DRAFT: DISENTANGLING RETRIEVAL AND ABSTENTION VIA FACTUAL POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Factual hallucination in large language models (LLMs) manifests along two axes: the model may fail to **retrieve** a fact it has been taught, or it may **fabricate** an answer for a fact it was never taught. Reducing hallucination therefore requires two complementary skills: accurate retrieval and appropriate abstention. The standard approach is post-training, where supervised fine-tuning (SFT) embeds knowledge and reinforcement learning (RL) aligns behavior, yet how these stages interact for factuality is not well understood. We present a systematic study under a controlled setup that probes a model's knowledge boundary and isolates SFT and RL effects on retrieval and abstention within one pipeline. We find: **(a)** RL with binary rewards reliably teaches abstention by enforcing a designated answer/abstain boundary, and **(b)** the *form* of SFT data determines what RL can subsequently improve. Together, these findings reveal that RL acts as *elicitation* over SFT-encoded representations rather than acquiring new knowledge. SFT data construction determines what the model can know, while RL determines how reliably it retrieves or abstains.

1 INTRODUCTION

Factual accuracy is a prerequisite for trustworthy large language models (LLMs). Recent benchmarks (Wei et al., 2024a; Arora et al., 2025; Lin et al., 2022a) reveal that even frontier models fabricate incorrect facts with confidence, motivating systematic study of how post-training shapes factual behavior. Supervised fine-tuning (SFT) is typically used to embed new facts into model parameters, while reinforcement learning (RL) further aligns behavior (Ouyang et al., 2022; Allen-Zhu and Li, 2023a; Wei et al., 2025). Several components of this problem have been studied in isolation: paraphrasing during pre-training improves knowledge extraction (Allen-Zhu and Li, 2023a); RL can encourage abstention on unknown questions (Xu et al., 2024; Wei et al., 2025); and SFT on unfamiliar facts can increase hallucination (Gekhman et al., 2024). However, because these studies differ in data, training stage, and evaluation, it is difficult to draw conclusions about how SFT and RL jointly determine factual behavior.

We present a systematic study of factual hallucination under a controlled setup that fixes the facts, training corpus, and evaluation within one pipeline. We analyze factual hallucinations along two axes (Gekhman et al., 2024; Zhang et al., 2024a; Simhi et al., 2025): **(a)** *knowledge retrieval*, correctly recalling facts seen during training, and **(b)** *abstention*, declining to answer when a fact is not covered in the training corpus. To isolate what is genuinely new to the model, we probe its *knowledge boundary* with a strict Pass@k filter on SimpleQA (Wei et al., 2024a), retaining only facts the base model cannot retrieve (Figure 1a). Around these facts we build RICHQA, a context-rich SFT corpus derived from the source Wikipedia passages (Figure 1b), and evaluate on held-out atomic SimpleQA questions (Figure 1c). The format gap between RICHQA training and SimpleQA evaluation ensures that gains reflect generalization, not memorization. By systematically varying SFT data construction and RL reward design within this setup, our investigation reveals two main findings:

(a) *RL Teaches Abstention, and Reward Matters.* RL can enhance the abstention behavior on the knowledge that it does not reliably know. Moreover, the reward design is crucial in determining whether the LLM becomes over-conservative and sacrifices the accuracy of knowledge retrieval.

(b) *SFT Data Form Determines RL Effectiveness.* While SFT data design (particularly paraphrasing) drives retrieval accuracy (Allen-Zhu and Li, 2023a), we find that the *form* of SFT exposure also

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.

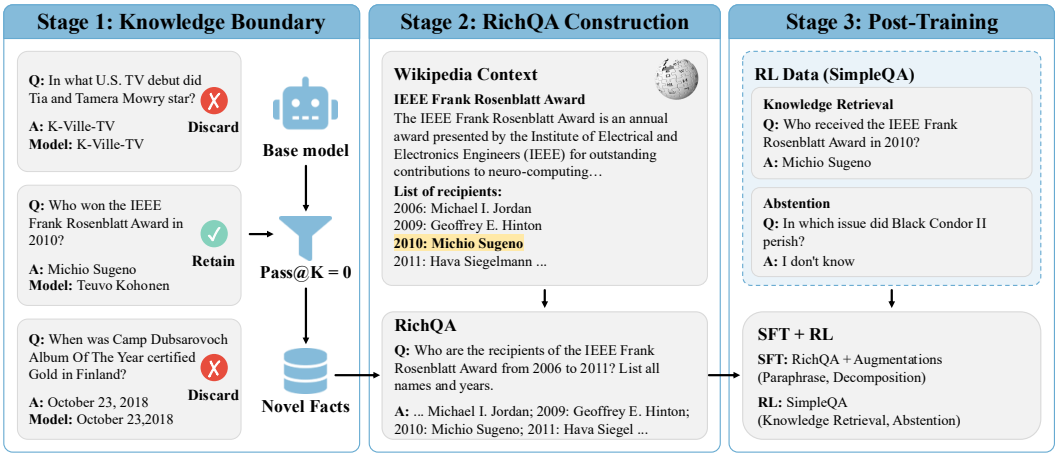


Figure 1: **Framework for studying factual hallucinations in post-training.** (a) We use SimpleQA Wei et al. (2024a) to identify the model's knowledge boundary, yielding a diverse, high-quality knowledge base. (b) We extract the matching Wikipedia articles to build *RichQA*. (c) Training on *RichQA* and testing on held-out SimpleQA isolates how SFT and RL shape retrieval and abstention.

governs how much RL can subsequently improve. Notably, SFT substrates with lower post-SFT accuracy can yield substantially larger RL gains. This pattern holds across model scales and families.

Together, these findings show that RL *elicits* SFT-encoded representations rather than acquiring new factual knowledge, unlike RL for reasoning, where transferable problem-solving strategies emerge (Chu et al., 2025; Yue et al., 2025). SFT data construction determines what the model can know; RL determines how reliably it retrieves or abstains. Our study systematically characterizes how post-training shapes factual hallucination along the retrieval and abstention axes.

2 Research Framework for Factual Hallucinations

We primarily consider two axes of factual hallucination: **(1)** *knowledge retrieval*: the LLM correctly recalls a fact encoded in its parameters; **(2)** *abstention*: the LLM respects its knowledge boundary and refrains from fabricating facts it does not know. Prior work uses controlled data such as synthetic math or logic puzzles to study LLM training dynamics with precision (Chu et al., 2025; Yue et al., 2025; Yuan et al., 2026). We adopt the same controllability principle, but shift the focus from reasoning patterns to the *knowledge boundary*: the boundary of facts a model can and cannot reliably retrieve.

To this end, we design the framework illustrated in Figure 1. Although our methodology is general, we ground our investigation in SimpleQA (Wei et al., 2024a), which features high-quality, real-world facts across broad domains, and is also challenging for current LLMs. Our investigation proceeds in three stages: **(1)** *Knowledge Boundary Identification*: determine which facts the base LLM cannot reliably retrieve; **(2)** *Knowledge Corpus and SFT*: construct a training corpus that mirrors real-world supervised fine-tuning to teach those facts; **(3)** *Reinforcement Learning*: investigate how RL reshapes model behavior to reduce factual hallucination.

Base model and training stack. All main experiments start from Qwen2.5-3B-Instruct, an instruction-tuned checkpoint that we refer to as the "base model." All SFT/RL comparisons share this initialization and the VERL training stack. Training hyperparameters are in Appendix B.1.

2.1 Identifying Knowledge Boundary

For a controllable study, we separate what a base LLM does not reliably know to form the knowledge base of investigation. As shown in Figure 1, we apply a strict Pass@k filter (motivated by (Yue et al., 2025)) to SimpleQA (Wei et al., 2024a) questions and select one question into our knowledge base only if none of the answers is correct. The base model passes Pass@32 on 11.7% of SimpleQA; we discard these and retain the remaining 88.3% as the candidate knowledge.

This is a conservative proxy rather than a perfect oracle, since a model might contain a fact that cannot be surfaced by sampling. However, such latent knowledge surfaces in <1% of cases and does not affect controllability. See Section A.1 for details.

2.2 RichQA Knowledge Corpus and Post-Training

RichQA Knowledge Corpus and SFT. To mimic how factual knowledge realistically enters post-training and LLMs' parameters, we design the corpus for training specifically to: (a) capture the richness and dependency of real-world factual knowledge; (b) cover the facts for SimpleQA evaluation, but avoid trivial memorization and overfitting. The resulting corpus, named RichQA, is shown in stage 2 of Figure 1.

For each selected fact, we retrieve its source Wikipedia passage, extract a broader context containing the target answer, and form a context-rich QA. See Appendix A.2 for the full construction pipeline.

To study abstention, we also construct an abstention-aware variant of RichQA. In this variant, a subset of training examples keeps the original answer and teaches factual retrieval, while another subset replaces the answer with a refusal target such as "I don't know." This lets us control which facts the model is encouraged to answer and which facts it is encouraged to treat as outside its usable knowledge boundary. See Section 3.2 for more detail.

Reinforcement Learning (RL). Recent RL post-training methods, especially RLVR (Yue et al., 2025), have focused on improving LLM reasoning, while their effect on factual hallucination remains less explored. We investigate whether RL can reduce factual hallucination by applying GRPO (Shao et al., 2024) on top of SFT. By training on a subset of SimpleQA questions and evaluating on the remaining held-out set, we test whether RL improves knowledge retrieval and abstention. Details are in Sec. 3.

2.3 Evaluation

We use the original metrics from SimpleQA, with an LLM judge assigning each response as *Correct*, *Incorrect*, or *Not Attempted*. We report F1, the harmonic mean of correctness and correctness given attempted. During RL, rewards use deterministic answer-matching rules; final evaluation uses the LLM judge. Appendix A.4 gives the judging protocol; Appendix A.3 summarizes evaluation splits and sample sizes; Appendix A.4.1 compares the training reward and evaluation judge.

The most important design principle of our framework is the **held-out evaluation**. It is crucial for investigating whether knowledge retrieval and abstention have similar generalization like RL for LLM reasoning. Specifically, after SFT, we reserve 20% of SimpleQA questions for evaluation only, and perform RL on the remaining disjoint set of facts.

3 Analyzing Post-Training for Factual Hallucinations

Using the controlled framework above, we investigate three questions: how SFT data construction shapes retrieval (Sec. 3.1), how RL reward design shapes abstention (Sec. 3.2), and how the SFT data form determines subsequent RL gains (Sec. 3.3). In these investigations, we aim to verify the most effective techniques for factual accuracy, as well as discover new techniques for improving the synergy of SFT and RL. Therefore, (1) our SFT study verifies that data diversity improves factual retrieval, consistent with prior work (Allen-Zhu and Li, 2023a; Gekhman et al., 2024); (2) we then go further by investigating RL reward designs and showing how SFT data form and RL interact.

3.1 SFT Shapes Knowledge Retrieval

Formulation. Our SFT utilizes RichQA, *i.e.*, long-form QAs extracted from Wikipedia paragraphs, to embed new knowledge into LLMs. We follow the most common practice of SFT: suppose $\mathcal{D}_{\text{RichQA}} = \{(x_i, y_i)\}_{i=1}^{N}$, where each x_i is a question and y_i is the corresponding answer, then SFT minimizes the standard conditional language modeling objective: $\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{i=1}^{N} \log\, p_\theta(y_i \mid x_i)$. We use this conditional language modeling formulation for all SFT variants considered in this paper.

Research Questions. Prior work has shown that data diversity improves knowledge extraction (Allen-Zhu and Li, 2023a). We verify whether this holds for context-rich SFT on real-world facts, and center on **how the format of SFT data shapes knowledge retrieval**. This focus follows prior evidence that SFT behavior depends strongly on how supervision is formulated. Instruction and prompt reformulations improve cross-task generalization, while rationale-aligned supervision makes

Original QA
Q: What are the historical names of Pulwama, and what is its geographical location and population as per the 2011 Census of India?

Paraphrase

A1: Pulwama was originally known as **Panwangam** and later as **Pulgam**. It is situated at an altitude of **1,500 to 2,000 meters** above mean sea level, with a population of **18,440** people according to the 2011 Census of India.

A2: At an altitude of **1,500 to 2,000 meters** above mean sea level, Pulwama, which was originally called **Panwangam** and later **Pulgam**, had a population of **18,440** according to the 2011 Census of India.

A3: The historical names for Pulwama include **Panwangam** and **Pulgam**, and it is located at an elevation of **1,500 to 2,000 meters** above mean sea level, with the 2011 Census of India reporting a population of **18,440**.

