TRUST OR ABSTAIN? A SELF-AWARE RAG APPROACH UNDER KNOWLEDGE CONFLICTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation (RAG) improves large language models (LLMs) by incorporating external evidence, but it also introduces knowledge conflicts when retrieved contextual knowledge (CK) and parametric knowledge (PK) disagree or are both unreliable. Existing approaches mainly coordinate which source to use, without explicitly asking whether each answer path is correct. We argue that faithful RAG requires LLM self-awareness, namely the ability to recognize the limits of its own knowledge and reasoning. To ground this problem, we construct a model-specific, ground-truth-aligned knowledge conflict benchmark by evaluating LLM backbones on PK-only and CK-conditioned answer paths over approximately 69K query-context instances, drawn from five conflict-QA data sources. We then introduce SABER, a Self-Aware Belief Estimator for RAG that requires no LLM fine-tuning. SABER combines a self-prior with PK-side and CK-side conditional reasoning representations from multi-trace inference, then estimates reliability beliefs with two lightweight predictors to drive a 4-cell decision over trust PK, trust CK, trust either, or abstain. Across four LLM backbones, SABER improves end-to-end accuracy and conflict-specific faithfulness over ten inference-time and fine-tuning baselines, with the largest gains on conflict-heavy datasets. Under abstention, SABER's risk-coverage curve Pareto-dominates every prompt-based abstainer, providing a tunable balance between coverage and answer risk. Our code is available at https://anonymous.4open.science/r/SABER-21EA.

1 INTRODUCTION

Retrieval-augmented generation (RAG) (Lewis et al., 2020; Wang et al., 2026) is a widely-used inference-time strategy that supplies a large language model (LLM) with external evidence, complementing its parametric knowledge (PK) with retrieved contextual knowledge (CK). However, PK and CK are not always consistent, as they may agree, disagree, or both fail. This gives rise to a knowledge conflict problem in which the LLM must decide how to use PK and retrieved CK when generating a faithful answer (Longpre et al., 2021; Xie et al., 2024; Mallen et al., 2023; Xu et al., 2024).

Existing approaches follow three broad trajectories on how to choose between or fuse PK and CK. Source-arbitration methods decide which source should be trusted (Wu et al., 2024; Shi et al., 2024; Huang et al., 2025). In parallel, context-quality methods determine whether the CK should be filtered or repaired (Yu et al., 2024; Zhang et al., 2025; Gao et al., 2026). Retrieval-regulation methods consider whether and how CK should be invoked (Asai et al., 2024; Jeong et al., 2024; Su et al., 2024a; Yao et al., 2025; Guo et al., 2026). Despite these efforts, they do not explicitly and independently ask whether the PK-based answer path and the CK-based answer path would each produce a correct answer with respect to the ground truth. This omission leaves the both-unreliable outcome outside the decision space, pushing the model toward a confidently incorrect and unsupported response rather than an **honest abstention**, i.e., *I don't know*, when neither answer path is reliable.

An open-book exam analogy in Figure 1 illustrates how humans make such decisions. Before the exam, a student has a rough sense of their own knowledge, including what they know and how certain they are. This corresponds to **knowledge-boundary awareness**. During the exam, the student may answer from memory or consult a reference book to form an answer. Since the reference book may be flawed, the student must judge which answer path, if any, is likely to lead to a correct answer. This corresponds to **reasoning-reliability awareness**. If only one path seems reliable, the student should

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.

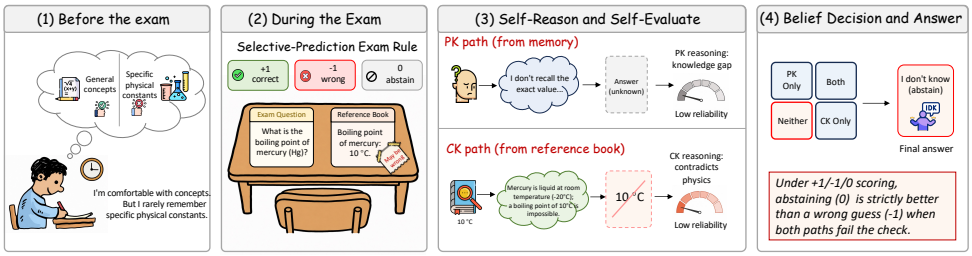

Figure 1: A student facing a $+1$ / -1 / 0 exam rule, where a correct answer earns $+1$, a wrong one -1, and abstention earns 0. The student's memory corresponds to PK, while the potentially flawed reference book corresponds to retrieved CK. The example highlights two forms of self-awareness, namely knowledge-boundary awareness, which reveals when internal knowledge is insufficient, and reasoning-reliability awareness, which detects when retrieved context leads to an implausible answer. When both paths are unreliable, abstention is preferable to an unsupported prediction.

follow that path. If both seem reliable, either can work. If neither does, the honest response is to admit uncertainty or ignorance rather than make an unsupported guess.

In the RAG setting, the LLM acts as the student, PK as memory, and CK as the reference book. Without such self-assessment, two failures recur. Confident hallucination, a knowledge-boundary failure, arises when the model overestimates PK, dismisses corrective evidence, and hallucinates from memory (Mallen et al., 2023; Wu et al., 2024; Chen et al., 2024). Context over-reliance, a reasoning-reliability failure, arises when the model over-relies on CK and adopts misleading retrieved context even when its own knowledge and reasoning could expose the error (Xie et al., 2024; Huang et al., 2025). We trace both vulnerabilities to a **self-awareness** gap, where the model lacks a knowledge-boundary self-prior that reflects what it actually knows (Yin et al., 2023) and a reasoning-reliability self-evaluation that assesses whether its reasoning under a given source is sound. Faithful RAG therefore requires more than source coordination. It requires the model to recognize the boundaries and reliability of its own knowledge and reasoning before committing to a final answer.

