evaluations, omission is more common than incorrect invocation. (b) Under explicit knowledge-point context, the model distinguishes relevant omitted knowledge points from irrelevant controls.

Observation. As illustrated in Fig. 2, the per-question gap distribution is right-skewed: questions with stronger isolated knowledge-point accuracy than in-chain knowledge use are common. This pattern indicates that knowledge-intensive underperformance cannot be explained solely by the absence of required knowledge.

The degradation also persists when the analysis is restricted to knowledge points that the model answers reliably under isolated assessment. Thus, stronger isolated accessibility does not guarantee that the same knowledge will appear in the CoT when the original problem requires it. This result motivates a more specific question: when knowledge use breaks down inside the chain, what observable form does that breakdown take?

2.2 What Is the Dominant Mode of Utilization Breakdown?

The previous subsection establishes a systematic gap between isolated accessibility and in-chain knowledge use. To characterize the nature of this gap, we further examine whether the missing utilization mainly reflects incorrect use of relevant knowledge, or whether the knowledge is absent from the generated reasoning trajectory.

Setting. We categorize each in-chain knowledge-point outcome into three mutually exclusive types: *correct invocation*, *incorrect invocation*, and *omission*. Correct invocation indicates that the knowledge point is used as a valid premise in the reasoning chain; incorrect invocation indicates that the knowledge point is mentioned or applied in an erroneous way; omission indicates that the knowledge point does not appear in the explicit text of the generated trajectory.

The interpretation of omission requires care. It does not claim that the knowledge is absent from the model's internal representations. Rather, it records whether the knowledge is written into the observable reasoning trajectory. This distinction matters because CoT reasoning relies on visible intermediate premises: if a relevant fact is not verbalized, later generated steps cannot reliably cite or extend it as an explicit premise.

Observation. As shown in Fig. 3a, among knowledge-point evaluations where utilization drops from isolated assessment to in-chain reasoning, omission is more common than incorrect invocation. The main observable pattern is therefore not the invocation of an incorrect version of the relevant knowledge, but the absence of that knowledge from the explicit reasoning chain.

This restricted analysis narrows the diagnostic focus: rather than primarily concerning how the model handles a fact once invoked, the discrepancy concerns why a relevant fact that is available under isolated assessment does not appear in the explicit trajectory.

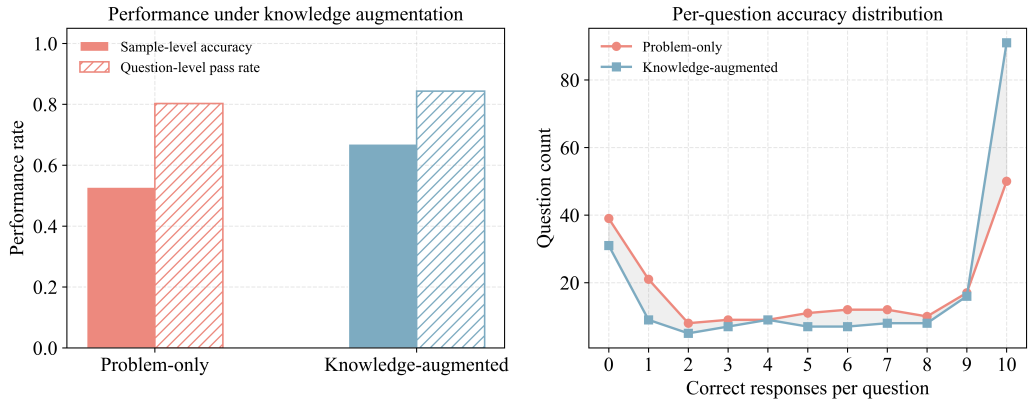


Figure 4: Post-hoc relevant knowledge prompting. Compared with the problem-only baseline, explicitly providing relevant knowledge points yields higher sample-level accuracy and question-level pass rate, and the per-question distribution shifts toward larger numbers of correct responses.