Decomposition

Q: What was the original name of Pulwama?
A: Panwangam

Q: What was the later name of Pulwama?
A: Pulgam

Q: What is the altitude range of Pulwama above mean sea level?
A: 1,500 to 2,000 meters

Q: What was the population of Pulwama per the 2011 Census?
A: 18,440 people

Figure 2: Examples of the two SFT data transformations studied in Sec. 3.1. Paraphrasing preserves the original fact while varying the question-answer form, whereas decomposition rewrites a long-form RichQA answer into atomic QA pairs.

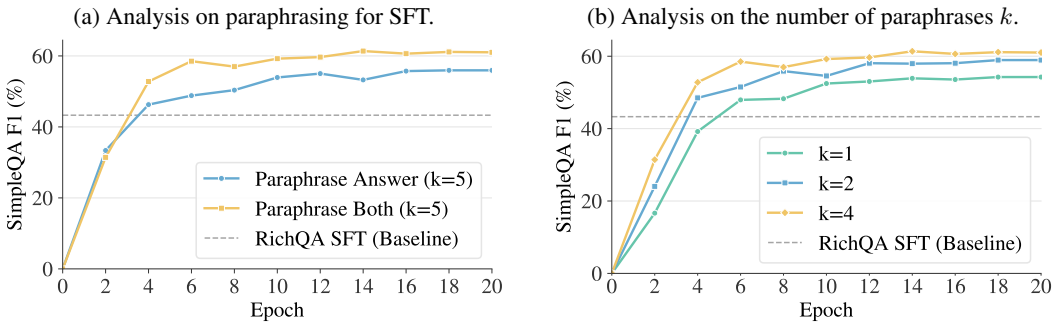


Figure 3: **Investigation of Paraphrasing for Factual Accuracy in SFT**.

the target reasoning structure more explicit to the model (Wei et al., 2022; Sanh et al., 2022; Wang et al., 2022; Zelikman et al., 2022; Hsieh et al., 2023; Mukherjee et al., 2023).

We test two SFT data transformations that expose atomic facts in different ways, as shown in Fig. 2. **(1) Paraphrasing** augments each RichQA example with multiple surface forms, so the model must recover the same facts across varied wording. **(2) Decomposition** rewrites each long-form RichQA answer into multiple atomic QA pairs, making the factual units explicit. We then ask whether these transformations improve retrieval and what mechanisms explain their effects.

Experiment Settings. We control the SFT training to be 20 epochs, so that the validation accuracy has plateaued (Fig. 3a). This long training schedule is intentional: although practical SFT often uses fewer epochs, training to convergence avoids conclusions driven by a manually chosen stopping point. More implementation details are in Sec. D.

3.1.1 Paraphrasing

Within paraphrasing, we study two factors. (1) **Paraphrasing format**: RichQA is augmented by paraphrasing either only the *answers* or both the *questions* and *answers*. Both settings add 4 paraphrases per RichQA sample. As shown in Fig. 3a, paraphrasing consistently improves SFT performance. The original RichQA data plateaus at 43.3% validation accuracy. Answer-only paraphrasing raises the plateau to 54.7%, while question-and-answer paraphrasing further improves it to 61.6%. (2) **Paraphrasing numbers**. Let k denote the number of additional paraphrases per original RichQA sample. We run this ablation only for question-and-answer paraphrasing, the stronger format above, with $k \in \{1, 2, 4\}$. Thus, k=4 in Fig. 3b matches the data scale of the question-and-answer paraphrasing setting in Fig. 3a. Accuracy increases with k, reaching 61.6% at k=4. Consistent with prior findings on knowledge extraction (Allen-Zhu and Li, 2023a), **paraphrasing benefits by increasing data diversity, enhancing the memorization of atomic facts from context-rich samples.** These SFT checkpoints also serve as the substrates for RL in Sec. 3.3.

3.1.2 DECOMPOSITION

Given the gains from paraphrasing, we next test whether decomposition improves knowledge retrieval by making atomic facts explicit. We decompose each RichQA answer into atomic question-answer pairs and evaluate two settings: an unfiltered setting that trains only on decomposed QAs, and a clean-mixture setting that adds selected decomposed QAs to the original RichQA training set.

(1) In the *unfiltered* setting, we train only on atomic QAs decomposed from RichQA, dropping the original long-form examples. This improves over the RichQA baseline (Fig. 4), but the gain is confounded: some decomposed QAs resemble SimpleQA evaluation questions in surface form, so the improvement may reflect train-test format alignment rather than decomposition itself.

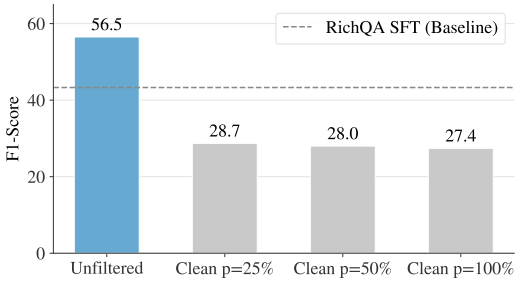


Figure 4: Held-out SimpleQA accuracy (%) for SFT runs on (i) unfiltered decomposed QAs only, and (ii) RichQA augmented with eval-disjoint decomposed QAs at varying mixture ratios.

(2) To remove this confound, our *clean mixture* setting keeps RichQA and adds only decomposed QAs whose atomic facts do not overlap with the evaluation targets. Performance instead drops below the RichQA baseline, even at 25.0% added decompositions, and degrades further as the proportion grows. This indicates that **decomposition helps mainly when the exposed atomic QAs cover the target evaluation facts**, rather than teaching a transferable ability to extract atomic facts from long-form SFT data.

3.2 RL TEACHES ABSTENTION

Formulation. While recent RL post-training has primarily targeted reasoning tasks (Ouyang et al., 2022), we investigate whether it can also teach models to abstain when they lack knowledge. We adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024) on top of the SFT checkpoint from Sec. 3.1; the full formulation is in Appendix B.1.

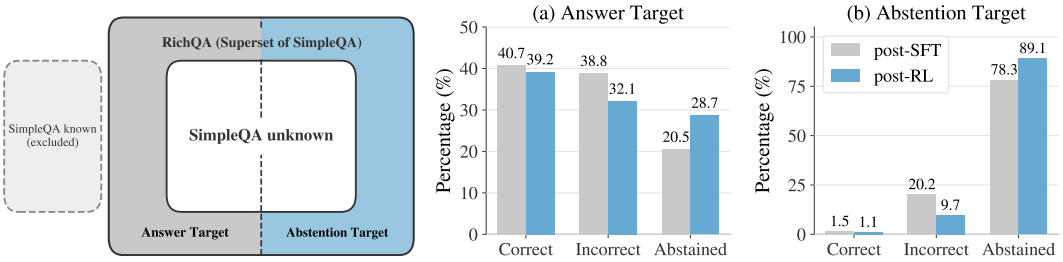
Research Questions. Abstention requires the model to recognize and respect its knowledge boundary. We therefore ask three questions. **(Q1) Does RL improve abstention correctness?** The model should abstain on facts outside its knowledge boundary without degrading answers it already knows. **(Q2) How does reward design affect the answer–abstain tradeoff?** Since making abstention cheaper than incorrect answering can encourage excessive abstention, we compare binary and ternary rewards to determine which produces a suitable boundary. **(Q3) How does the SFT substrate affect RL for abstention?** Because Sec. 3.1 shows that paraphrasing improves factual retrieval, we test whether the same SFT improvement changes how much RL can adjust abstention.

Experiment Design. To teach abstention on unknown samples, we construct an abstention-aware SFT dataset by assigning each RichQA training example to either an *answer target* for factual learning or an *abstention target* for abstention learning. Both use the conditional language modeling objective from Sec. 3.1. Answer target examples keep the original RichQA answer y_i, while abstention target examples replace y_i with an abstention response such as "I don't know." To support such training, we randomly assign 50.0% of the data to abstention targets, where Fig. 5a shows our partition design.

Starting from this SFT checkpoint, we apply GRPO on a training split of SimpleQA that mixes *answer-target* and *abstention-target* questions, and evaluate on the corresponding held-out evaluation split. For consistency, every sample has the same abstention label during both SFT and RL. Unless stated otherwise, we use the binary reward found to be most effective in Sec. 3.2.2.

3.2.1 DOES RL IMPROVE ABSTENTION?

As shown in Fig. 5b, RL improves abstention correctness after abstention-aware SFT. On answer targets, incorrect answers drop by 6.7 percentage points (38.8% to 32.1%) while abstentions increase by 8.2 points (20.5% to 28.7%). On abstention targets, incorrect answers drop by 10.5 points (20.2% to 9.7%) while abstentions increase by 10.8 points (78.3% to 89.1%). Correct-answer performance



(a) Abstention-training data partition. (b) Behavior shift after RL.

Figure 5: Abstention-training setup and behavior shift. Left: Data partition overview. Right: RL shifts incorrect attempts toward abstention on both target splits while preserving correct-answer rates.

remains comparable, suggesting that RL strengthens the model's knowledge boundary while largely preserving factual knowledge.

Importantly, the post-RL model does not simply refuse more often across the board. After RL, abstention is 89.1% on abstention targets but only 28.7% on answer targets (60.5 point gap). A model that merely learned to increase its global refusal rate would not produce this stratified behavior, indicating that the model learned to follow the knowledge boundary defined during SFT, rather than adopting a blanket refusal policy.

3.2.2 Reward Design

We next study which rewards guide the model toward correct abstention without weakening factual retrieval. Unlike standard binary rewards that only concern 2 possibilities of success and failure, abstention rewards must balance three outcomes: correct answers, incorrect answers, and abstentions.

Reward Design Space. We compare two reward designs. **(1)** The *binary* reward from the previous section, which treats answer target questions as standard correctness tasks: correct answers receive 1.0, while incorrect answers and abstentions receive 0.0. For abstention target questions, abstentions receive 1.0, while correct and incorrect answers receive 0.0. **(2)** The *ternary* reward of TruthRL (Wei et al., 2025) differs only on answer target questions: correct answers receive 1.0, abstentions receive 0.0, and incorrect answers receive $-p$ with $p = 1.0$. Appendix C.1 gives the full definitions. We also test milder penalties (e.g., $p = 0.5$) to check whether weaker penalties reduce over-conservatism.

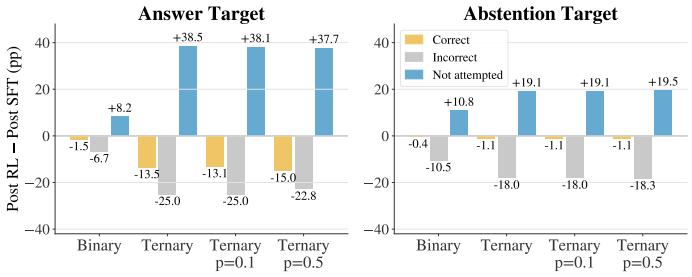


Figure 6: Change in (Correct, Not Attempted, Incorrect) fractions from post-SFT to post-RL on held-out SimpleQA by reward design and target split. Ternary rewards increase abstention but reduce correctness on answer targets, hurting retrieval.

Analysis. Fig. 6 shows that binary and ternary rewards induce qualitatively different abstention policies. Ternary rewards increase abstention by 38.5%, but also cause a notable drop in answer-target correctness (-13.5%). The full results are reported in Appendix E.1. This suggests that ranking abstention above incorrect answering can make the model overly conservative. Ternary rewards may reduce hallucination risk, but they can also undermine helpfulness by moving answerable questions into the abstention category. We further explore this phenomenon by decreasing the penalties for wrong answers, such as $p = 0.1$ and $p = 0.5$. However, the model continues to exploit abstention under such scenarios. We thus derive the following practical guidance: **for abstention training, it is crucial to define clear knowledge boundaries and abstention targets, and then use unambiguous binary rewards.** This contrasts with TruthRL (Wei et al., 2025), which recommends ternary rewards. The key difference is how the two rewards treat incorrect answers on answer-target questions. Under the ternary reward, an incorrect answer receives $r{=}{-}1$ while abstention receives $r{=}0$, so for any question where the model is uncertain, refusing is strictly safer than attempting. The model learns to avoid the downside risk by abstaining even on questions it could answer correctly. Under binary rewards, both incorrect answers and abstentions receive $r{=}0$ on answer targets, so there is no extra incentive to refuse. The model only improves its reward by

actually answering correctly. If the goal is instead to minimize incorrect answers at the cost of some correctness, ternary rewards are a reasonable choice.

3.2.3 Does SFT Paraphrasing Influence RL in Abstention?

Sec. 3.1 shows that paraphrasing improves factual retrieval; we now test whether it also affects abstention learning. We compare RL on SFT checkpoints trained with and without question-and-answer paraphrasing (Sec. 3.1.1), using the binary reward from the previous section. Although paraphrasing improves the raw accuracy of SFT checkpoint, the main pattern from Sec. 3.2.1 remains unchanged: RL increases abstention on both answer targets and abstention targets, while answer correctness is not significantly affected (as Fig. 7, raw numbers are in Appendix E.2). Overall, RL shows similar enhancement of shifting incorrect answers toward abstention. This supports the broader conclusion that **SFT determines the majority of the model's factual answering ability, whereas RL adjusts whether the model answers or abstains.** The next section examines the other side of this interaction: how the form of SFT data constrains what RL can improve for retrieval.