To close this gap, we first construct a ground-truth-aligned knowledge conflict benchmark by running four LLM backbones under PK-only and CK-conditioned answer paths, recording the observed correctness against the ground truth for approximately 69K query-context instances per backbone, drawn from five conflict-QA sources. Then, we introduce **Self-Aware Belief Estimator for RAG (SABER)**. Mirroring the open-book exam mechanism above, SABER decomposes self-awareness into a knowledge-boundary self-prior and a reasoning-reliability self-evaluation. It first extracts the self-prior representation from the query-only hidden state to capture the model's inherent knowledge boundary. SABER then generates and self-evaluates multiple reasoning traces under PK and CK, aggregating them through multi-trace test-time inference into robust PK- and CK-conditional states. Two lightweight predictors combine the self-prior with the PK-side and CK-side states to estimate their reliability beliefs, which drive a 4-cell decision over PK, CK, both, or neither. SABER then commits to the more reliable PK or CK answer, or honestly abstains when neither path is reliable.

Our contributions are summarized as follows.

- **Problem Formulation.** We formulate knowledge conflict as a ground-truth-aligned decision problem. Instead of choosing between PK and CK, we independently evaluate the correctness of the PK-only and CK-conditioned paths, yielding a 4-cell decision space over PK, CK, both, and neither.
- **Benchmark Construction.** We construct a benchmark with approximately 69K query–context instances across four LLM backbones and five conflict-QA data sources. Each instance records observed PK-path and CK-path correctness against the ground truth. We further introduce three conflict-specific faithfulness metrics to support evaluation and reveal failures behind overall accuracy.
- **Method.** We introduce SABER, a lightweight self-aware belief reasoning framework requiring no LLM fine-tuning. SABER combines a self-prior with PK- and CK-side conditional reasoning representations from multi-trace inference, then estimates reliability beliefs to drive a 4-cell decision over trust PK, trust CK, trust both, or abstain.
- **Empirical Findings.** SABER improves end-to-end accuracy and conflict-specific faithfulness over ten inference-time and trained baselines, with the largest gains on conflict-heavy datasets. Under abstention, SABER's risk-coverage curve Pareto-dominates every prompt-based abstainer, providing a tunable threshold that lets practitioners trade coverage against answer risk without retraining.

2 A KNOWLEDGE CONFLICT BENCHMARK FOR LLM SELF-AWARENESS

LLM self-awareness cannot be studied without observing the actual behavior of LLMs. Therefore, we construct an model-specific benchmark in which each instance is labeled by observed answer correctness. For every LLM backbone, we record answers produced under a closed-book PK path and an open-book CK path, then label each path by whether it matches the ground truth.

Datasets and LLM Backbones. We construct the query-context pool from five conflict-QA data sources, namely ConFiQA (Bi et al., 2025) with three sub-tasks (QA, MR, MC), ConflictQA-PopQA (Xie et al., 2024) (abbr. ConflictQA), ConflictBank (Su et al., 2024b), and the TriviaQA and Natural Questions (NQ) subsets prepared by Huang et al. (2025). We evaluate four LLM backbones on this pool, namely Qwen-2.5-3B-Instruct, Qwen-2.5-7B-Instruct, Llama-3.1-8B-Instruct, and Llama-3.2-3B-Instruct, yielding approximately 69K ground-truth-aligned PK/CK instances per backbone. Processing details and statistics, the 4-cell distribution, and the labeling prompts are in Appendix A.

Grounding PK and CK in Answer Correctness. For each LLM backbone $\mathcal{M}$ and instance $(q, c, a^\star)$, consisting of a query, a retrieved context, and a ground-truth answer or alias set, we generate two candidate answers. The PK-only answer is produced from the query alone, while the CK-conditioned answer is produced from both the query and the retrieved context.

$$a_{\mathrm{PK}} = \mathcal{M}(\mathrm{prompt}_{\mathrm{PK}}(q)), \qquad a_{\mathrm{CK}} = \mathcal{M}(\mathrm{prompt}_{\mathrm{CK}}(q, c)). \tag{1}$$

We then evaluate both answers against the ground truth using an alias-aware matching function.

$$y_{\mathrm{PK}} = \mathrm{match}(a_{\mathrm{PK}}, a^\star), \qquad y_{\mathrm{CK}} = \mathrm{match}(a_{\mathrm{CK}}, a^\star). \tag{2}$$

A positive match is then confirmed by an LLM judge, and a label is 1 only if both accept the answer (Appendix A). The pair $(y_{\mathrm{PK}}, y_{\mathrm{CK}})$ partitions the benchmark into four reliability cells C_{ab}, where $a, b \in \{0, 1\}$ indicate the correctness of the PK and CK paths. These cells correspond to both paths correct (C_{11}), only PK correct (C_{10}), only CK correct (C_{01}), and neither path correct (C_{00}). These cells provide a ground-truth basis for studying whether the model can form accurate beliefs about its self-awareness.