2.3 What Drives Systematic Knowledge Omission?

The preceding results identify a more specific phenomenon: some knowledge points are available under isolated assessment but absent from the generated reasoning trajectory. This leads to the diagnostic question of what accounts for such omissions.

Setting. One possibility is **association failure**: the model does not recognize that the omitted knowledge point is relevant to the current question. If this explanation is dominant, then even when the knowledge point is explicitly placed in the context, the model would be expected to show low relevance-judgment accuracy.

Another possibility is **explicitization failure**: the model can recognize or use the knowledge when it is explicitly provided, but does not reliably surface that knowledge in the free-form reasoning trajectory. If this explanation is dominant, then making the knowledge explicit would be expected to support relevance recognition and improve downstream reasoning performance.

To distinguish these explanations, we design two targeted diagnostics. **Relevance Probing** presents each omitted knowledge point together with the original question and requires the model to judge whether the knowledge is needed or useful for solving the problem. This diagnostic tests a strong form of Association Failure: whether the model still fails to associate the knowledge with the task even after the knowledge has been placed in context.

Knowledge Prompting prepends the relevant knowledge points to the original question and evaluates the model under this knowledge-augmented context. If omission reflects a difficulty in surfacing relevant knowledge during free-form generation, explicitly providing that knowledge would be expected to improve performance. We also construct an irrelevant-prompting control with matched prompt format and number of added knowledge points to separate task-relevant knowledge effects from generic effects of longer prompts or changed formatting.

Observation. First, relevance probing yields high recognition rates for omitted knowledge. As shown in Fig. 3b, once omitted knowledge points are explicitly shown together with the original question, the model recognizes most of them as relevant. This result is not fully consistent with association failure as a complete explanation: when the knowledge is placed in context, the model frequently retains the behavioral capacity to link it to the task.

Second, explicitly providing relevant knowledge is associated with higher reasoning performance. Fig. 4 shows higher sample-level accuracy and question-level pass rate under relevant knowledge prompting than under the problem-only baseline. This comparison indicates that the observed omission is unlikely to be explained solely by an inability to use these knowledge points; for a meaningful fraction of cases, the model can incorporate the knowledge once it is made explicit in the working context.

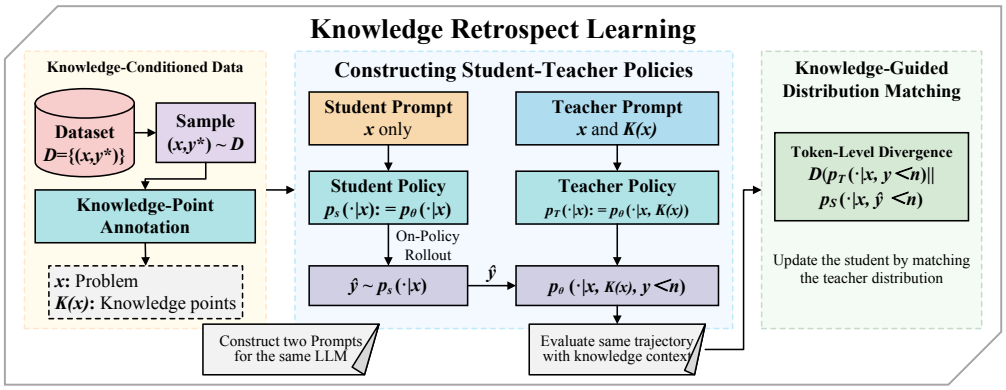


Figure 5: Overview of Knowledge Retrospect Learning. The student generates on-policy trajectories from the problem-only context, while a knowledge-conditioned teacher retrospectively evaluates the same prefixes and provides token-level distributional supervision.

Together, these diagnostics are more consistent with a generation-side explicitization problem than with a lack of knowledge access or a complete inability to recognize relevance: in many cases, the model can recognize omitted knowledge as relevant and use it when explicitly provided.