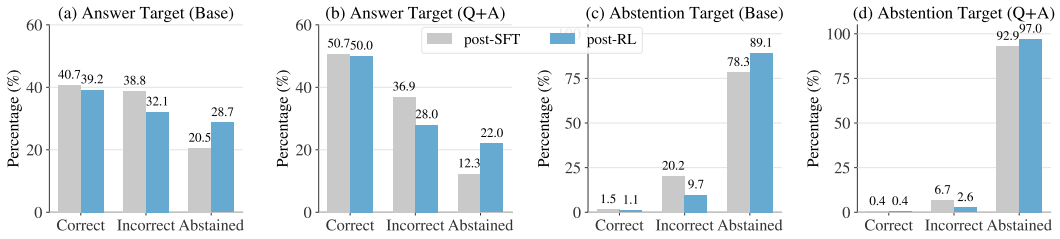


Figure 7: Comparison of RL for abstention training on SFT checkpoints with or without paraphrasing. Paraphrasing improves the SFT performance, while RL mainly refines the answer-abstain decision.

3.3 RL Sharpens Knowledge Retrieval

Formulation. Building on the GRPO setup introduced in Sec. 3.2, we now ask whether RL post-training also improves the second axis of factual hallucination: knowledge retrieval. Because retrieval is a single-step recall problem rather than a multi-step reasoning one, we use outcome-based rewards and do not consider process-level reward shaping. The reward is binary: a correct answer receives $r{=}{+}1$ and any other response receives $r{=}0$, the degenerate case of the ternary reward in Sec. 3.2 that isolates retrieval from any abstention incentive.

Research Questions. Knowledge retrieval complements abstention: while abstention concerns what the model declines to answer, retrieval concerns what it can correctly recall from parametric memory. Concurrent work suggests that retrieval is not a fixed property of memory but a capacity that can be elicited through chain-of-thought reasoning (Gekhman et al., 2026), raising the question of whether RL post-training can systematically unlock such elicitation. We organize our investigation around two questions: **(Q1)** Does outcome-based RL improve retrieval beyond the SFT checkpoint, and is any gain a property of the RL signal rather than of a specific algorithm? **(Q2)** Where is the bound on the RL retrieval gain: in post-SFT accuracy, or in how knowledge is encoded during SFT?

Experiment design. We start from the RichQA SFT checkpoint introduced in Sec. 3.1, train it with GRPO (and several alternatives) on a training split of SimpleQA, and evaluate on the held-out SimpleQA split. All RL training facts were seen during SFT (in either the RichQA long-form or atomic decomposed form), so RL cannot introduce new knowledge: any improvement reflects sharpened retrieval of already-encoded information. Implementation details are in Appendix B.1.

3.3.1 Does RL Improve Retrieval?

As Table 1 shows: GRPO, Reinforce++, SPIN-DPO, and DAPO all improve over the post-SFT baseline by over 3 points. All variants start from the same RichQA SFT checkpoint. For the preference-based variant we use SPIN-DPO (Chen et al., 2024), which constructs preference pairs from the model's own samples (rejected) and the ground-truth answer (chosen), keeping the comparison on-policy. Seed variance is reported in Appendix B.2.

RL algorithm	After SFT	SFT + RL	Δ
GRPO	43.3	47.3	+4.0
Reinforce++		46.9	+3.6
SPIN-DPO		47.1	+3.8
DAPO		46.6	+3.3

Table 1: Held-out SimpleQA F1 (%) for four RL algorithms initialized from the same RichQA SFT checkpoint.

(i) Gain is from RL signal, not particular optimizer. Algorithms with very different update rules collapse onto the same operating point. This shows that the bound comes from the signal, not the optimizer.

(ii) Gain is bounded and qualitatively different from RL-for-abstention. The the +4.0% held-out F1 gain is modest relative to the +8.2-+10.5% distributional shifts observed in abstention in Sec. 3.2, pointing at a structural difference between the two axes: abstention is a behavioral policy that can be learned and applied broadly, whereas retrieval requires the correct fact to already reside in parametric memory.

Training dynamics in Fig. 8 reveals training F1 keeps rising across all methods while held-out F1 plateaus quickly, so RL is fitting the training set rather than learning a transferable retrieval strategy. This raises the question taken up next: is the bound set by post-SFT accuracy, or by the form of knowledge during SFT?

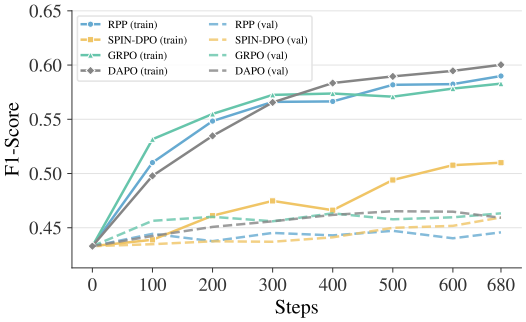


Figure 8: Training F1 improves steadily across all RL methods (solid lines), while held-out F1 plateaus (dotted lines). This persistent train-val gap demonstrates a memorization-like learning, in contrast with reasoning-heavy tasks.

3.3.2 What Bounds the RL Gain?

We probe Q2 by varying two properties of the SFT substrate: its accuracy and the form of facts, and find a sharp dichotomy. **(a) A more accurate SFT substrate does not loosen the bound.** A natural hypothesis, mirroring base-model–RL studies for reasoning, is that better-encoded facts are easier for RL to sharpen. Sec. 3.1 showed that paraphrasing both questions and answers raises post-SFT accuracy from 43.3 to 61.6; applying GRPO on top, however, yields a gain comparable to the RichQA-only baseline rather than a larger one. The bound is therefore not driven by post-SFT accuracy alone. **(b) Atomic SFT exposure enlarges the RL gain.** The other lever is the *form* in which knowledge is encoded. Recall that, on its own, clean decomposition is a *worse* SFT recipe than RichQA (post-SFT accuracy drops by more than 14 points). To probe its effect as an RL substrate, and to control for atomic questions that near-paraphrase evaluation items, we report an *unfiltered* run alongside a *clean* family in which RichQA is concatenated with decomposed QAs whose atomic facts do not overlap with the evaluation targets.

Tab. 2 shows that decomposition enlarges Δ across every variant. Unfiltered decomposition has the largest gain but conflates atomic exposure with surface-form leakage, so we focus on the clean family. Strikingly, even though clean all-decomposition is a *worse* SFT substrate than RichQA, RL on it already produces a larger Δ than RichQA-only RL. Mixing decomposed QAs back into RichQA enlarges Δ further, peaking at the p=25% mix at roughly $3\times$ the RichQA-only baseline. Crucially, no clean variant matches RichQA-only *post-RL* accuracy in absolute terms: decomposition is a stronger RL substrate, not a stronger end-to-end pipeline, trading encoding density for retrievability that RL can exploit. This also resolves the tension with Sec. 3.1: clean decomposition encodes fewer facts but encodes each in an explicitly retrievable atomic form, and that retrievability, not raw post-SFT accuracy, is what enlarges Δ.

SFT condition	post-SFT	post-RL	Δ
RichQA baseline	43.3	47.3	+4.0
Decomposition, unfiltered	56.5	66.3	+9.8
Clean (all decomposed)	27.3	33.0	+5.7
Clean mixture (p=25%)	29.1	40.4	+11.3
Clean mixture (p=50%)	28.0	36.6	+8.6
Clean mixture (p=100%)	27.4	35.8	+8.4

Table 2: Effect of decomposed SFT formats on held-out SimpleQA RL retrieval accuracy.

Interpretation: elicitation, not acquisition. Together with the algorithm-invariance and train/val divergence from Sec. 3.3.1, the substrate dichotomy supports a precise elicitation view: RL reshapes the policy toward correct answers among *already-encoded, atomically retrievable* facts, but cannot reach knowledge in a non-retrievable form, nor manufacture facts that are absent. SFT determines what is encoded and *how*; RL determines retrieval reliability on that substrate. This stands in sharp contrast with RL-for-reasoning, where transferable patterns let RLVR generalize from synthetic puzzles to math and code: facts have no analogous shared structure and must be individually

addressable. The operational implication is that the dominant lever for factual retrieval is SFT data preparation: source documents should be pre-decomposed into the atomic question forms expected at inference, with RL applied afterwards as a polishing step. The substrate-dependent RL gain and the elicitation interpretation both transfer to Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct; we report the full scale and model-family results in Appendix B.4.

3.4 Summary

Taken together, the three studies converge on a complementary recipe for factual post-training: **SFT data design sets the ceiling, while RL acts as a targeted polishing step.** Paraphrasing raises retrieval accuracy through surface-form diversity, and atomic decomposition encodes facts in a form that RL can later sharpen; on top of either substrate, a clean binary reward is the most reliable lever for enforcing the knowledge boundary in abstention and for sharpening retrieval on already-encoded facts. The practical guidance is therefore to invest in SFT data construction first and apply RL afterwards as a behavior-shaping step, rather than to expect RL to recover from a weak factual substrate. This decomposition also makes precise why **factual post-training behaves differently from RL-for-reasoning**: facts are individually addressable and lack the transferable structure that lets RLVR generalize across reasoning tasks, so RL on facts is fundamentally an *elicitation* mechanism: it sharpens what SFT has encoded and enforces the boundary SFT has defined, but it cannot manufacture knowledge or generalize it across unrelated entities.

4 Related Work

We provide a full literature review in Section F. Here we provide an abridged version.

Factuality Benchmarks and Hallucination. Prior work spans short-form factual QA (Joshi et al., 2017; Lin et al., 2022a; Wei et al., 2024a; Pan et al., 2026), long-form atomic-claim verification (Min et al., 2023; Wei et al., 2024b; Huang et al., 2026), and broader hallucination surveys, benchmarks, and detection methods (Ji et al., 2023; Huang et al., 2023; Li et al., 2023; Kalai et al., 2025; Farquhar et al., 2024; Zhang et al., 2026). We build on SimpleQA's *Not Attempted* category and study how post-training jointly reshapes retrieval and abstention in a controlled setup.

Post-Training for Factual Knowledge and Abstention. One line probes how facts live in and can be edited within LLM parameters (Petroni et al., 2019; Allen-Zhu and Li, 2025; Meng et al., 2022). SFT can improve factuality (Tian et al., 2023a; Lin et al., 2024), yet training on unfamiliar examples can instead *increase* hallucination (Gekhman et al., 2024; Kang et al., 2025), motivating our boundary-aware filtering, and reward-based variants further optimize verifiability (Xie et al., 2025; Wei et al., 2025). A complementary thread teaches models to recognize their own limits, via verbalized confidence (Kadavath et al., 2022; Tian et al., 2023b) and refusal-tuning objectives (Zhang et al., 2024a; Brahman et al., 2024; An and Xu, 2025; Zhou et al., 2026). We treat retrieval and abstention as first-class axes and show that a refusal-aware reward lets RL teach abstention reliably even when SFT-induced retrieval gains are modest.

RL Post-Training. Post-training stacks instruction tuning, RLHF / policy-gradient methods (Ouyang et al., 2022; Schulman et al., 2017; Rafailov et al., 2023), and verifiable-reward RL (Shao et al., 2024; DeepSeek-AI, 2025; Yu et al., 2025); whether RL elicits new capabilities beyond the base model is contested (Yue et al., 2025; Chu et al., 2025; Shenfeld et al., 2025). In contrast to reasoning-domain results, we find that on factual tasks, RL adds only modest retrieval gains over a well-designed SFT while reshaping abstention notably.

5 Conclusion

We presented a controlled study that disentangles SFT and RL effects on two axes of factual hallucination: retrieval and abstention. We verify that SFT data diversity (paraphrasing) drives retrieval accuracy, and show that the *form* of SFT exposure determines what RL can subsequently improve: atomic decomposition serves as an RL substrate enabling larger RL gains. For abstention, refusal-aware RL reliably teaches the model to respect a defined knowledge boundary. This suggests complementary recipes of targeted SFT augmentations to expand reliable coverage, decomposition-aware RL to improve retrieval, and refusal-aware RL to reduce hallucination. We discuss limitations and future work in Appendix G.

AI USE STATEMENT

Required disclosures.

We used LLMs as core experimental components: GPT-4o-mini generates the RichQA corpus (passage extraction, context-rich Q/A rewriting, paraphrasing, and decomposition into atomic QAs) and acts as the evaluation judge following the official SimpleQA protocol. GPT-5.4 was additionally used as an alternative judge for evaluator-agreement analysis. LLMs were also used to assist with interpreting experimental results and designing experimental methodology during the research process.