Evaluation Metrics. Beyond **end-to-end answer accuracy**, we use three conflict-specific faithfulness metrics. **CF (Context Faithfulness)** measures accuracy on instances where only the CK-conditioned answer is correct, testing whether a method follows reliable CK when PK is misleading. **KF (Knowledge Faithfulness)** measures accuracy on instances where only the PK-only answer is correct, testing whether a method preserves correct PK when CK is misleading. **MFS (Macro Faithfulness Score)** averages CF and KF, rewarding methods that handle both conflict directions. The selective-prediction setting in §4.2 additionally evaluates reliability-based refusals using Score, Coverage, Risk on answered (R_C), and abstention F1 (F_1). Formal definitions appear in Appendix D.

3 METHOD: SABER

SABER is a lightweight two-stage belief estimator that operates at test time without fine-tuning the LLM backbone. Building on the benchmark, it extracts a self-prior representation from the frozen LLM (§3.1) and per-side conditional reasoning representations (§3.2), then feeds them into two lightweight predictors that estimate reliability beliefs over PK-side and CK-side correctness. These beliefs drive an internal 4-cell decision that produces SABER's final response, including explicit abstention when neither side is reliable (§3.3). The full pipeline is illustrated in Figure 2.

3.1 SELF-PRIOR REPRESENTATION

The self-prior representation captures the LLM backbone's internal confidence before any retrieved context or reasoning trace is introduced. It reflects the model's self-assessed competence on the current query, signaling whether later reasoning should rely on PK or seek support from CK. Following prior work that reads implicit beliefs from frozen-LLM hidden states (Kadavath et al., 2022; Burns et al., 2022; Azaria & Mitchell, 2023), we extract this signal by feeding the query alone (Appendix B.3) into the frozen LLM $\mathcal{M}$ and reading the hidden state at a fixed prior layer ℓ_p,

$$h_q = \phi_{\mathcal{M}}^{(\ell_p)}\big(\mathrm{prompt}_{\mathrm{prior}}(q)\big). \tag{3}$$

This mirrors how a student, before an exam, judges whether they can answer a question from memory or need to consult a potentially flawed reference book, shaping how heavily they rely on the book. Likewise, the self-prior informs both PK and CK predictions without determining either.

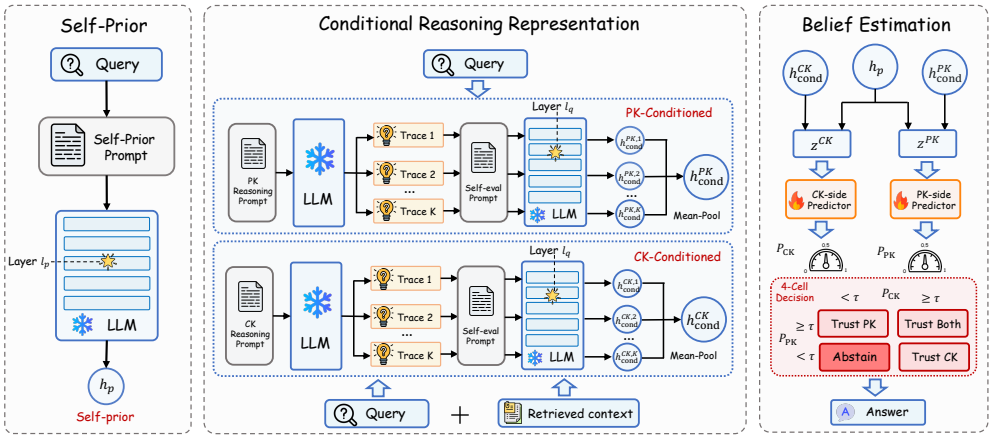

Figure 2: SABER pipeline. The frozen LLM produces a query-only self-prior and per-side conditional reasoning representations via multi-trace generation and self-evaluation. Two lightweight predictors output CK- and PK-beliefs that drive the 4-cell decision with optional abstention.

3.2 Conditional Reasoning Representation

The conditional reasoning representation captures how the model reasons under each source assumption and how it self-evaluates that reasoning after generation. Unlike the self-prior, which reflects pre-reasoning confidence, this representation provides a post-reasoning self-awareness signal. We obtain this signal through a generate-then-evaluate procedure, where the model first generates a source-conditioned reasoning trace and then runs a self-evaluation forward pass over that trace.

Source-conditional Reasoning Trace. For each source, we let the same LLM sample chain-of-thought reasoning traces and answers under the corresponding knowledge condition. On the PK side, the model answers in a closed-book setting from its own knowledge. On the CK side, the model anchors its reasoning only on the retrieved context. We denote the k-th sampled trace under source $s \in \{\mathrm{PK}, \mathrm{CK}\}$ by $t_{s,k}$; the resulting trace provides a side-specific view of how the model walks toward an answer, revealing which knowledge it draws on and how it integrates or resists the available evidence. The exact reasoning prompts are given in Appendix B.4 and Appendix B.5.

Self-evaluation Forward Pass. To estimate confidence in the generated reasoning trace, we ask the LLM to self-evaluate it in a second forward pass, indicating whether the trace is likely to support a correct answer under the assumed source (Kadavath et al., 2022; Mei et al., 2025). Specifically, we read per-token hidden states at a fixed conditional layer ℓ_c over the reasoning span (Ni et al., 2026) and mean-pool them into a per-trace vector,

$$h_{\mathrm{cond}}^{(s,k)} = \bar{\phi}_{\mathcal{M}}^{(\ell_c)}\big(\mathrm{prompt}_{\text{self-eval}}(q, c, t_{s,k})\big), \tag{4}$$

where $t_{s,k}$ denotes the k-th reasoning trace under source s, and $\bar{\phi}_{\mathcal{M}}^{(\ell_c)}(\cdot)$ averages hidden states at layer ℓ_c over the reasoning span. This self-evaluation may reflect the trace's coherence, internal consistency, and use of available knowledge, which together shape the LLM's confidence in the trace. It mirrors how students, when uncertain about memory or a flawed reference book, reason through a problem and judge whether the answer is plausible. The self-evaluation prompt is in Appendix B.6.