⋆ Takeaway

The factorized analysis identifies a systematic discrepancy between isolated knowledge accessibility and explicit knowledge use during reasoning. **This discrepancy indicates a knowledge dispatch bottleneck: accessible and task-relevant knowledge points are more often omitted from the generated reasoning trajectory than incorrectly invoked.**

3 METHOD

Motivated by the knowledge dispatch bottleneck identified in Section 2, we view effective reasoning as requiring not only access to relevant knowledge, but also its activation and expression during generation. To transfer this behavior without using knowledge hints at inference time, we propose *Knowledge Retrospect Learning*, a contextual student–teacher framework that uses knowledge-conditioned contexts as privileged supervision for problem-only reasoning. As illustrated in Figure 5, the student samples trajectories from the original problem context, while a knowledge-conditioned teacher retrospectively evaluates the same prefixes and provides token-level distributional targets.

3.1 KNOWLEDGE-CONDITIONED TRAINING DATA

For each problem x, let $K(x) = \{k_1, \ldots, k_m\}$ denote the task-relevant knowledge points identified by the diagnostic pipeline in Section 2. The resulting knowledge-conditioned training set is

$$\mathcal{D}_K = \{(x_i, K(x_i))\}_{i=1}^{N}, \tag{1}$$

where $K(x)$ defines the privileged teacher context rather than a target solution.

3.2 CONTEXTUAL STUDENT–TEACHER POLICIES

Knowledge Retrospect Learning defines two conditional policies that differ only in their conditioning context. The student policy is conditioned on the problem-only prompt:

$$p_S(\cdot \mid x) \triangleq p_\theta(\cdot \mid c_S(x)), \tag{2}$$

where $c_S(x)$ contains the original problem and the standard reasoning instruction, thereby aligning the training-time student condition with the inference-time condition.

The teacher policy is conditioned on a knowledge-augmented prompt:

$$p_T(\cdot \mid x, K(x)) \triangleq p_{\tilde{\theta}}(\cdot \mid c_T(x, K(x))), \tag{3}$$

where $c_T(x, K(x))$ contains the same problem together with the knowledge-point set $K(x)$ and an instruction to assess their relevance for solving the problem.

The formulation does not rely on an externally stronger teacher model. Instead, the teacher denotes the model distribution induced by the privileged knowledge context. Consequently, the student and teacher differ in the information available through the prompt, and the teacher provides only next-token distributions rather than an independently generated solution.

3.3 On-Policy Knowledge Retrospection

Given a problem x, a response is sampled from the current student policy under the problem-only context:

$$\hat{y} = (\hat{y}_1, \ldots, \hat{y}_{|\hat{y}|}) \sim p_S(\cdot \mid x). \tag{4}$$

The resulting sequence is therefore on-policy with respect to the model being optimized. Both policies subsequently evaluate this same trajectory. At each generated position t, the student and teacher induce next-token distributions over the vocabulary $\mathcal{V}$ conditioned on the same student prefix:

$$p_S(\cdot \mid x, \hat{y}_{<t}), \qquad p_T(\cdot \mid x, K(x), \hat{y}_{<t}). \tag{5}$$

This construction aligns the training signal with the level at which the bottleneck is observed. The omissions identified in Section 2 arise along the model's own autoregressive trajectory rather than on an externally prescribed trajectory. Accordingly, the objective is evaluated on prefixes that the student actually visits under the problem-only context. The teacher then provides a retrospective distributional assessment of these prefixes under explicit knowledge conditioning.

3.4 Knowledge-Guided Distribution Matching

The student is optimized by matching its next-token distribution to the teacher distribution along the student-generated trajectory. For a sampled response $\hat{y}$, the trajectory-level matching loss is

$$\ell_{\mathrm{KRL}}(x, K(x), \hat{y}) = \frac{1}{|\hat{y}|} \sum_{t=1}^{|\hat{y}|} D\Big(p_T(\cdot \mid x, K(x), \hat{y}_{<t}) \,\Big\|\, p_S(\cdot \mid x, \hat{y}_{<t}) \Big). \tag{6}$$

The divergence D is instantiated as a temperature-scaled generalized Jensen–Shannon divergence; the main configuration uses teacher-to-student KL matching. The full objective is provided in Appendix C.