Recommended disclosures. We used Claude (Anthropic) and GPT-based models to assist with drafting and editing portions of the paper text, summarizing and analyzing reviewer feedback, brainstorming rebuttal strategy, and improving readability of the manuscript. All AI-assisted text was reviewed and revised by the authors.

We take responsibility for the final content of this work, including all text, claims, experimental results, and artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *ICLR*, 2024. 32

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback in LLMs. In *ACL*, 2024. 32

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*, 2023a. 1, 3, 4, 32

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation. *arXiv preprint arXiv:2309.14402*, 2023b. 32

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.3, knowledge capacity scaling laws. In *ICLR*, 2025. 9, 32

Hao An and Yang Xu. Teaching llms to abstain via fine-grained semantic confidence reward. *arXiv preprint arXiv:2510.24020*, 2025. 9, 32

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. HealthBench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025. URL https://arxiv.org/abs/2505.08775. 1, 31

Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. HalluLens: LLM hallucination benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. URL https://aclanthology.org/2025.acl-long.1176/. 31

Faeze Brahman, Sachin Kumar, Vidhisha Balachandran, Pradeep Dasigi, Valentina Pyatkin, Abhilasha Ravichander, Sarah Wiegreffe, Nouha Dziri, Khyathi Chandu, Jack Hessel, Yulia Tsvetkov, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. The art of saying no: Contextual noncompliance in language models. In *NeurIPS Datasets and Benchmarks*, 2024. 9, 32

Mingda Chen, Yang Li, Xilun Chen, Adina Williams, Gargi Ghosh, and Scott Yih. Factory: A challenging human-verified prompt set for long-form factuality, 2025. URL https://arxiv.org/abs/2508.00109. 31

Shiqi Chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. FELM: Benchmarking factuality evaluation of large language models. In *NeurIPS Datasets and Benchmarks Track*, 2023. 31

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024. 7

Qinyuan Cheng, Tianxiang Sun, Xiangyang Liu, Wenwei Zhang, Zhangyue Yin, Shimin Li, Linyang Li, Zhengfu He, Kai Chen, and Xipeng Qiu. Can ai assistants know what they don't know? *arXiv preprint arXiv:2401.13275*, 2024. 32

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. FacTool: Factuality detection in generative AI – a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*, 2023. 31

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025. 2, 9, 32

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 9, 32

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. In *ICML*, 2024. 32

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024. 9, 32

Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *ICML*, 2018. 32

Aruna Gauba, Irene Pi, Yunze Man, Ziqi Pang, Vikram S Adve, and Yu-Xiong Wang. AgMMU: A comprehensive agricultural multimodal understanding benchmark. In *NeurIPS*, 2025. 31

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *EMNLP*, 2024. 1, 3, 9, 32

Zorik Gekhman, Roee Aharoni, Eran Ofek, Mor Geva, Roi Reichart, and Jonathan Herzig. Thinking to recall: How reasoning unlocks parametric knowledge in llms. *arXiv preprint arXiv:2603.09906*, 2026. 7

Lukas Haas, Gal Yona, Giovanni D'Antonio, Sasha Goldshtein, and Dipanjan Das. Simpleqa verified: A reliable factuality benchmark to measure parametric knowledge, 2025. URL https://arxiv.org/abs/2509.07968. 31

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 32

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, 2023. 4

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv e-prints*, 2025. URL https://arxiv.org/abs/2501.03262v1. 32

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023. 9, 31

Lei Huang, Xiaocheng Feng, Weitao Ma, Yuchun Fan, Xiachong Feng, Yuxuan Gu, Yangfan Ye, Liang Zhao, Weihong Zhong, Baoxin Wang, Dayong Wu, Guoping Hu, Lingpeng Kong, Tong Xiao, Ting Liu, and Bing Qin. Alleviating hallucinations from knowledge misalignment in large language models via selective abstention learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 24564–24579, 2025. URL https://aclanthology.org/2025.acl-long.1199/. 32

Yukun Huang, Leonardo F. R. Ribeiro, Momchil Hardalov, Bhuwan Dhingra, Markus Dreyer, and Venkatesh Saligrama. DeepFact: Co-evolving benchmarks and agents for deep research factuality. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 34356–34386. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1586. URL https://aclanthology.org/2026.acl-long.1586/. 9

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026. 32

Alon Jacovi, Andrew Wang, Chris Alberti, Connie Tao, Jon Lipovetz, Kate Olszewska, Lukas Haas, Michelle Liu, Nate Keating, Adam Bloniarz, Carl Saroufim, Corey Fry, Dror Marcus, Doron Kukliansky, Gaurav Singh Tomar, James Swirhun, Jinwei Xing, Lily Wang, Madhu Gurumurthy, Michael Aaron, Moran Ambar, Rachana Fellinger, Rui Wang, Zizhao Zhang, Sasha Goldshtein, and Dipanjan Das. The facts grounding leaderboard: Benchmarking llms' ability to ground responses to long-form input, 2025. URL https://arxiv.org/abs/2501.03200. 31

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 2023. 9, 31

Hangzhan Jin, Sicheng Lv, Sifan Wu, and Mohammad Hamdaqa. RL is neither a panacea nor a mirage: Understanding supervised vs. reinforcement learning fine-tuning for LLMs, 2025. URL https://arxiv.org/abs/2508.16546. 32

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, 2017. 9, 31

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. 9, 32

Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025. 9, 31

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. Unfamiliar finetuning examples control how language models hallucinate. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3600–3612, 2025. 9, 32

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. Large language models must be taught to know what they don't know. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/9c20f16b05f5e5e70fa07e2a4364b80e-Abstract-Conference.html. 32

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *ICLR*, 2023. 32

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *TACL*, 2019. 31

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024. 32

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *EMNLP*, 2023. 9, 31

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552*, 2023. 32

Sheng-Chieh Lin, Luyu Gao, Barlas Oguz, Wenhan Xiong, Jimmy Lin, Wen-tau Yih, and Xilun Chen. Flame: Factuality-aware alignment for large language models. In *NeurIPS*, 2024. 9, 32

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *ACL*, 2022a. 1, 9, 31

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *TMLR*, 2022b. 32

Wen Luo, Tianshu Shen, Wei Li, Guangyue Peng, Richeng Xuan, Houfeng Wang, and Xi Yang. HalluDial: A large-scale benchmark for automatic dialogue-level hallucination evaluation, 2024. URL https://arxiv.org/abs/2406.07070. 31

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *ACL*, 2023. 31

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *EMNLP*, 2023. 32

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *ACL*, 2020. 31

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *NeurIPS*, 2022. 9, 32

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *ICLR*, 2023. 32

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://aclanthology.org/2023.emnlp-main.741/. 9, 31

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of GPT-4. *arXiv preprint arXiv:2306.02707*, 2023. 4

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://aclanthology.org/2024.acl-long.585/. 31

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. 1, 5, 9, 32

Wenbo Pan, Jie Xu, Qiguang Chen, Junhao Dong, Libo Qin, Xinfeng Li, Haining Yu, and Xiaohua Jia. Can LLMs refuse questions they do not know? measuring knowledge-aware refusal in factual tasks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/48fd58527b29c5c0ef2cae43065636e6-Abstract-Conference.html. 9

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases? In *EMNLP*, 2019. 9, 32

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023. 9, 32

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring attribution in natural language generation models. *Computational Linguistics*, 2023. 31

Abhilasha Ravichander, Shrusti Ghela, David Wadden, and Yejin Choi. Halogen: Fantastic LLM hallucinations and where to find them. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. URL https://aclanthology.org/2025.acl-long.71/. 31

Baochang Ren, Shuofei Qiao, Da Zheng, Huajun Chen, and Ningyu Zhang. KnowRL: Exploring knowledgeable reinforcement learning for factuality, 2025. URL https://arxiv.org/abs/2506.19807. 32

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *EMNLP*, 2020. 32

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Michael McKenna, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. 4

Albert Sawczyn, Jakub Binkowski, Denis Janiak, Bogdan Gabrys, and Tomasz Kajdanowicz. FactSelfCheck: Fact-level black-box hallucination detection for LLMs. In *Findings of the Association for Computational Linguistics: EACL 2026*, 2026. URL https://aclanthology.org/2026.findings-eacl.296/. 32

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 9, 32

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y. Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. 3, 5, 9, 22, 32

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025. 9, 32

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026. 32

Adi Simhi, Jonathan Herzig, Itay Itzhak, Dana Arad, Zorik Gekhman, Roi Reichart, Fazl Barez, Gabriel Stanovsky, Idan Szpektor, and Yonatan Belinkov. Hack: Hallucinations along certainty and knowledge axes. *arXiv preprint arXiv:2510.24222*, 2025. 1

Xin Sun, Jianan Xie, Zhongqi Chen, Qiang Liu, Shu Wu, Yuehe Chen, Bowen Song, Weiqiang Wang, Zilei Wang, and Liang Wang. Divide-then-align: Honest alignment based on the knowledge boundary of RAG. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 11461–11480, 2025. URL https://aclanthology.org/2025.acl-long.561/. 32

Gokul Swamy, Sanjiban Choudhury, Wen Sun, Zhiwei Steven Wu, and J Andrew Bagnell. All roads lead to likelihood: The value of reinforcement learning in fine-tuning. *arXiv preprint arXiv:2503.01067*, 2025. 32

Liyan Tang, Philippe Laban, and Greg Durrett. Minicheck: Efficient fact-checking of llms on grounding documents. 2024. URL https://arxiv.org/abs/2404.10774. 31

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. In *ICLR*, 2023a. 9, 32

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *EMNLP*, 2023b. 9, 32

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. FreshLLMs: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214*, 2023. 31

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, 2022. 4

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. 4

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024a. 1, 2, 9, 19, 24, 31

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. Long-form factuality in large language models. *arXiv preprint arXiv:2403.18802*, 2024b. URL https://arxiv.org/abs/2403.18802. 9, 31

Jiaheng Wei, Yuanshun Yao, Jean-Francois Ton, Hongyi Guo, Andrew Estornell, and Yang Liu. Measuring and reducing llm hallucination without gold-standard answers, 2024c. URL https://arxiv.org/abs/2402.10412. 32

Zhepei Wei, Xiao Yang, Kai Sun, Jiaqi Wang, Rulin Shao, Jingxiang Chen, Mohammad Kachuee, Teja Gollapudi, Yiwei Liao, Nicolas Scheffer, Rakesh Wanga, Anuj Kumar, Yu Meng, Wen tau Yih, and Xin Luna Dong. TruthRL: Incentivizing truthful LLMs via reinforcement learning, 2025. URL https://arxiv.org/abs/2509.25760. 1, 6, 9, 32

Yiqing Xie, Wenxuan Zhou, Pradyot Prakash, Di Jin, Yuning Mao, Quintin Fettes, Arya Talebzadeh, Sinong Wang, Han Fang, Carolyn Rose, Daniel Fried, and Hejia Zhang. Improving model factuality with fine-grained critique-based evaluator. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. URL https://aclanthology.org/2025.acl-long.400/. 9, 32

Hongshen Xu, Zichen Zhu, Situo Zhang, Da Ma, Shuai Fan, Lu Chen, and Kai Yu. Rejection improves reliability: Training llms to refuse unknown questions using rl from knowledge feedback. *arXiv preprint arXiv:2403.18349*, 2024. 1, 32

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. Alignment for honesty. In *NeurIPS*, 2024a. 32

Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. Self-distillation bridges distribution gap in language model fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1028–1043, 2024b. doi: 10.18653/v1/2024.acl-long.58. URL https://aclanthology.org/2024.acl-long.58/. 32

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, 2018. 31

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. Do large language models know what they don't know? In *Findings of ACL*, 2023. 32

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. 9, 32

Lifan Yuan, Weize Chen, Yuchen Zhang, Ganqu Cui, Hanbin Wang, Ziming You, Ning Ding, Zhiyuan Liu, Maosong Sun, and Hao Peng. From $f(x)$ and $g(x)$ to $f(g(x))$: LLMs learn new skills in RL by composing old ones. In *ICLR*, 2026. 2, 32

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025. 2, 3, 9, 32

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488, 2022. 4

Charlie Zhang, Graham Neubig, and Xiang Yue. On the interplay of pre-training, mid-training, and RL on reasoning language models. *arXiv preprint arXiv:2512.07783*, 2025. URL https://arxiv.org/abs/2512.07783. 32

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. R-Tuning: Instructing large language models to say 'I don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7113–7139, 2024a. 1, 9, 32

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. How language model hallucinations can snowball. In *ICML*, 2024b. 31

Yu Zhang, Peter Belcak, Shizhe Diao, Yonggan Fu, Shaona Ghosh, Morteza Mardani, Eileen Margaret Peters Long, Bei Yu, and Pavlo Molchanov. PROBE: PROcess-based BEnchmark for hallucination detection. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 42303–42320. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.2099. URL https://aclanthology.org/2026.findings-acl.2099/. 9

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. In *NeurIPS*, 2023. 32

Xinyu Zhou, Chang Jin, Carsten Eickhoff, Zhijiang Guo, and Seyed Ali Bahrainian. When silence is golden: Can LLMs learn to abstain in temporal QA and beyond? In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PhUCxfS0yf. 9

Zhiying Zhu, Yiming Yang, and Zhiqing Sun. HaluEval-Wild: Evaluating hallucinations of language models in the wild, 2024. URL https://arxiv.org/abs/2403.04307. 31

A RESEARCH FRAMEWORK: ADDITIONAL DETAILS

This appendix section provides additional methodological details and caveats for the research framework in Sec. 2. The goal is to keep the main paper focused while making the assumptions and controllability trade-offs explicit. Unless otherwise specified, all auxiliary LLM calls described in the appendices use GPT-4o mini. We state the sampling temperature at first use: deterministic extraction, paraphrasing, verification, and judging use temperature 0, while RICHQA generation uses temperature 0.2.