Multi-trace Test-time Inference. A single trace is one sample from a stochastic generator and may run into a dead end, hallucinate a step, or react to surface features of c. Following multi-trace test-time inference (Wang et al., 2023b), we sample K independent reasoning traces per side and mean-pool their per-trace vectors into a robust per-side conditional state,

$$h_{\mathrm{cond}}^{(s)} = \mathrm{MeanPool}\big(h_{\mathrm{cond}}^{(s,1)}, h_{\mathrm{cond}}^{(s,2)}, \ldots, h_{\mathrm{cond}}^{(s,K)}\big), \quad s \in \{\mathrm{PK}, \mathrm{CK}\}. \tag{5}$$

3.3 Belief Estimation and Decision with Abstention

This section turns SABER's two self-awareness signals into an actionable belief decision. We first combine the self-prior and conditional reasoning representations into joint representations for the PK and CK sides. We then estimate separate PK-side and CK-side reliability beliefs using two lightweight predictors and map them into a 4-cell decision space with explicit abstention.

Joint Representation. For each side s, we combine the self-prior with the corresponding conditional reasoning representation to form a side-specific joint representation,

$$z^{(s)} = f_{\text{fuse}}\big(h_q,\, h^{(s)}_{\text{cond}}\big), \quad s \in \{\text{PK}, \text{CK}\}, \tag{6}$$

where f_{fuse} concatenates the self-prior with the side-specific reasoning representation. Notably, the same self-prior plays complementary roles across the two sides. For PK, it reflects the model's initial confidence in its own parametric knowledge. For CK, it reflects how much the model may need to lean on external evidence. Combined with the corresponding source-conditioned reasoning state, this prior yields a side-specific joint representation used to estimate the reliability of each answer path.

Per-side Correctness Predictor. Because SABER's self-awareness signals are anchored in ground-truth answer correctness, the PK-side and CK-side paths give rise to two independent correctness questions whose outcomes span four possible combinations. Along this line, we employ two lightweight correctness predictors, one for each side, each taking the concatenation of the two joint representations as input and outputting a scalar reliability belief,

$$\hat{p}_s = f_{\text{pred},s}\big([z^{(\text{PK})};\, z^{(\text{CK})}]\big), \quad s \in \{\text{PK}, \text{CK}\}. \tag{7}$$

The shared input allows each predictor to draw on evidence from both sides, such as using CK-side unreliability to sharpen the PK-side judgment, while the independent outputs preserve all four reliability outcomes and feed directly into the 4-cell decision space. The two predictors are trained independently with binary cross-entropy on y_{PK} and y_{CK} from our constructed ground-truth-aligned benchmark. The LLM backbone remains frozen throughout. Only the parameters of the predictors are updated; the predictors themselves add few trainable parameters and negligible inference cost on top of the representation-extraction passes described above. Architecture and training hyperparameters are reported in Appendix B.

4-Cell Decision with Abstention. Thresholding the two reliability beliefs $(\hat{p}_{\text{PK}}, \hat{p}_{\text{CK}})$ at τ assigns each instance to one of the four cells defined in §2. For the three reliable cells, SABER returns the PK answer on C_{10}, the CK answer on C_{01}, or the candidate with the higher reliability belief on C_{11}, all without invoking the LLM again. The C_{00} cell, where neither side is reliable, supports two deployment modes. The default mode falls back to the higher-belief candidate, preserving full coverage for low-stakes settings, while the abstention mode returns *I don't know*, suiting high-stakes settings where unsupported answers are costly. The threshold τ controls how often SABER commits to an answer and, under abstention, trades coverage against answer risk. As shown in §4.2, sweeping τ produces a continuous risk-coverage curve that existing methods cannot match, making τ a deployment-relevant control knob rather than a purely internal hyperparameter.

4 Experiments

We run experiments on the seven datasets in our ground-truth-aligned benchmark from §2. We compare against ten methods from four categories. **(1) Vanilla baselines.** Vanilla LLM (Closed-Book) and Vanilla RAG. **(2) Verbal self-evaluation.** Internal-Eval, Context-Eval, Implicit-SCR, and Explicit-SCR (Huang et al., 2025), which prompt the LLM to judge source reliability or its own confidence. **(3) Probability- and filtering-based heuristics.** TPC (Wu et al., 2024) which compares the token-level probability, and TACS-P, a prompt-based adaptation of truth-aware context selection (Yu et al., 2024) that filters unreliable context before answering. **(4) Trained methods.** CR-DPO (Huang et al., 2025), a DPO-style extension of Explicit-SCR, and R-Tuning (Zhang et al., 2024), which fine-tunes the LLM with uncertain instances. Descriptions are provided in Appendix C. We evaluate all methods using the benchmark metrics from §2, namely end-to-end answer accuracy (Acc), Context Faithfulness (CF), Knowledge Faithfulness (KF), and Macro Faithfulness Score (MFS). Selective-answering metrics (Score, Coverage, R_C, abstention F_1) are introduced in §4.2 and used only there to study abstention decisions. Computational cost is reported in Appendix G.

4.1 Main Results

Tables 1 and 2 report Acc, CF, KF, and MFS on the five sources across four LLM backbones from Llama and Qwen. We highlight four observations.