To improve optimization stability, unusually large vocabulary-level contribution terms are clipped before being summed into the position-level divergence. This preserves dense full-vocabulary supervision while reducing the influence of isolated outlier vocabulary entries.

4 Experiments

4.1 Experimental Setup

Models. We evaluate KRL and comparison methods on Qwen3-1.7B, Qwen3-4B, and Qwen3-8B (Yang et al., 2025) since they represent diverse parameter scales that are widely used in reasoning tasks. All experiments are run on a server with 8 NVIDIA L20 GPUs.

Datasets. We evaluate on four knowledge-intensive benchmarks (GPQA-Diamond (Rein et al., 2023), MMLU-Pro (Wang et al., 2024), SuperGPQA (M-A-P Team et al., 2025), and TheoremQA (Chen et al., 2023)) and three out-of-domain logical or mathematical benchmarks (Log_D (Suzgun et al., 2022), AIME24 (AIME, 2025), and MATH500 (Hendrycks et al., 2021)); details are provided in Appendix B.

Table 1: **Accuracy (%) on knowledge-intensive and out-of-domain logical and mathematical reasoning benchmarks.**

Model	Method	GPQA-D	MMLU-Pro	SuperGPQA	TheoremQA	Log_D	AIME24	MATH500	Avg.
Qwen3 1.7B	Base	35.5	59.4	30.5	48.8	89.3	47.5	90.5	57.4
	+ Vanilla SFT	29.2	48.4	19.4	29.2	76.0	15.8	79.3	42.5
	+ GRPO	41.3	59.7	30.7	48.7	90.5	44.2	90.0	57.9
	+ KRL	**43.7**	**60.6**	**31.0**	**49.6**	**94.3**	**50.8**	**91.4**	**60.2**
Qwen3 4B	Base	55.9	72.5	43.0	58.7	99.2	73.7	**95.2**	71.2
	+ Vanilla SFT	39.2	60.5	29.4	39.5	95.3	35.8	89.7	55.6
	+ GRPO	56.4	72.5	41.8	58.7	99.1	74.2	94.4	71.0
	+ KRL	**57.6**	**72.7**	**43.5**	**59.5**	**99.9**	**75.8**	**95.2**	**72.0**
Qwen3 8B	Base	62.0	75.9	46.8	59.9	99.7	**76.7**	**95.9**	73.8
	+ Vanilla SFT	45.6	64.5	34.3	39.8	93.6	41.7	86.5	58.0
	+ GRPO	60.2	75.9	46.7	**60.2**	99.6	74.2	95.5	73.2
	+ KRL	**62.9**	**76.5**	**46.9**	60.1	**99.9**	**75.2**	**95.8**	**73.9**

Training Data. Training uses approximately 9.6k problems sampled from Open-SciReasoning-R1 (NVIDIA Corporation, 2025), augmented with task-relevant knowledge points for constructing the privileged teacher context; additional details are given in Appendix B.

Baselines. We compare KRL with the following baseline methods: ***Vanilla SFT***: standard supervised finetuning on the $(\text{problem}, \text{solution})$ pairs, serving as the off-policy reference; ***GRPO*** (Shao et al., 2024): on-policy reinforcement learning driven by final-answer reward, representing the outcome-reward-only training paradigm.

4.2 Main Results

Table 1 reports results on knowledge-intensive and out-of-domain logical and mathematical benchmarks. Vanilla SFT substantially underperforms the Base row, with average scores of 42.5, 55.6, and 58.0 across the three model scales, suggesting that standard solution-level SFT is not the most effective adaptation strategy.