A.1 PASS@k FILTERING AS A PROXY FOR THE KNOWLEDGE BOUNDARY

Why this proxy is conservative. Intuitively, this procedure prioritizes controllability: if a model can answer a fact reliably, it will almost surely pass at least once under sufficiently large k. Conversely, a fact that only appears under extremely rare sampling events is treated as effectively outside the model's retrievable knowledge for practical purposes.

Failure modes and limitations. This proxy is not a perfect oracle. It can fail in at least the following ways:

- **Sampling miss:** the model does contain the fact, but none of the k samples hits the correct surface form.
- **Normalization issues:** the model answers correctly but uses an acceptable alias not captured by the LLM judge.
- **Prompt sensitivity:** the model may retrieve the fact under different phrasing or context but not under the SimpleQA prompt.

However, over-exclusion is acceptable: a conservative partition only reduces sample count, whereas under-exclusion would contaminate the known-knowledge split and undermine the experimental design.

A.1.1 CHOICE OF PASS@32 AND SENSITIVITY ANALYSIS

We use Pass@32 as a practical tradeoff between sampling cost and the strength of the initial knowledge-boundary screen. A fact is retained when none of the sampled responses is judged correct. The fraction excluded by this procedure is 1.72% at $k = 1$, 11.67% at $k = 32$, and 15.23% at $k = 64$. Thus, repeated sampling identifies substantially more retrievable facts than a single response, while doubling the budget from 32 to 64 samples increases the excluded fraction by a further 3.56 percentage points. The boundary remains specific to the prompt, sampler, and evaluator.

To test sensitivity, we repeat the retrieval comparison using Pass@64. Table 3 shows that the 2.1-point RL gain preserves the same qualitative pattern.

Table 3: Sensitivity of held-out retrieval F1 (%) to the initial screening budget. Each row reports the downstream experiment associated with its screening procedure. Δ is the within-row RL gain.

Screening procedure	Post-SFT	Post-RL	Δ
Pass@32	43.3	47.3	+4.0
Pass@64	45.9	48.0	+2.1

Sensitivity to the screening judge. A separate audit regrades the same saved Pass@32 responses without resampling. Among 200 facts retained as non-retrievable, 3 (1.5%) are reclassified as retrievable by the alternative judge, confirming the screen is stable.

A.2 RICHQA CONSTRUCTION PIPELINE

This section provides additional details on how we convert each selected SIMPLEQA sample into a context-rich training example in RICHQA. The conversion is implemented as a grounded LLM-based rewriting pipeline with explicit failure modes, rather than an unconstrained augmentation.

Inputs and outputs. Each SIMPLEQA sample contains (i) a question q, (ii) a reference answer a (the ground truth used for evaluation), and (iii) a set of URLs pointing to Wikipedia passages. The pipeline outputs a rewritten pair $(q_{\text{rich}}, a_{\text{rich}})$ that is *fully grounded* in a selected passage window, and that *strictly subsumes* the original fact: the original answer a must be deducible from the rewritten Q/A.

Step 1: Retrieve and scan source passage. Retrieve the Wikipedia textual content through Wikipedia's web API

Step 2: Find an answer-supporting window. To ensure the rewritten Q/A is grounded, we first identify a passage window that contains evidence for the answer. We use an LLM extractor to return a *verbatim* window centered around the most relevant supporting span. If no window is found across all sources, the sample is skipped. This windowing step discourages the augmentation model from "hallucinating" extra facts during corpus construction.

Step 3: Generate a richer Q/A grounded in the window. Given the extracted window and the original (q, a), we prompt an LLM to generate a richer question and answer. The core prompt template is:

```
System:
You generate richer Q/A pairs grounded in the passage. Return JSON only with keys
'rich_question' and 'rich_answer'.

User:
Given a passage and an original question/answer, produce a richer question and
answer that are fully answerable from the passage. The rich question should ask
for more detail or a broader slice of facts that includes the original answer.
The rich answer should be concise and directly supported by the passage.

### EXAMPLE 1 ###
[Wikipedia Passage]
The IEEE Frank Rosenblatt Award is a Technical Field Award established by the
Institute of Electrical and Electronics Engineers Board of Directors in 2004. ...
Recipients
- 2026: Andrew G. Barto & Richard S. Sutton
- 2025: Yaochu Jin
- ...
- 2010: Michio Sugeno

  [Original Question]
Who received the IEEE Frank Rosenblatt Award in 2010?

[Original Answer]
Michio Sugeno

  [Rich Question]
List all the IEEE Frank Rosenblatt Award recipients from 2010 to 2026.

  [Rich Answer]
- 2026: Andrew G. Barto & Richard S. Sutton
- 2025: Yaochu Jin
- ...
- 2010: Michio Sugeno

### EXAMPLE 2 ###
[Wikipedia Passage]
Mirza Hameedullah Beg (M. H. Beg) (22 February 1913 - 19 November 1988) was the
15th Chief Justice of India, serving from January 1977 to February 1978.
Appointed by Fakhruddin Ali Ahmed.

  [Original Question]
Who appointed the Chief Justice of India, Mirza Hameedullah Beg, in 1977?

[Original Answer]
Fakhruddin Ali Ahmed

  [Rich Question]
Provide the full name, lifespan, position, and term of service of the individual
who appointed Mirza Hameedullah Beg as Chief Justice of India in 1977.

  [Rich Answer]
The individual who appointed Mirza Hameedullah Beg as Chief Justice of India in
1977 was Fakhruddin Ali Ahmed (13 May 1905 - 11 February 1977), who served as
the 5th President of India from 1974 until his death in 1977.
```

```
### YOUR TASK ###
[Wikipedia Passage]
{passage}

[Original Question]
{question}

[Original Answer]
{answer}

Return JSON with keys rich_question and rich_answer only.
```

In practice, this expands single-fact questions into context-rich supervision that mirrors realistic post-training exposure (where facts tend to co-occur in natural documents).

Step 4: Verify coverage and breadth (automatic filter). To discourage trivial overfitting and ensure held-out evaluation remains meaningful, we apply an explicit verification step using an LLM verifier. The verifier returns *pass* only if **both** conditions hold:

- **Coverage.** A careful reader can deduce the original answer a from $(q_{\text{rich}}, a_{\text{rich}})$.
- **Breadth.** The rich question is meaningfully broader than the original; it fails if it is essentially the original question with only one minor extra clause appended.

Samples that fail verification are dropped. This is the key mechanism behind our "avoiding overfitting" objective in Sec. 2.2: the model cannot succeed on SIMPLEQA simply by memorizing a near-duplicate of the evaluation question.

A.3 EVALUATION SPLITS AND SAMPLE SIZES

Table 4 summarizes the evaluation partitions for the main 3B studies.

Table 4: Evaluation splits, sample sizes, and training exposure for the main studies. Evaluation prompts are excluded from RL training wherever RL is applied.

Study	Evaluation N	SFT exposure	In RL training?	Purpose
SFT design (Sec. 3.1)	550	Facts supplied through RichQA	No RL	Cross-format retrieval
Retrieval RL (Sec. 3.3)	550	Facts supplied through RichQA	No; 2,202 disjoint RL training prompts	Held-out RL effect
Abstention RL (Sec. 3.2)	535 (268 answer, 267 abstain)	Fixed answer/refusal targets	Evaluation prompts excluded	Target-conditioned behavior

A.4 EVALUATION DETAILS

This section provides additional details on the evaluation design described in Sec. 2.3, including the LLM judge and edge cases for the F-score metric.

Evaluation metric. Our evaluation utilizes the original metrics from SimpleQA, with an LLM judge assigning the model response into *Correct*, *Incorrect*, and *Not Attempted*, where the *Not Attempted* is explicitly proposed to encourage the LLMs to perform abstention on unknown knowledge. We follow the same metrics of SimpleQA in taking the harmonic mean of *Correct* (fraction of all questions answered correctly) and *Correct Given Attempted* (fraction of attempted questions answered correctly), yielding an F1-score $\text{F1} = \frac{2 \cdot \text{Correct} \cdot \text{CGA}}{\text{Correct} + \text{CGA}}$ that jointly rewards factual accuracy and penalizes overconfident fabrication.

LLM Judge. We evaluate model responses using the appendix-wide judge model at temperature 0, following the prompt template provided by the original SimpleQA benchmark (Wei et al., 2024a).

The template instructs the judge to classify each response as *Correct*, *Incorrect*, or *Not Attempted* given the question and reference answer. We reproduce the prompt verbatim below:

```
Your job is to look at a question, a gold target, and a predicted answer, and then
↪  assign a grade of either ["CORRECT", "INCORRECT", "NOT_ATTEMPTED"].
First, I will give examples of each grade, and then you will grade a new example.


The following are examples of CORRECT predicted answers.
```
Question: What are the names of Barack Obama's children?
Gold target: Malia Obama and Sasha Obama
Predicted answer 1: sasha and malia obama
Predicted answer 2: most people would say Malia and Sasha, but I'm not sure and would
↪ have to double check
Predicted answer 3: Barack Obama has two daughters. Their names are Malia Ann and
↪ Natasha Marian, but they are commonly referred to as Malia Obama and Sasha Obama.
↪ Malia was born on July 4, 1998, and Sasha was born on June 10, 2001.
```
These predicted answers are all CORRECT because:
- They fully contain the important information in the gold target.
- They do not contain any information that contradicts the gold target.
- Only semantic meaning matters; capitalization, punctuation, grammar, and order don't
↪  matter.
- Hedging and guessing are permissible, provided that the gold target is fully included
↪  and the response contains no incorrect information or contradictions.


The following are examples of INCORRECT predicted answers.
```
Question: What are the names of Barack Obama's children?
Gold target: Malia and Sasha
Predicted answer 1: Malia.
Predicted answer 2: Malia, Sasha, and Susan.
Predicted answer 3: Barack Obama does not have any children.
Predicted answer 4: I think it's either Malia and Sasha. Or it could be Malia and
↪ Jackie. Or it could be Joey and Malia.
Predicted answer 4: While I don't know their exact names, I can tell you that Barack
↪ Obama has three children.
Predicted answer 5: It's possible you may mean Betsy and Olivia. However, you should
↪ clarify further details with updated references if necessary. Is that the correct
↪ answer?
Predicted answer 6: It may be the case that Obama's child is named James. However, it's
↪ recommended to confirm the most accurate and updated information since this could
↪ change over time. This model may not always reflect the most current information.
```
These predicted answers are all INCORRECT because:
- A factual statement in the answer contradicts the gold target. Incorrect statements
↪  that have some hedging (e.g., "it is possible that", "although i'm not sure, i
↪  think") are also considered incorrect.


The following are examples of NOT_ATTEMPTED predicted answers.
```
Question: What are the names of Barack Obama's children?
Gold target: Malia and Sasha
Predicted answer 1: I don't know.
Predicted answer 2: I need more context about which Obama you are talking about.
Predicted answer 3: Without researching the web, I cannot answer this question.
↪ However, I can tell you that Barack Obama has two children.
Predicted answer 4: Barack Obama has two children. I know that one of them is Malia,
↪ but I'm not sure about the other one.
```
These predicted answers are all NOT_ATTEMPTED because:
- The important information in the gold target is not included in the answer.
- No statements in the answer contradict the gold target.


Also note the following things:
- For grading questions where the gold target is a number, the predicted answer needs
↪  to be correct to the last significant figure in the gold answer. For example,
↪  consider a question "How many citations does the Transformer Paper have?" with gold
↪  target "120k".
- Predicted answers "120k", "124k", and 115k" are all CORRECT.
- Predicted answers "100k" and "113k" are INCORRECT.
- Predicted answers "around 100k" and "more than 50k" are considered NOT_ATTEMPTED
↪  because they neither confirm nor contradict the gold target.
- The gold target may contain more information than the question. In such cases, the
↪  predicted answer only needs to contain the information that is in the question.
```

```
- For example, consider the question "What episode did Derek and Meredith get legally
↪  married in Grey's Anatomy?" with gold target "Season 7, Episode 20: White Wedding".
↪  Either "Season 7, Episode 20" or "White Wedding" would be considered a CORRECT
↪  answer.
- Do not punish predicted answers if they omit information that would be clearly
↪  inferred from the question.
- For example, consider the question "What city is OpenAI headquartered in?" and the
↪  gold target "San Francisco, California". The predicted answer "San Francisco" would
↪  be considered CORRECT, even though it does not include "California".
- Consider the question "What award did A pretrainer's guide to training data:
↪  Measuring the effects of data age, domain coverage, quality, & toxicity win at
↪  NAACL '24?", the gold target is "Outstanding Paper Award". The predicted answer
↪  "Outstanding Paper" would be considered CORRECT, because "award" is presumed in the
↪  question.
- For the question "What is the height of Jason Wei in meters?", the gold target is
↪  "1.73 m". The predicted answer "1.75" would be considered CORRECT, because meters
↪  is specified in the question.
- For the question "What is the name of Barack Obama's wife?", the gold target is
↪  "Michelle Obama". The predicted answer "Michelle" would be considered CORRECT,
↪  because the last name can be presumed.
- Do not punish for typos in people's name if it's clearly the same name.
- For example, if the gold target is "Hyung Won Chung", you can consider the following
↪  predicted answers as correct: "Hyoong Won Choong", "Hyungwon Chung", or "Hyun Won
↪  Chung".


Here is a new example. Simply reply with either CORRECT, INCORRECT, NOT ATTEMPTED.
↪  Don't apologize or correct yourself if there was a mistake; we are just trying to
↪  grade the answer.
```
Question: {question}
Gold target: {target}
Predicted answer: {predicted_answer}
```

Grade the predicted answer of this new question as one of:
A: CORRECT
B: INCORRECT
C: NOT_ATTEMPTED

Just return the letters "A", "B", or "C", with no text around it.
```