(1) SABER consistently outperforms all inference-time baselines on Acc and MFS. Across the 20 backbone-dataset settings, SABER achieves the strongest Acc in most settings and the strongest MFS in every setting. Its gains over the best inference-time baseline reach +10.6% Acc on (Llama-3.1-8B,

Table 1: Main results on Llama-3.1-8B and Llama-3.2-3B. We report Acc (↑), CF, KF, and MFS (↑), all as percentages, on the five sources. ConFiQA combines its three sub-tasks (QA / MR / MC) into a single dataset due to the limited space (breakdown is in Appendix E.1). Since CF or KF alone can be saturated by always trusting one source, MFS is the primary conflict-faithfulness metric. For Acc and MFS, the best is highlighted with a **blue** background and the second-best with a green background.

Method	ConFiQA				ConflictQA				ConflictBank				TriviaQA				NQ			
	Acc↑	CF	KF	MFS↑	Acc↑	CF	KF	MFS↑	Acc↑	CF	KF	MFS↑	Acc↑	CF	KF	MFS↑	Acc↑	CF	KF	MFS↑
Llama-3.1-8B																				
Closed-Book	31.2	7.9	100.0	53.9	22.7	3.0	100.0	51.5	23.5	0.6	100.0	50.3	80.8	16.7	100.0	58.3	41.3	6.3	100.0	53.2
Vanilla RAG	57.0	100.0	17.5	58.7	46.8	100.0	7.5	53.7	69.4	100.0	1.7	50.9	57.1	100.0	4.2	52.1	46.7	100.0	4.1	52.0
Internal-Eval	45.4	55.3	62.8	59.0	31.9	37.6	72.1	54.9	61.2	78.1	38.1	58.1	78.8	50.0	85.9	68.0	49.6	66.7	59.2	62.9
Context-Eval	39.2	34.5	85.7	60.1	31.7	36.6	79.6	58.1	58.9	61.6	90.3	76.0	75.4	62.5	73.2	67.9	45.9	52.4	67.3	59.9
Implicit-SCR	53.7	78.6	43.9	61.3	47.3	89.5	36.1	62.8	29.3	39.6	5.1	22.4	65.0	91.7	36.6	64.1	50.4	88.9	32.7	60.8
Explicit-SCR	40.0	45.6	75.3	60.5	40.9	73.9	52.4	63.1	22.2	23.8	28.7	26.3	63.3	83.3	56.3	69.8	46.3	73.0	53.1	63.0
TPC	55.1	86.6	60.5	73.6	47.0	92.7	34.7	63.7	69.6	92.3	36.4	64.3	68.8	91.7	40.8	66.3	51.7	93.7	36.7	65.2
TACS-P	43.6	59.7	30.5	45.1	42.9	90.1	12.2	51.2	33.8	46.8	3.7	25.2	53.8	87.5	5.6	46.6	47.5	92.1	20.4	56.2
R-Tuning	20.7	9.1	32.7	20.9	10.5	2.6	28.3	15.5	2.3	0.0	8.4	4.2	72.5	28.6	82.1	55.4	33.9	18.8	58.8	38.8
CR-DPO	47.9	60.1	77.1	68.6	**53.8**	94.9	81.6	88.2	52.3	50.3	85.8	68.0	84.2	95.8	88.7	**92.3**	**61.2**	79.4	73.5	**76.4**
SABER (Ours)	**58.4**	92.8	72.2	**82.5**	53.5	97.0	85.0	**91.0**	**80.2**	97.0	88.1	**92.5**	**85.2**	66.7	98.6	82.6	52.1	73.0	65.3	69.2
Llama-3.2-3B																				
Closed-Book	23.9	5.5	100.0	52.7	13.9	0.8	100.0	50.4	16.8	0.3	100.0	50.1	73.3	3.2	100.0	51.6	36.4	7.6	100.0	53.8
Vanilla RAG	50.3	100.0	9.7	54.8	42.0	100.0	5.4	52.7	63.5	100.0	0.8	50.4	52.1	100.0	7.1	53.6	43.8	100.0	4.7	52.3
Internal-Eval	41.1	62.3	60.7	61.5	37.2	74.1	65.0	69.5	59.6	81.0	62.1	71.5	72.1	61.3	78.6	69.9	43.8	56.1	67.4	61.8
Context-Eval	46.6	77.4	63.7	70.6	29.5	51.9	68.1	60.0	49.9	61.5	82.3	71.9	57.6	80.6	26.7	53.7	39.9	57.6	44.2	50.9
Implicit-SCR	41.7	68.3	24.9	46.6	41.3	88.1	20.3	54.2	64.7	93.3	7.3	50.3	48.2	83.9	11.5	47.7	43.0	91.0	11.4	51.2
Explicit-SCR	28.1	32.8	70.4	51.6	27.6	54.6	68.5	61.6	27.8	33.2	50.0	41.6	52.1	61.3	42.9	52.1	30.6	60.6	27.9	44.3
TPC	50.7	90.6	56.9	73.7	41.4	92.4	37.8	65.1	57.0	85.0	24.2	54.6	67.9	100.0	48.8	74.4	47.5	93.9	30.2	62.1
TACS-P	38.6	63.7	20.4	42.0	36.7	82.9	6.2	44.6	60.7	90.0	4.0	47.0	51.4	90.3	21.4	55.9	40.9	84.8	13.6	49.2
R-Tuning	3.0	0.9	7.0	4.0	0.4	0.0	0.9	0.5	0.2	0.0	0.8	0.4	42.5	15.1	55.9	35.5	23.1	10.9	45.7	28.3
CR-DPO	34.6	41.7	81.5	61.6	39.8	79.2	85.6	82.4	39.2	44.5	69.4	56.9	63.7	67.7	76.2	72.0	40.1	56.1	55.8	55.9
SABER (Ours)	**54.1**	94.8	79.4	**87.1**	**46.9**	97.9	82.9	**90.4**	**70.9**	97.6	82.3	**89.9**	**77.9**	77.4	85.7	**81.6**	**54.1**	90.9	69.8	**80.3**

ConflictBank) and +27.3% MFS on (Llama-3.1-8B, ConflictQA). On average, SABER leads by +3.4% in Acc and +15.4% in MFS.