KRL improves over Vanilla SFT by 17.7, 16.4, and 15.9 average points, and outperforms GRPO by 2.3, 1.0, and 0.7 points, indicating that knowledge-conditioned trajectory-level supervision provides an effective post-training signal. Compared with Base, KRL improves Qwen3-1.7B and Qwen3-4B by 2.8 and 0.8 points and remains close for Qwen3-8B, where Base is slightly stronger on some high-scoring benchmarks such as AIME24 and MATH500.

4.3 Ablation Study

To analyze the design factors in KRL, we conduct ablation studies on trajectory source and privileged-context relevance, as summarized in Tables 2 and 3. These ablations use Qwen3-1.7B, where the main-result table shows the largest room for improvement.

Table 2: **Ablation on trajectory source.**

Training setting	GPQA-D	MMLU-Pro	SuperGPQA	TheoremQA	Log_D	AIME24	MATH500	Avg.
Vanilla SFT	29.2	48.4	19.4	29.2	76.0	15.8	79.3	42.5
Off-policy trajectory distillation	38.8	58.7	30.4	49.4	91.1	50.2	90.3	58.4
On-policy trajectory distillation	**43.7**	**60.6**	**31.0**	**49.6**	**94.3**	**50.8**	**91.4**	**60.2**

Trajectory source. The off-policy variant uses trajectories sampled from DeepSeek-R1-0528 as fixed distillation trajectories, while the on-policy variant applies distillation to trajectories generated by the current student. Both variants substantially improve over Vanilla SFT, indicating that knowledge-conditioned distributional supervision is beneficial even when applied to fixed trajectories. However, on-policy trajectory distillation further increases the average score from 58.4 to 60.2. This result is consistent with the motivation of KRL: the dispatch bottleneck arises along the states visited by the student during autoregressive generation, so supervising those states provides a more aligned training signal.

Table 3: **Ablation on privileged-context relevance.**

Teacher context	GPQA-D	MMLU-Pro	SuperGPQA	TheoremQA	Log_D	AIME24	MATH500	Avg.
Shuffled knowledge context	37.8	57.2	28.2	45.3	89.6	40.8	88.4	55.3
Irrelevant knowledge context	35.2	54.8	25.9	41.7	86.8	35.0	86.2	52.2
Relevant knowledge context	**43.7**	**60.6**	**31.0**	**49.6**	**94.3**	**50.8**	**91.4**	**60.2**

Privileged-context relevance. We compare teacher contexts with relevant, irrelevant, and shuffled knowledge. The relevant context contains task-related knowledge points for the problem. The irrelevant context uses knowledge sampled from unrelated questions while preserving the presence of additional knowledge text. The shuffled context perturbs the semantic alignment between the problem and the provided knowledge. Relevant knowledge context achieves an average score of 60.2, compared with 52.2 for irrelevant context and 55.3 for shuffled context. This gap suggests that the improvement is not explained by adding more text to the teacher prompt alone; the privileged context must be semantically aligned with the problem.

Design implication. Together, these ablations indicate that both trajectory alignment and context relevance contribute to the effectiveness of KRL. The results support the view that knowledge-conditioned supervision is most useful when it is applied to the student's own reasoning states and when the teacher context contains task-relevant knowledge rather than unrelated information.

4.4 Discussion

The experiments support two main observations. First, KRL yields stable gains on the in-domain knowledge-intensive benchmarks, indicating that knowledge-conditioned supervision improves the setting targeted by our diagnosis. Second, on out-of-domain logical and mathematical benchmarks, KRL remains competitive with Base and does not exhibit severe degradation, suggesting that the training signal does not substantially compromise broader reasoning ability. The ablation results further indicate that both the trajectory source and the semantic relevance of the privileged context affect the effectiveness of this transfer.

Taken together, these findings are consistent with the central motivation of Knowledge Retrospect Learning: training-time knowledge conditioning can be used to improve problem-only generation, without requiring external knowledge hints during inference.