A.4.1 Agreement Between Training Rewards and the LLM Judge

The RL reward uses deterministic answer matching, while final evaluation uses the SimpleQA LLM judge. Table 5 compares the two on the same 550 held-out outputs at each stage.

Table 5: Reported agreement between rule-based reward checks and the SimpleQA LLM judge on the same held-out outputs at each stage.

Stage	Questions	Agreement (%)	Rule correct / judge incorrect	Rule incorrect / judge correct
Post-SFT	550	92.9	8/550	31/550
Post-RL	550	93.1	7/550	31/550

Agreement exceeds 93% at both stages and does not deteriorate after RL.

B Additional Details

All post-training experiments in this work, including both SFT and RL runs, are implemented with VERL (Volcano Engine RL). Using a single post-training stack keeps implementation differences from confounding comparisons across data construction, supervised objectives, and reward-based objectives. This section records the RL-specific setup used in Sec. 3.3; SFT data construction and augmentation details are reported separately in Appendix D.

B.1 RL IMPLEMENTATION DETAILS

GRPO formulation. For each question q, GRPO samples G responses $\{o_1, \ldots, o_G\}$ from the current policy $\pi_{\theta_{\text{old}}}$ and computes the group-normalized advantage:

$$\hat{A}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}, \tag{1}$$

where $\mathbf{r} = \{r_1, \ldots, r_G\}$ are the group rewards. Let $\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}$ denote the importance sampling ratio, ϵ the clipping range, and β the KL penalty. The policy is updated by maximizing the clipped surrogate objective with a KL regularization term; see Shao et al. (2024) for the full derivation.

Unless otherwise noted, retrieval-RL policies are initialized from the RICHQA SFT checkpoint in Sec. 3.1; the base model used for reference/base evaluation is Qwen2.5-3B-Instruct. RL training uses 2,202 SIMPLEQA training questions whose corresponding facts were already included in the RICHQA SFT corpus, and evaluation is performed on a held-out split of 550 SIMPLEQA questions. During RL, all algorithms use the same binary rule-based reward: a response receives $r = 1$ if the normalized reference SIMPLEQA answer is contained in the normalized model response, and $r = 0$ otherwise. Final correctness is evaluated with the SIMPLEQA LLM-judge protocol in Appendix A.4.

Table 6: Core RL training hyperparameters for the retrieval experiments.

Algorithm	Prompt	Response	Batch	Mini-batch	Rollouts	Epochs	GPUs
GRPO	1024	256	64	32	8	20	4 H200
Reinforce++	1024	256	64	32	8	20	4 H200
DAPO	1024	256	64	32	8	20	4 H200
SPIN-DPO	1024	256	64	32	2	20	4 H200

Table 7: Algorithm-specific deviations from the common retrieval-RL setup.

Algorithm	Additional settings
GRPO	Uses vLLM rollouts with $n = 8$ samples per prompt, rollout temperature $T = 1.0$, and the shared sequence lengths in Tab. 6.
Reinforce++	Uses the same prompt/response lengths, batch size, mini-batch size, rollout engine, rollout count, rollout temperature, and binary reward as GRPO, but uses smaller actor/log-probability micro-batches (2 per GPU rather than 8).
DAPO	Uses a generation prompt batch size of $3\times$ the training batch size (192 prompts) before group filtering. We enable group filtering by accuracy, allow up to 10 generation batches, use token-mean loss aggregation, enable the overlong buffer, and set DAPO clipping parameters to $(\epsilon_{\text{low}}, \epsilon_{\text{high}}, c) = (0.20, 0.28, 10.0)$. Rollout temperature is set to 1.0.
SPIN-DPO	Uses two rollouts per prompt. Preference pairs are generated from the model's sampled outputs using the ground-truth answer as the chosen response and the sampled response as the rejected response when appropriate.

For final reporting, we evaluate post-SFT and post-RL checkpoints on the held-out SIMPLEQA questions using the same LLM judge model and prompt as Appendix A.4.

B.2 VARIATION ACROSS TRAINING SEEDS

The comparison among GRPO, Reinforce++, SPIN-DPO, and DAPO checks sensitivity to the optimization method. We additionally report four GRPO runs with independent seeds to examine whether the qualitative retrieval improvement recurs across runs. Table 8 gives each post-RL F1 and its reported within-run gain. These additional runs are reported separately from the shared-checkpoint algorithm comparison in Table 1.

All four runs improve retrieval. Post-RL F1 ranges from 47.5 to 48.4 (spread of 0.9 points), confirming that RL consistently provides a retrieval gain of a few F1 points.

Table 8: Retrieval results for four additional GRPO seeds. F1 is expressed in percent and gains in percentage points. Gains are the reported per-run pre/post differences; they are not calculated by subtracting the 43.3 baseline of the separate algorithm comparison.

Seed	Post-RL F1	Reported gain
0	48.1	+3.03
1	47.5	+2.79
2	48.4	+3.71
3	47.8	+3.07

B.3 MMLU Holdout Evaluation

To assess how factual post-training affects performance beyond the training task, we evaluate the Qwen2.5-3B-Instruct RichQA retrieval pipeline on MMLU at three stages: the initial instruction-tuned checkpoint, after SFT, and after RL. MMLU is held out from our SFT and RL training. The evaluation covers 14,042 questions across 57 subjects and reports macro accuracy.

Table 9: MMLU holdout performance for the Qwen2.5-3B-Instruct RichQA retrieval pipeline. Accuracy is reported in percent; changes are percentage points relative to the initial instruction-tuned checkpoint.

Checkpoint	Macro accuracy (%)	Change from initial
Initial (before our SFT/RL)	64.4	0.0
Post-SFT	38.5	−25.9
Post-RL	45.6	−18.8

MMLU accuracy falls by 25.9 points after RichQA SFT. RL on SimpleQA recovers 7.1 points despite using no MMLU training examples, though accuracy remains 18.8 points below the initial checkpoint.

B.4 Transfer Across Model Scale and Family

We extend the retrieval investigation to Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct to examine whether the observed SFT–RL interaction persists beyond the original 3B backbone. Table 10 reports the RichQA baseline for each model. RL improves held-out F1 by 4.0 points on both Qwen models and by 2.3 points on Llama.

Table 10: Held-out SimpleQA F1 (%) after RichQA SFT and subsequent retrieval RL. All backbones are instruction-tuned. Δ is post-RL minus post-SFT F1, in percentage points.

Backbone	Post-SFT	Post-RL	Δ
Qwen2.5-3B-Instruct	43.3	47.3	+4.0
Qwen2.5-7B-Instruct	49.2	53.2	+4.0
Llama-3.1-8B-Instruct	50.8	53.1	+2.3

We next vary the SFT data form on both larger backbones (Table 11). Question-and-answer paraphrasing gives the highest post-SFT F1 among the tested conditions. However, unfiltered decomposition produces a larger subsequent RL gain on both models. Because unfiltered decomposition can overlap evaluation facts, the clean mixtures provide the more informative control: they start substantially below RichQA after SFT but gain 11.7–13.1 points on Qwen2.5-7B and 17.2–25.2 points on Llama-3.1-8B. The pattern that lower post-SFT F1 can accompany greater RL improvement therefore persists across these additional backbones.

The distinction between RL gain and final performance remains important. Although the clean mixtures gain more from RL, none exceeds the corresponding RichQA baseline after RL. Increasing the mixture ratio also lowers post-SFT F1 on both larger backbones. These results support selecting factual training data by its effect on the complete SFT–RL pipeline: a condition that provides more room for RL improvement need not produce the best final model.

Table 11: Effect of SFT data form on held-out SimpleQA retrieval F1 (%) for the larger backbones. Δ is calculated from the displayed rounded endpoints. The paraphrasing condition uses $k = 3$ as reported for these experiments. Clean-mixture conditions follow the construction in Sec. 3.3.2; unfiltered decomposition retains the evaluation-overlap confound.

	Qwen2.5-7B			Llama-3.1-8B		
SFT condition	SFT	RL	Δ	SFT	RL	Δ
RichQA	49.2	53.2	+4.0	50.8	53.1	+2.3
Q+A paraphrase ($k = 3$)	59.9	60.3	+0.4	60.4	64.6	+4.2
Decomposition, unfiltered	59.5	66.2	+6.7	54.1	60.2	+6.1
Clean mixture ($p = 25\%$)	36.4	49.5	+13.1	30.0	47.2	+17.2
Clean mixture ($p = 50\%$)	31.7	44.8	+13.1	23.3	42.7	+19.4
Clean mixture ($p = 100\%$)	20.8	32.5	+11.7	16.4	41.6	+25.2
Clean decomposition	22.9	27.3	+4.4	14.8	20.6	+5.8

C ABSTENTION: REWARD DETAILS

This appendix records the reward definitions used for RL-based abstention training in Sec. 3.2. We distinguish *answer-target* questions (the model is expected to answer) and *abstention-target* questions (the model is expected to abstain).

C.1 REWARD DEFINITIONS

We use the same outcome categories as SIMPLEQA (Wei et al., 2024a): *Correct*, *Incorrect*, and *Not Attempted* (abstention; *e.g.*, "I don't know"). The rule first checks whether the prompt is an answer-target or abstention-target prompt. For answer-target prompts, a response is marked correct if the normalized reference answer exactly matches, or appears as a substring of, the normalized model response. A detected abstention is always marked *Not Attempted*; empty responses and empty reference answers receive reward 0.

Binary reward. The binary reward gives $+1$ for the desired behavior and 0 otherwise:

$$r_{\text{bin}} = \begin{cases} 1, & \text{answer-target prompt and the response is correct,} \\ 1, & \text{abstention-target prompt and the response abstains,} \\ 0, & \text{otherwise.} \end{cases}$$

Thus, for answer-target prompts, *Incorrect* and *Not Attempted* are both assigned 0.

Ternary reward. The ternary reward is the same as the binary reward on abstention-target prompts, but it penalizes attempted wrong answers on answer-target prompts:

$$r_{\text{ter}} = \begin{cases} 1, & \text{answer-target prompt and the response is correct,} \\ 0, & \text{answer-target prompt and the response abstains,} \\ -p, & \text{answer-target prompt and the response attempts an incorrect answer,} \\ 1, & \text{abstention-target prompt and the response abstains,} \\ 0, & \text{abstention-target prompt and the response attempts an answer.} \end{cases}$$

Unless otherwise stated, $p = 1$. Empty responses and empty reference answers receive reward 0.

C.2 IMPLEMENTATION NOTES

For GRPO in Sec. 3.2, we train and evaluate on SIMPLEQAprompts labeled as answer-target vs. abstention-target according to the same knowledge-boundary partition used to construct RICHQA; this per-prompt label is held fixed throughout SFT and GRPO.