(2) SABER is competitive with fully fine-tuned CR-DPO while keeping the LLM frozen. CR-DPO's gains concentrate on larger LLMs, whereas SABER matches or exceeds CR-DPO in most settings and strictly dominates it on both 3B models across on Acc and MFS. This shows that SABER's hidden-state reliability signal can rival full DPO fine-tuning, especially on smaller models.

(3) SABER's MFS gains substantially exceed its Acc gains, indicating that its main advantage is conflict resolution rather than raw accuracy. Several baselines can obtain reasonable Acc by always favoring CK, as in Vanilla RAG, or PK, as in Closed-Book, but they sacrifice either KF or CF. SABER maintains high CF and KF simultaneously, deciding when to trust context and when to preserve internal knowledge. Its only inference-time Acc loss on (Qwen-2.5-7B, ConFiQA) is offset by a +15.7% MFS gain, illustrating the bias-vs-symmetry trade-off of prompt-only methods.

(4) Acc gains are larger on Llama than on Qwen, while MFS gains remain uniform across families. SABER's mean Acc lead is +4.9% on Llama and +2.0% on Qwen, while mean MFS gains stay between +14% and +16% on both families. The narrower Qwen Acc margin reflects the stronger Implicit-SCR baseline on Qwen rather than model scale, and the uniform MFS gain shows that SABER's conflict-resolution advantage transfers across families even when the Acc gap narrows.

4.2 Selective Answering with Abstention

We further evaluate whether SABER and prompt-based baselines can refuse to answer when neither PK nor CK is reliable. For a fair comparison, we augment each prompt-based baseline with an explicit instruction to respond *I don't know* as an option. SABER abstains when both probe scores fall below a shared threshold $\tau = 0.5$, the natural midpoint of the $[0, 1]$ belief scale. This default is used across all backbones, with no per-backbone tuning. Following the selective-prediction framework of Geifman & El-Yaniv (2017), each answered instance receives $+1$ if correct and -1 if wrong, while abstentions receive 0. We report four metrics in percent. **Score** aggregates this utility and measures the answer-abstain trade-off. **Coverage** (Cov) is the fraction of questions answered. **Risk on answered** (R_C) is the error rate among answered questions. **Abstention F1** (F_1) measures how well abstentions match the gold abstention set, namely instances where neither source is correct.

Main comparison. Table 3 reports results across four backbones. We highlight three observations.

Table 2: Main results on Qwen-2.5-7B and Qwen-2.5-3B. Same setup as Table 1.