5 Related Work

Prior work improves language-model reasoning through chain-of-thought prompting, related decoding strategies (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2022), or retrieval-augmented evidence at inference time (Lewis et al., 2020), but does not directly isolate how accessible knowledge is used within generated reasoning trajectories.

Related diagnostic studies reduce reliance on memorized knowledge through linguistic reasoning tasks (Bean et al., 2024; Khouja et al., 2025), study reasoning under knowledge updates (Li & Goyal, 2025), classify knowledge and reasoning dependence (Thapa et al., 2025), or analyze factual knowledge and faithfulness in CoT traces (Wu et al., 2025; Arcuschin et al., 2025). We complement them by directly comparing isolated accessibility with in-chain knowledge use.

6 Conclusion

This paper studies a knowledge dispatch bottleneck: models may possess task-relevant knowledge yet fail to activate and express it in generated reasoning. We introduce Knowledge Retrospect Learning, a contextual student–teacher framework that uses knowledge-augmented teacher contexts to provide distributional supervision for problem-only student trajectories, preserving inference without external knowledge annotations. Experiments show that KRL improves reasoning accuracy, while ablations confirm the importance of on-policy trajectories and task-aligned privileged context. These findings suggest training-time knowledge conditioning as a promising way to help reasoning models surface relevant knowledge during free-form generation.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024.

AIME. AIME problems and solutions, 2025. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. Chain-of-thought reasoning in the wild is not always faithful, 2025. URL https://arxiv.org/abs/2503.08679.

Andrew Michael Bean, Simeon Hellsten, Harry Mayne, Jabez Magomere, Ethan A. Chi, Ryan Andrew Chi, Scott A. Hale, and Hannah Rose Kirk. LINGOLY: A benchmark of olympiad-level linguistic reasoning puzzles in low-resource and extinct languages. In *The Thirty-Eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=cLga8GStdk.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset, 2023. URL https://arxiv.org/abs/2305.12524.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

Jude Khouja, Lingyi Yang, Karolina Korgul, Simi Hellsten, Vlad Neacşu, Harry Mayne, Ryan Othniel Kearns, Andrew Bean, and Adam Mahdi. LINGOLY-TOO: Disentangling reasoning from knowledge with templatised orthographic obfuscation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=CQIkN2uuBr.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2022.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2020. URL https://arxiv.org/abs/2005.11401.

Aochong Oliver Li and Tanya Goyal. Memorization vs. reasoning: Updating llms with new knowledge, 2025. URL https://arxiv.org/abs/2504.12523.

M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Kang Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixin Deng, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, Yizhi Li, Yunwen Li, Dehua Ma, Yuansheng Ni, Haoran Que, Qiyao Wang, Zhoufutu Wen, Siwei Wu, Tianshun Xing, Ming Xu, Zhenzhu Yang, Zekun Moore Wang, Junting Zhou, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines, 2025. URL https://arxiv.org/abs/2502.14739.

NVIDIA Corporation. Opensciencereasoning-2. https://huggingface.co/datasets/nvidia/OpenScienceReasoning-2, 2025. Synthetic multi-domain reasoning dataset released by NVIDIA.

OpenAI. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. *CoRR*, abs/2311.12022, 2023.

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.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Rahul Thapa, Qingyang Wu, Kevin Wu, Harrison Zhang, Angela Zhang, Eric Wu, Haotian Ye, Suhana Bedi, Nevin Aresh, Joseph Boen, Shriya Reddy, Ben Athiwaratkun, Shuaiwen Leon Song, and James Zou. Disentangling reasoning and knowledge in medical large language models, 2025. URL https://arxiv.org/abs/2505.11462.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2022.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

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, 2022.

Juncheng Wu, Sheng Liu, Haoqin Tu, Hang Yu, Xiaoke Huang, James Zou, Cihang Xie, and Yuyin Zhou. Knowledge or reasoning? a close look at how llms think across domains, 2025. URL https://arxiv.org/abs/2506.02126.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025.