C.2.1 ABSTENTION SFT DATA CONSTRUCTION

To create supervised training data that explicitly teaches abstention, we start from the RICHQAtraining set and create a mixture of *answer-target* and *abstention-target* examples. Using a fixed random seed

Table 12: Binary and ternary reward definitions.

Question type	Reward design	Correct	Not Attempted	Incorrect
Answer-target	Binary	$+1$	0	0
Answer-target	Ternary	$+1$	0	-1
Abstention-target	Binary	0	$+1$	0
Abstention-target	Ternary	0	$+1$	0

(seed $= 42$), we sample 50% of RICHQAtraining examples to be abstention-target; for each selected example, we replace the target output with a randomly chosen abstention template from the list below. The remaining 50% of examples are answer-target and retain the original reference answer.

```
ABSTENTION_TEMPLATES = (
  "I don't know.",
  "I'm not sure.",
  "I'm not certain about that.",
  "I can't recall that.",
  "I don't have that information.",
  "I'm not aware of that.",
  "I'm unsure of the answer.",
  "I don't know the answer to that.",
  "I can't answer that.",
  "I don't have enough information.",
  "I don't have enough information to answer that.",
  "I don't have enough information to answer this question.",
  "I'm not able to determine the answer.",
  "I can't determine that from what I know.",
  "I don't have the answer.",
  "I'm not familiar with that.",
  "That's not something I know.",
  "I don't have knowledge of that.",
  "I don't have that detail.",
  "I'm not sure about the details.",
  "I can't recall the specifics.",
  "I don't have enough context to answer.",
  "I'm unable to answer that.",
  "I don't have sufficient information.",
  "I'm not certain enough to answer.",
  "I don't have a definite answer.",
  "I don't have that information available.",
  "I'm not able to provide an answer.",
  "I don't have the necessary information.",
  "I can't provide an answer to that.",
  "I'm not confident in an answer.",
  "I don't have a reliable answer.",
  "I'm not able to answer that accurately.",
  "I don't have enough knowledge to answer that.",
  "I'm unsure and don't want to guess.",
  "I don't want to guess on that.",
  "I'd rather not guess.",
  "I'm not certain, so I'll say I don't know.",
  "I can't confidently answer that.",
  "I don't have enough information to give a confident answer.",
  "I don't have enough information to answer accurately.",
  "I'm not sure and don't have enough information to answer.",
  "I don't have the information needed to answer that.",
  "I'm unable to find an answer based on what I know.",
```

```
  "I don't have that information in my knowledge.",
  "I don't have access to that information.",
  "I'm not able to recall that information.",
  "I can't recall the answer right now.",
  "I don't have enough information to determine that.",
  "I'm not sure, so I can't answer that.",
)
```

Motivation for a uniform $50/50$ split. A natural alternative to a random split is to choose abstention targets based on the base model's empirical behavior (*e.g.*, marking an example as abstention-target if the model refuses often enough under sampling). We find this approach to be unprincipled because base-model refusal on questions outside the knowledge boundary is highly inconsistent across draws: under a Pass@32 analysis on the SimpleQA refusal-target split, only 0.9% of prompts are *stably refused* (all samples are *Not Attempted*) and 20.9% are *stably answered* (all samples are non-refusals, typically incorrect), while the remaining 78.2% exhibit both behaviors across samples. As a result, any heuristic threshold would largely encode sampling noise as supervision for the majority of the data. A uniform $50/50$ random split is the maximum-entropy choice given this ambiguity: it is agnostic to the base model's unstable prior, yields a clean control for testing whether abstention can be taught via SFT (and later reinforced via RL), and also ensures that even the stably-confident-wrong subset is exposed to abstention targets.

Rule-based reward during GRPO. In our experiments, we avoid using an LLM judge during GRPO due to cost. Instead, we implement a lightweight rule-based reward that assigns outcome categories by string matching.

Correct is determined by checking whether the reference SIMPLEQAanswer string is contained in the model output after normalization (Appendix C.2.3). *Not Attempted* is determined by a refusal/abstention substring heuristic (Appendix C.2.2).

C.2.2 NOT ATTEMPTED HEURISTIC

We normalize the model output using Appendix C.2.3 and check whether it contains any phrase from the following pattern list:

```
def _is_not_attempted(predicted_answer: str) -> bool:
  pred_norm = _normalize(predicted_answer)
  if not pred_norm:
    return False
  return any(p in pred_norm for p in _NOT_ATTEMPTED_PATTERNS)

_NOT_ATTEMPTED_PATTERNS = (
  "i dont know",
  "i do not know",
  "dont know",
  "do not know",
  "im not sure",
  "i am not sure",
  "im unsure",
  "i am unsure",
  "im not certain",
  "i am not certain",
  "i cant recall",
  "i cannot recall",
  "i dont have that information",
  "i do not have that information",
  "i dont have enough information",
  "i do not have enough information",
  "i dont have sufficient information",
  "i do not have sufficient information",
```

```
  "i dont have enough context",
  "i do not have enough context",
  "i cant answer",
  "i cannot answer",
  "im unable to answer",
  "i am unable to answer",
  "im not able to answer",
  "i am not able to answer",
  "i cant determine",
  "i cannot determine",
  "im unable to determine",
  "i am unable to determine",
  "i dont have the answer",
  "i do not have the answer",
  "im not familiar with that",
  "i am not familiar with that",
  "thats not something i know",
  "that is not something i know",
  "i dont have knowledge of that",
  "i do not have knowledge of that",
  "i dont have that detail",
  "i do not have that detail",
  "i cant recall the specifics",
  "i cannot recall the specifics",
  "i dont have a definite answer",
  "i do not have a definite answer",
  "i dont have the necessary information",
  "i do not have the necessary information",
  "i cant provide an answer",
  "i cannot provide an answer",
  "im not confident in an answer",
  "i am not confident in an answer",
  "i dont have a reliable answer",
  "i do not have a reliable answer",
  "i dont want to guess",
  "i do not want to guess",
  "id rather not guess",
  "i would rather not guess",
  "i cant confidently answer",
  "i cannot confidently answer",
  "i dont have enough information to answer accurately",
  "i do not have enough information to answer accurately",
  "i dont have the information needed",
  "i do not have the information needed",
  "i dont have access to that information",
  "i do not have access to that information",
  "i cant recall the answer right now",
  "i cannot recall the answer right now",
)
```

C.2.3 TEXT NORMALIZATION

We normalize text for lightweight exact/sub-string matching by: (i) lowercasing, (ii) stripping XML-style tags (e.g., `<DATE>2010</DATE>`), (iii) normalizing quotes and dashes, (iv) removing punctuation, (v) removing articles (`a`, `an`, `the`), and (vi) collapsing whitespace.

D SFT: IMPLEMENTATION DETAILS

D.1 PARAPHRASING: SURFACE-FORM DIVERSIFICATION

We augment RICHQA by paraphrasing either (i) the *answer* only, or (ii) both the *question and answer* (Q+A), using the appendix-wide LLM setup with temperature $= 0$.

Strategy-conditioned prompting. A known failure mode in LLM-based paraphrasing is frame collapse: all variants reduce to minor synonym swaps within a single syntactic frame (e.g., repeated "`to whom was X given/awarded/presented`"). To counter this, we use a strategy-conditioned rewriting prompt. Each paraphrase is assigned one of seven syntactic strategies: *passive_voice*, *cleft_or_fronting*, *embedded_clause*, *reorder_context*, *nominalize_qword*, *imperative_name*, and *tail_question*. When generating multiple sibling paraphrases for the same source question, we force different strategies across siblings. We reproduce the prompt verbatim as follows. :

```
You paraphrase short factual question-answer pairs for a QA dataset.

    Given a question and its answer, generate {n_paraphrases} paraphrased question-answer
    ↪  pair(s) that:
    1. Preserve EXACTLY the same factual meaning as the original pair -- do not change what
    ↪  is being asked or what fact the answer expresses.
    2. Paraphrase BOTH the question and the answer whenever possible.
    3. Vary the syntactic STRUCTURE of the questions substantially -- not just synonym
    ↪  substitution over the same frame.
    4. Keep each paraphrased answer fully compatible with its corresponding paraphrased
    ↪  question.
    5. Are natural, fluent English.

    Hard constraints:
    - Never negate or invert the meaning of the question or answer.
    - Never add facts, specificity, interpretation, or assumptions not implied by the
    ↪  original.
    - Every question-answer pair must express the same underlying QA relation as the
    ↪  original.
    - A paraphrased answer must be semantically equivalent to the original answer.
    - Do not replace named entities with different entities, aliases, titles,
    ↪  abbreviations, or descriptions unless that equivalence is explicitly contained in
    ↪  the original text.
    - Do not alter numerical values, dates, quantities, units, or other factual values.
    - For atomic answers that have no natural semantic paraphrase -- such as a person's
    ↪  name, a single number, a year, or a short proper noun -- preserve the factual value
    ↪  and only make minimal grammatical changes if natural (for example, "1919" -> "the
    ↪  year 1919"). Do NOT invent variation simply for the sake of changing the string.
    - Avoid the overused "to whom was X given/awarded/presented" passive frame unless
      the chosen strategy is explicitly passive_voice; even then, prefer fresher passives.

    Across the paraphrased questions, use a DIFFERENT strategy for each one. Available
    ↪  strategies:
    - passive_voice: Use passive voice; the original subject becomes the by-phrase or
    ↪  disappears.
    - cleft_or_fronting: Use a cleft or fronted construction.
    - embedded_clause: Embed the query inside an outer clause.
    - reorder_context: Move the contextual modifier to a different position than in the
    ↪  original.
    - nominalize_qword: Replace the question word with a nominal or different question
    ↪  word.
    - imperative_name: Phrase as an imperative starting with "Name", "Identify", or
    ↪  "State".
    - tail_question: Phrase as a near-declarative with a question tail.

    For paraphrasing answers:
    - Preserve the exact underlying fact.
    - Prefer natural grammatical or phrasal reformulations rather than adding information.
    - Noun phrases may be reworded when their meaning can be preserved exactly.
    - Clausal or descriptive answers should use genuinely different wording and structure.
    - Names, numbers, dates, symbols, and other atomic factual answers may remain unchanged
    ↪  when no faithful paraphrase exists.
    - Never infer an alias, expanded name, category, occupation, relationship, or
    ↪  description that was not given.

    Few-shot examples:
    Example A
    Question: Who received the IEEE Frank Rosenblatt Award in 2010?
    Answer: Michio Sugeno
    Five GOOD paraphrased pairs (each question uses a DIFFERENT structure):
```

```
   1. (passive_voice)
      Question: The 2010 IEEE Frank Rosenblatt Award was received by whom?
      Answer: Michio Sugeno
   2. (cleft_or_fronting)
      Question: In 2010, it was which person who received the IEEE Frank Rosenblatt Award?
      Answer: Michio Sugeno
   3. (embedded_clause)
      Question: Can you identify who received the IEEE Frank Rosenblatt Award in 2010?
      Answer: Michio Sugeno
   4. (reorder_context)
      Question: For 2010, which individual received the IEEE Frank Rosenblatt Award?
      Answer: Michio Sugeno
   5. (imperative_name)
      Question: Name the recipient of the 2010 IEEE Frank Rosenblatt Award.
      Answer: Michio Sugeno

   Note: "Michio Sugeno" is an atomic proper-name answer, so it should not be artificially
   rewritten into an unsupported title, description, or alias.

   Example B
   Question: In which year did the Treaty of Versailles end World War I?
   Answer: 1919
   Five GOOD paraphrased pairs (each question uses a DIFFERENT structure):
   1. (passive_voice)
      Question: World War I was ended by the Treaty of Versailles in what year?
      Answer: In 1919.
   2. (cleft_or_fronting)
      Question: It was in which year that the Treaty of Versailles ended World War I?
      Answer: The year 1919.
   3. (embedded_clause)
      Question: Do you know the year in which the Treaty of Versailles ended World War I?
      Answer: 1919.
   4. (tail_question)
      Question: The Treaty of Versailles ended World War I -- in what year?
      Answer: In the year 1919.
   5. (imperative_name)
      Question: State the year the Treaty of Versailles ended World War I.
      Answer: 1919.

   Example C
   Question: What was the primary purpose of the expedition?
   Answer: To study marine life
   Five GOOD paraphrased pairs (each question uses a DIFFERENT structure):
   1. (cleft_or_fronting)
      Question: What was it that the expedition primarily aimed to accomplish?
      Answer: To research marine life.
   2. (embedded_clause)
      Question: Can you state what the expedition's main objective was?
      Answer: Studying marine life.
   3. (nominalize_qword)
      Question: What was the expedition's principal objective?
      Answer: The study of marine life.
   4. (imperative_name)
      Question: Identify the main purpose of the expedition.
      Answer: To investigate marine life.
   5. (tail_question)
      Question: The expedition had a primary goal -- what was it?
      Answer: Researching marine life.

   BAD question paraphrases (do NOT do this -- all five collapse to the same passive
   ↪  frame):
   1. To whom was the award given in 2010?
   2. To whom was the award awarded in 2010?
   3. To whom was the award presented in 2010?
   4. To whom was the award granted in 2010?
   5. To whom was the award handed in 2010?
   This is the failure mode. Synonym swaps over a single frame are NOT structural
   ↪  paraphrases.

   BAD answer paraphrases:
   Original answer: Michio Sugeno
   1. The Japanese researcher Michio Sugeno
   2. Professor Michio Sugeno
   3. Sugeno
   These introduce information, titles, or alternate forms that are not guaranteed by the
   original answer.

   Original answer: 1919
   1. The year after 1918
   2. The first year after World War I
   These add relational information or assumptions instead of simply preserving the
   ↪  answer.
```

```
Output a JSON object with a single key "paraphrases" whose value is a list of
{n_paraphrases} objects. Each object must contain exactly two keys:
- "question": the paraphrased question
- "answer": the corresponding paraphrased answer

No other keys, no commentary.

---
Question: {question}
Answer: {answer}

Output:
```

Number of paraphrases. For the answer-only vs. Q+A comparison we generate five total versions per source sample (original plus four paraphrases). For the k-ablation (Sec. 3.1), k denotes the number of additional paraphrases; we vary $k \in \{1, 2, 4\}$ by taking the first k from the four-paraphrase pool.