Method	ConFiQA				ConflictQA				ConflictBank				TriviaQA				NQ			
	Acc↑	CF	KF	MFS↑	Acc↑	CF	KF	MFS↑	Acc↑	CF	KF	MFS↑	Acc↑	CF	KF	MFS↑	Acc↑	CF	KF	MFS↑
Qwen-2.5-7B																				
Closed-Book	30.1	4.1	100.0	52.0	14.6	1.3	100.0	50.6	18.8	1.1	100.0	50.6	56.7	9.4	100.0	54.7	26.4	2.6	100.0	51.3
Vanilla RAG	52.1	100.0	6.7	53.4	44.5	100.0	7.0	53.5	66.5	100.0	0.0	50.0	52.9	100.0	3.9	52.0	47.5	100.0	0.0	50.0
Internal-Eval	51.6	83.3	55.6	69.5	38.2	72.1	63.0	67.6	59.4	78.9	43.4	61.1	65.4	71.7	84.3	78.0	43.4	70.5	58.1	64.3
Context-Eval	44.8	53.6	89.2	71.4	31.0	55.9	50.0	53.0	60.7	82.5	42.7	62.6	59.2	79.2	51.0	65.1	44.6	75.6	48.4	62.0
Implicit-SCR	**59.5**	90.4	59.9	75.2	47.8	94.5	36.0	65.3	68.6	93.1	13.3	53.2	67.1	100.0	52.9	76.5	52.5	93.6	35.5	64.5
Explicit-SCR	40.6	57.3	44.7	51.0	34.8	70.7	30.0	50.3	58.6	81.2	28.7	54.9	50.0	62.3	51.0	56.6	41.3	65.4	54.8	60.1
TPC	52.8	92.7	43.9	68.3	44.3	96.0	27.0	61.5	66.4	94.7	25.2	59.9	58.8	92.5	33.3	62.9	50.0	97.4	29.0	63.2
TACS-P	45.4	67.1	35.1	51.1	41.9	84.0	13.0	48.5	65.8	90.7	13.3	52.0	55.4	94.3	19.6	57.0	47.9	89.7	19.4	54.5
R-Tuning	38.0	18.3	78.2	48.3	15.9	4.4	70.9	37.7	18.5	5.1	55.0	30.1	67.5	34.7	93.1	63.9	33.5	16.7	81.8	49.3
CR-DPO	51.6	83.3	57.9	70.6	47.8	98.5	61.0	79.8	62.8	85.7	41.3	63.5	**72.5**	92.5	86.3	**89.4**	49.6	78.2	71.0	74.6
SABER (Ours)	57.0	91.8	90.1	**90.9**	**49.6**	99.3	90.0	**94.6**	**73.5**	96.4	75.5	**86.0**	71.2	86.8	88.2	87.5	**53.7**	92.3	71.0	**81.6**
Qwen-2.5-3B																				
Closed-Book	24.8	4.7	100.0	52.3	9.1	1.1	100.0	50.6	13.6	0.3	100.0	50.1	47.5	8.2	100.0	54.1	27.3	2.7	100.0	51.4
Vanilla RAG	52.7	100.0	9.1	54.5	42.1	100.0	7.0	53.5	65.9	100.0	0.9	50.4	51.7	100.0	8.7	54.3	45.5	100.0	6.5	53.2
Internal-Eval	45.8	74.1	40.2	57.1	30.6	64.9	57.9	61.4	41.8	53.5	51.7	52.6	63.3	86.9	78.3	82.6	41.7	58.1	83.9	71.0
Context-Eval	30.8	22.0	96.7	59.4	12.3	13.8	70.2	42.0	39.6	47.3	70.7	59.0	49.6	21.3	95.7	58.5	31.0	29.7	71.0	50.3
Implicit-SCR	53.2	86.3	33.0	59.6	43.4	92.9	19.3	56.1	66.4	93.2	5.2	49.2	53.3	93.4	26.1	59.8	48.3	95.9	19.4	57.7
Explicit-SCR	39.8	69.2	35.5	52.4	31.8	78.4	14.0	46.2	47.1	72.1	7.8	39.9	39.6	83.6	17.4	50.5	31.4	71.6	9.7	40.6
TPC	52.4	93.3	39.1	66.2	41.6	96.7	40.4	68.5	63.0	90.9	24.1	57.5	55.4	95.1	30.4	62.8	46.7	98.6	25.8	62.2
TACS-P	45.9	81.3	17.4	49.4	39.5	90.9	3.5	47.2	62.3	89.0	0.0	44.5	50.4	90.2	15.2	52.7	45.0	94.6	12.9	53.7
R-Tuning	41.8	31.2	76.0	53.6	10.4	6.4	46.2	26.3	8.6	1.9	40.9	21.4	56.2	35.5	80.8	58.2	28.1	11.7	78.8	45.3
CR-DPO	48.3	85.4	34.8	60.1	38.6	92.2	31.6	61.9	48.8	69.1	32.8	50.9	54.2	83.6	60.9	72.2	36.0	73.0	22.6	47.8
SABER (Ours)	**55.9**	94.0	76.4	**85.2**	**44.0**	98.0	75.4	**86.7**	**71.6**	97.6	73.3	**85.4**	**63.7**	85.2	87.0	**86.1**	**50.0**	97.3	45.2	**71.2**

(1) SABER achieves the best abstention-aware performance on every backbone. It obtains the highest Score, lowest Risk on answered, and highest abstention F1, with Implicit-SCR consistently second on Score. The stable gap suggests that prompt-based abstention can only elicit what the model verbalizes about uncertainty, while reliable refusal requires access to internal belief states.

(2) SABER's Score advantage grows as backbone size decreases. The gain over the strongest baseline increases from $+18.5\%$ on Llama-3.1-8B to $+24.6\%$ on Qwen-2.5-7B, $+27.1\%$ on Llama-3.2-3B, and $+32.0\%$ on Qwen-2.5-3B. This pattern suggests that prompt-based *I don't know* behavior depends on instruction-following capacity and degrades on smaller models, whereas SABER's hidden-state-based decision avoids this dependency.

(3) Judgment-style baselines fail structurally under abstention. Context-Eval, Explicit-SCR, and Internal-Eval answer most questions and achieve much lower abstention F1 than SABER. Their verbal-judgment prompts still frame the task as choosing whether to trust PK or CK, leaving no natural role for abstention. Reliable refusal therefore requires belief estimation, not an instruction.

Risk-coverage analysis. The main comparison uses one operating point. To test whether SABER's advantage is robust, we sweep its abstention threshold τ over $[0.05, 0.95]$ in steps of 0.05 and trace the full risk-coverage curve. Each baseline yields a single Coverage-Risk point determined by its prompt template. Figure 3 visualizes this comparison, and full sweep numbers are reported in Appendix E.2.

(4) SABER's risk-coverage curve Pareto-dominates every prompt-based baseline on all four backbones. At Implicit-SCR's Coverage, SABER reduces Risk on answered by 19% to 53% relative across the four backbones, with the largest reduction on Llama-3.1-8B. Prompt-based methods commit to one operating point, whereas SABER's continuous threshold τ provides a deployment-relevant knob for trading willingness to answer against answer risk without retraining.

(5) SABER's Score remains stable over $\tau \in [0.40, 0.55]$. The per-backbone fluctuation is at most 0.009, showing that the default $\tau = 0.5$ does not require per-backbone recalibration. This stability is consistent with the view that confident instances concentrate away from the decision boundary, making Score insensitive to small threshold changes.