A Diagnostic Evaluation Protocol

Benchmark and target model. The diagnostic analysis in Section 2 uses 150 questions sampled from GPQA-Diamond and evaluates DeepSeek-R1-0528-Qwen3-8B as the target model. We evaluate the target model under two aligned views of the same required knowledge: an isolated knowledge-point view and an in-chain reasoning view. This protocol complements final-answer evaluation by identifying discrepancies between isolated knowledge accessibility and explicit knowledge use during generation.

Knowledge-point extraction. For each question q, we extract a set of atomic knowledge points $\mathcal{K}(q) = \{k_1, \ldots, k_m\}$ required for solving the question. Each k_i is a concise declarative statement corresponding to a fact, concept, relation, theorem, or domain-specific principle. The extraction pipeline asks a strong LLM to identify the necessary knowledge points, normalize them into atomic statements, and remove duplicated or overly broad items. These extracted knowledge points define the shared evaluation units used by both assessment paths.

Isolated Assessment. For each extracted knowledge point $k_i \in \mathcal{K}(q)$, we convert the knowledge point into a standalone query that does not include the original question q or its reasoning context. In the experiments, this query is implemented as a multiple-choice knowledge-point assessment: the correct option expresses the fact encoded by k_i, while the distractor options are plausible but incorrect alternatives from the same domain. We then query the target model on this standalone item. If S_{iso} samples are drawn for the isolated query, isolated accessibility is estimated as

$$\hat{p}_{\text{iso}}(k_i) = \frac{1}{S_{\text{iso}}} \sum_{s=1}^{S_{\text{iso}}} \mathbf{1}[a_{i,s} = a_i^\star], \tag{7}$$

where $a_i^\star$ is the correct option for the knowledge-point query and $a_{i,s}$ is the model answer in sample s. In the reported analysis, we use $S_{\text{iso}} = 5$ samples per knowledge-point query. This assessment estimates whether the model can answer the targeted knowledge demand when it is presented independently of the original reasoning problem.

In-Chain Assessment. For the same question q, we sample S_{chain} complete CoT trajectories from the target model under the original problem prompt. Each pair (k_i, r_s) is judged according to whether the knowledge point appears and is used correctly in the generated reasoning trajectory r_s. We use three mutually exclusive outcome categories: *correct invocation*, if k_i is used as a valid premise; *incorrect invocation*, if k_i is mentioned or applied incorrectly; and *omission*, if k_i does not appear in the explicit trajectory. The in-chain knowledge-use frequency is estimated as

$$\hat{p}_{\text{chain}}(k_i) = \frac{1}{S_{\text{chain}}} \sum_{s=1}^{S_{\text{chain}}} \mathbf{1}[y_{i,s} = \text{correct}], \tag{8}$$

where $y_{i,s}$ denotes the judged outcome for knowledge point k_i in trajectory r_s. In the reported analysis, we use $S_{\text{chain}} = 10$ CoT trajectories per question, sampled with temperature 0.6.

Knowledge utilization gap. The single-knowledge-point gap is defined as $\Delta(k_i) = \hat{p}_{\text{iso}}(k_i) - \hat{p}_{\text{chain}}(k_i)$. We also compute a question-level summary by averaging over knowledge points:

$$\Delta(q) = \frac{1}{|\mathcal{K}(q)|} \sum_{k_i \in \mathcal{K}(q)} \Delta(k_i). \tag{9}$$

The knowledge-point-level statistic keeps the two assessment paths aligned around the same item k_i, while the question-level statistic summarizes how the discrepancy accumulates within a problem instance.

Relevance probing. For knowledge points labeled as omissions in the generated CoT, we construct a relevance-probing query by presenting the original question together with the omitted knowledge point. The model is asked to judge whether the knowledge point is required or clearly useful for solving the question. This diagnostic evaluates whether an omitted knowledge point can still be behaviorally associated with the task once it is explicitly placed in context.