D.2 Decomposition: rewriting into atomic QA pairs

Decomposition rewrites a single information-dense QA into multiple atomic factual QA pairs, with the original answer as the sole source of truth. We use the appendix-wide LLM setup under a JSON-only output contract, capping output at **10 subquestions per source**; the mean across our decomposition runs is **6–7** atomic pairs. Key constraints require each subquestion to ask for one concrete fact, be self-contained and non-redundant, and stay within the scope of the original question. We reproduce the prompt verbatim as follows:

```
You decompose information-dense factual questions into atomic factual
subquestions with matching answers.
Each subquestion must:
- ask for one concrete fact
- be answerable with a short factual answer
- be self-contained
- be non-redundant with the others
- stay within the scope of the original question

Use the provided original answer as the source of truth for the subquestion answers.
Do not invent new facts or themes beyond what is implied by the original question and
↪  answer.
Do not produce vague, essay-style, or subjective questions.
Return only valid JSON with key "subquestions".

Original question:
{question}

Original answer:
{answer}

Break this into atomic factual subquestions with matching answers.

Rules:
- Use as many subquestions as needed.
- Each subquestion should target one specific fact.
- Prefer short-answer factual questions.
- Keep them standalone and precise.
- Avoid redundancy and overlap.
- Each answer must match its subquestion and be supported by the original answer.
- Do not write broad questions like "What is the significance of X?" unless that can
be expressed as concrete factual questions.
- Return at most 10 subquestions.

Return JSON:
{
"subquestions": [
{"question": "...", "answer": "..."}
]
}
```

E Raw Numerical Results

This appendix reports raw counts and percentages for the numerical results discussed in the main text.

E.1 Abstention Reward Design

Reward	Stage	Target split	Correct	Incorrect	Not attempted
Binary	Post SFT	Answer targets	109 (40.7%)	104 (38.8%)	55 (20.5%)
	Post SFT	Refusal targets	4 (1.5%)	54 (20.2%)	209 (78.3%)
	Post RL	Answer targets	105 (39.2%)	86 (32.1%)	77 (28.7%)
	Post RL	Refusal targets	3 (1.1%)	26 (9.7%)	238 (89.1%)
Ternary	Post SFT	Answer targets	109 (40.7%)	104 (38.8%)	55 (20.5%)
	Post SFT	Refusal targets	4 (1.5%)	54 (20.2%)	209 (78.3%)
	Post RL	Answer targets	73 (27.2%)	37 (13.8%)	158 (59.0%)
	Post RL	Refusal targets	1 (0.4%)	6 (2.2%)	260 (97.4%)
Ternary (p=0.1)	Post SFT	Answer targets	109 (40.7%)	104 (38.8%)	55 (20.5%)
	Post SFT	Refusal targets	4 (1.5%)	54 (20.2%)	209 (78.3%)
	Post RL	Answer targets	74 (27.6%)	37 (13.8%)	157 (58.6%)
	Post RL	Refusal targets	1 (0.4%)	6 (2.2%)	260 (97.4%)
Ternary (p=0.5)	Post SFT	Answer targets	109 (40.7%)	104 (38.8%)	55 (20.5%)
	Post SFT	Refusal targets	4 (1.5%)	54 (20.2%)	209 (78.3%)
	Post RL	Answer targets	69 (25.7%)	43 (16.0%)	156 (58.2%)
	Post RL	Refusal targets	1 (0.4%)	5 (1.9%)	261 (97.8%)

Table 13: Outcome distributions for reward-design variants on the held-out SimpleQA split. Each cell reports count and percentage within the corresponding target split. For ternary variants with p, incorrect answer-target outputs receive reward $-p$.

E.2 Abstention SFT Paraphrasing

SFT variant	Stage	Target split	Correct	Incorrect	Not attempted
Refuse	Post SFT	Answer targets	109 (40.7%)	104 (38.8%)	55 (20.5%)
	Post SFT	Refusal targets	4 (1.5%)	54 (20.2%)	209 (78.3%)
	Post RL	Answer targets	105 (39.2%)	86 (32.1%)	77 (28.7%)
	Post RL	Refusal targets	3 (1.1%)	26 (9.7%)	238 (89.1%)
Refuse + Q+A para.	Post SFT	Answer targets	136 (50.7%)	99 (36.9%)	33 (12.3%)
	Post SFT	Refusal targets	1 (0.4%)	18 (6.7%)	248 (92.9%)
	Post RL	Answer targets	134 (50.0%)	75 (28.0%)	59 (22.0%)
	Post RL	Refusal targets	1 (0.4%)	7 (2.6%)	259 (97.0%)

Table 14: Outcome distributions for abstention-aware SFT variants on the held-out SimpleQA split under the binary refusal reward. Each cell reports count and percentage within the corresponding target split.

F Related Work

Factuality Benchmarks and Hallucination. A large family of benchmarks measures parametric factual knowledge through short-form QA (Joshi et al., 2017; Kwiatkowski et al., 2019; Yang et al., 2018; Mallen et al., 2023; Lin et al., 2022a; Vu et al., 2023; Wei et al., 2024a; Haas et al., 2025; Arora et al., 2025; Gauba et al., 2025), atomic-claim verification of long-form generations (Min et al., 2023; Wei et al., 2024b; Chen et al., 2025), and source-grounded support checking (Tang et al., 2024; Jacovi et al., 2025; Niu et al., 2024; Rashkin et al., 2023). The SimpleQA *Not Attempted* category, which we adopt, encodes the view that a factual model should answer when it knows and abstain otherwise. Such factual behaviors can be considered from a broader hallucination perspective, with surveys and taxonomies that distinguish intrinsic from extrinsic errors (Ji et al., 2023; Huang et al., 2023; Bang et al., 2025; Maynez et al., 2020) and benchmarks that broaden evaluation to multi-domain or in-the-wild settings (Ravichander et al., 2025; Li et al., 2023; Zhu et al., 2024; Luo et al., 2024; Chen et al., 2023; Chern et al., 2023). A complementary line investigates *why* models hallucinate and how to detect them via consistency, semantic uncertainty, or fact-level probing (Kalai et al., 2025; Zhang

et al., 2024b; Manakul et al., 2023; Kuhn et al., 2023; Farquhar et al., 2024; Sawczyn et al., 2026; Wei et al., 2024c). We complement these benchmarks and detectors by asking how post-training reshapes both retrieval and abstention in a controlled factual-teaching setup.

Post-Training for Factual Knowledge and Abstention. A growing body of work probes how facts live in LLM parameters, ranging from relational probing and capacity studies (Petroni et al., 2019; Roberts et al., 2020; Allen-Zhu and Li, 2023a;b; 2025) to surgical knowledge editing (Meng et al., 2022; 2023). On top of this base knowledge, fine-tuning can improve factuality (Tian et al., 2023a; Lin et al., 2024), but controlled studies show that SFT on unfamiliar examples can *increase* hallucination by teaching the model to produce plausible answers outside what it reliably knows (Gekhman et al., 2024; Kang et al., 2025), motivating the boundary-aware filtering we adopt; critique- and reward-based variants further optimize explicit factuality and verifiability signals (Xie et al., 2025; Wei et al., 2025; Ren et al., 2025). A second axis is whether a model recognizes the limits of its own knowledge: LLMs partially know what they know and can verbalize calibrated confidence (Kadavath et al., 2022; Lin et al., 2022b; Tian et al., 2023b; Yin et al., 2023; Cheng et al., 2024), and a range of objectives turn these signals into reliable abstention via supervised refusal, honesty alignment, refusal-aware datasets, or RL with knowledge-derived rewards (Zhang et al., 2024a; Yang et al., 2024a; Brahman et al., 2024; Kapoor et al., 2024; Xu et al., 2024; An and Xu, 2025; Huang et al., 2025; Sun et al., 2025). Our work treats retrieval and abstention as first-class axes and shows that a refusal-aware reward enables RL to teach abstention reliably even when the SFT-induced retrieval gain is modest.

RL Post-Training and Self-Distillation. Modern post-training combines instruction tuning (Zhou et al., 2023; Lin et al., 2023) with RL or preference optimization, including RLHF and its policy-gradient variants (Ouyang et al., 2022; Schulman et al., 2017; Ahmadian et al., 2024; Hu, 2025), closed-form preference objectives (Rafailov et al., 2023; Ethayarajh et al., 2024), and verifiable-reward RL pipelines that have driven recent reasoning progress (Shao et al., 2024; DeepSeek-AI, 2025; Yu et al., 2025; Lambert et al., 2024). Whether RL incentivizes new capabilities beyond the base model is contested, with analyses pointing to a Pass@k ceiling, skill composition, and SFT–RL asymmetries in memorization, generalization, and forgetting (Yue et al., 2025; Yuan et al., 2026; Chu et al., 2025; Jin et al., 2025; Shenfeld et al., 2025; Swamy et al., 2025; Zhang et al., 2025). Self-distillation offers an intermediate recipe, extending classical and born-again knowledge distillation (Hinton et al., 2015; Furlanello et al., 2018) via on-policy student rollouts (Agarwal et al., 2024) and, for LLMs, casting iterative self-improvement as on-policy distillation connected to RL (Yang et al., 2024b; Shenfeld et al., 2026; Hübotter et al., 2026). These methods are a natural fit for future studies of whether student-generated supervision changes factual retrieval, abstention, or the interaction between the two.

G Limitations and Future Work

Our experiments are designed as controlled mechanistic studies rather than as a full scaling evaluation. We validate the framework on Qwen2.5-3B-Instruct, which lets us run many SFT, RL, and ablation variants under matched conditions, but leaves open whether the same quantitative trade-offs hold for substantially larger base models. Future work should test whether the separation we observe between retrieval and abstention persists across model scales, families, and pretraining mixtures.

The additional retrieval experiments in Sec. B.4 extend this evidence to Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct. They reproduce the dependence of RL improvement on SFT data form across these backbones. The complete matched ablation grid and target-conditioned abstention study remain centered on the 3B model, and broader validation across scales, pretraining mixtures, and factual tasks remains necessary.

The benchmark scope is also intentionally narrow. SimpleQA provides short, high-precision factual probes and a useful *Not Attempted* category, but it does not cover all forms of factuality, such as multi-hop synthesis, long-form generation, domain-specific QA, or rapidly changing knowledge. Additional benchmarks would help determine whether the post-training recipes identified here transfer beyond short-form parametric factual recall.

Finally, self-distillation is a natural future direction for this framework. Future work could study whether on-policy distillation changes knowledge retrieval, abstention, or their interaction when inserted before or after RL.

H BROADER IMPACTS

Positive impacts. This work studies how post-training reduces factual hallucinations in large language models. More reliable factual recall and better-calibrated abstention directly benefit applications in high-stakes settings—including healthcare, legal research, and education—where fabricated facts can cause tangible harm. The framework also provides diagnostic tools (knowledge-boundary identification, held-out evaluation) that researchers can use to measure and improve factual accuracy in any post-training pipeline.

Negative impacts. The same recipe that teaches a model to embed and reliably retrieve new knowledge could in principle be used to implant targeted misinformation into a model's parameters. We consider this risk limited in practice: the technique requires white-box access to model weights and significant compute, and our work focuses specifically on identifying and exposing knowledge gaps rather than concealing them. We do not release fine-tuned model weights.