Knowledge prompting and matched control. To test whether omitted knowledge can improve reasoning once made explicit, we prepend the relevant knowledge points to the original question and ask the model to reason again. We also construct an irrelevant-prompting control with the same prompt format and the same number of added knowledge points, but using knowledge points sampled from other questions. This matched control separates the effect of task-relevant knowledge from nonspecific effects of longer prompts or changed formatting.

B Experimental Details

Evaluation benchmarks. The main experiments use seven benchmarks. GPQA-Diamond (Rein et al., 2023), MMLU-Pro (Wang et al., 2024), SuperGPQA (M-A-P Team et al., 2025), and TheoremQA (Chen et al., 2023) evaluate knowledge-intensive reasoning across scientific, professional, and theorem-based domains. Log_D, the logical-deduction subset from BIG-Bench Hard (Suzgun et al., 2022), evaluates structured symbolic reasoning. AIME24 (AIME, 2025) and MATH500 (Hendrycks et al., 2021) evaluate mathematical reasoning outside the primary knowledge-intensive setting.

Training data. For training data construction, we use approximately 9.6k problems sampled from Open-SciReasoning-R1 (NVIDIA Corporation, 2025). Each problem is paired with a long CoT trajectory generated by a strong reasoning model. We additionally annotate the atomic knowledge points $\mathcal{K}(q)$ required to solve each problem using the same multi-stage extraction pipeline as in Section 2.3. These annotations are used only in the teacher prompt during training; the student never receives external knowledge hints at inference time.

Evaluation setup. All main evaluations use a unified decoding setting with temperature 0.6, top-p 0.95, four sampled responses per test problem, and a maximum generation length of 32,768 tokens. We report average accuracy on each benchmark.

C Method Details

The divergence D in Eq. 6 is instantiated as a temperature-scaled generalized Jensen–Shannon divergence. For $0 < \beta < 1$,

$$\mathrm{JSD}_\beta(p_T \| p_S) = \beta D_{\mathrm{KL}}(p_T \| m) + (1-\beta) D_{\mathrm{KL}}(p_S \| m), \quad m = \beta p_T + (1-\beta) p_S. \tag{10}$$

Endpoint settings reduce to KL-style distillation objectives; the main configuration uses teacher-to-student KL matching. The overall objective is

$$\mathcal{L}_{\mathrm{KRL}}(\theta) = \mathbb{E}_{(x,K(x))\sim\mathcal{D}_K}\left[\mathbb{E}_{\hat{y}\sim p_S(\cdot|x)}\left[\ell_{\mathrm{KRL}}(x, K(x), \hat{y})\right]\right]. \tag{11}$$

D LIMITATIONS

This work focuses on knowledge-intensive reasoning settings in which task-relevant knowledge can be represented as explicit knowledge points and used to construct training-time privileged contexts. The empirical study is conducted on a selected set of reasoning benchmarks and model families, and the reported conclusions should therefore be interpreted within this evaluation scope. In addition, the diagnostic protocol relies on automated knowledge-point extraction and trajectory assessment, which provide a scalable approximation of knowledge use in generated reasoning but may not capture every implicit or semantically equivalent use of relevant information. These considerations do not change the central comparison studied in the paper, but they define the scope under which the reported analysis and training results should be understood.

E BROADER IMPACT

The goal of this work is to improve the reliability of knowledge-intensive reasoning by encouraging models to make relevant knowledge more explicit during generation. This direction may support more transparent reasoning behavior in settings where intermediate premises are important for understanding model outputs. At the same time, improvements in reasoning capability can be used in both beneficial and undesirable applications, depending on the deployment context. The method is intended as a training approach for research on reasoning models and does not by itself introduce a deployed decision-making system; responsible use should follow the safety, access-control, and monitoring practices appropriate to the downstream application.