4.3 Additional Analysis and Discussions

Ablation study. To assess which components drive SABER's performance, we compare the full method with four reduced variants: a prior-only baseline (Prior, using only the query-side hidden state), two conditional-only variants (C_1 and C_3, using one or three mean-pooled traces), and a single-trace joint variant ($P{+}C_1$). Our full method, SABER, is the joint variant with $K{=}3$, equivalent to $P{+}C_3$. From Figure 4, we highlight four observations.

Table 3: Selective answering results across four backbones over the whole benchmark ($\tau = 0.5$). We report Score (↑), Coverage (Cov), Risk on answered (R_C, ↓), and abstention F1 (F_1, ↑), all as percentages. For Score, R_C, and F_1, within each backbone block the best value is highlighted with a **blue** background and the second-best with a green background; Coverage receives no markings.

Method	Llama-3.1-8B				Qwen-2.5-7B				Llama-3.2-3B				Qwen-2.5-3B			
	Score↑	Cov	R_C ↓	F_1 ↑	Score↑	Cov	R_C ↓	F_1 ↑	Score↑	Cov	R_C ↓	F_1 ↑	Score↑	Cov	R_C ↓	F_1 ↑
Closed-Book	−3.0	50.6	53.0	50.1	1.7	23.2	46.4	58.0	−5.0	37.0	56.8	57.3	−0.3	6.3	52.4	62.3
Vanilla RAG	19.8	72.9	36.4	49.6	11.1	87.1	43.6	27.9	8.2	79.4	44.9	43.1	6.2	79.5	46.1	35.8
Internal-Eval	0.5	89.2	49.7	20.0	0.8	45.3	49.2	50.4	−11.9	72.5	58.2	38.3	−6.8	68.4	54.9	40.8
Context-Eval	−7.6	99.2	53.8	3.3	−1.8	77.6	51.1	27.3	−34.6	100.0	67.3	0.1	−33.5	99.3	66.9	1.9
Implicit-SCR	25.4	59.8	28.8	56.2	18.8	81.6	38.5	39.8	11.4	70.7	41.9	51.0	7.6	73.0	44.8	42.0
Explicit-SCR	−8.7	97.0	54.5	8.3	11.4	43.7	36.9	55.4	−34.2	97.6	67.5	4.9	−5.9	43.2	56.8	54.5
TACS-P	13.6	64.4	39.5	49.9	8.7	68.3	43.7	42.8	5.4	68.7	46.1	47.5	5.1	71.2	46.4	40.6
SABER (Ours)	**43.9**	65.0	**16.2**	**72.5**	**43.4**	58.5	**12.9**	**78.0**	**38.5**	58.6	**17.2**	**76.8**	**39.6**	60.0	**17.0**	**76.6**

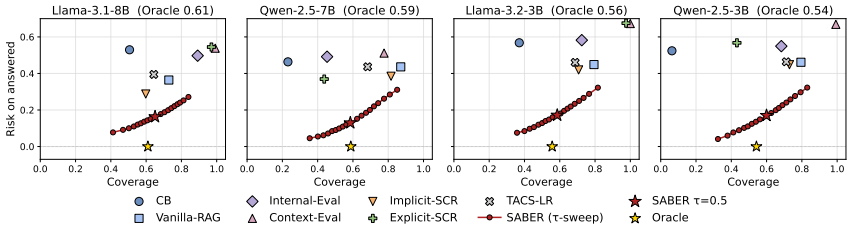

Figure 3: Risk-coverage curves. Each panel corresponds to one LLM backbone, and lower-left indicates better performance. SABER (red curve) sweeps its abstention threshold τ over $[0.05, 0.95]$, while each prompt-based baseline yields a single (Coverage, Risk on answered) point. The dark red star marks SABER at the default $\tau = 0.5$, and the gold star marks the Oracle.

(1) Each component contributes to SABER's overall performance. Every reduced variant outperforms or matches SABER in both MFS and Acc across all backbones. Among the single-component removals, dropping multi-trace averaging from the joint configuration causes the largest MFS reduction, up to 3.2% on Qwen-2.5-3B, indicating that the components capture complementary signals.

(2) The source-conditional reasoning representation drives most of the gain. Conditional-only variants C_1 and C_3 improve over the prior-only baseline by an average of $+2.8\%$ Acc and $+13.4\%$ MFS across the four backbones. This indicates that the post-reasoning, source-conditioned state carries most reliability information, while the query-only prior is useful but insufficient alone.

(3) Multi-trace averaging benefits all backbones, with the largest gain on the smallest model. For example, on Qwen-2.5-3B, increasing K from 1 to 3 in the joint variant, from $P{+}C_1$ to SABER, raises MFS from 84.1% to 87.3% ($+3.2\%$), the largest gain across the four backbones. This suggests that mean-pooling multiple conditional traces reduces noise from single samples, and the noise reduction matters most when the underlying LLM backbone is smaller.

(4) The gain is not explained by test-time compute or by access to labeled data. Majority voting over the same six reasoning traces that SABER uses, and few-shot prompting with labeled examples from the same training partition, both trail SABER on nearly every backbone–dataset cell (Appendix E.3). The advantage comes from reading path-level reliability out of the hidden states rather than from more sampling or more supervision.

Parameter sensitivity. We examine SABER's sensitivity to the prior layer ℓ_p and conditional layer ℓ_c. Figure 5 sweeps both layers at stride 2 over the two Llama backbones. The best layer pair lies within a broad high-accuracy plateau in the mid-to-late depth range. For both Llama-3.1-8B and Llama-3.2-3B, the 90th-percentile region covers more than one third of the grid, and the selected layer pair (red square) lies inside this region rather than at an isolated peak. The selected layer pair on each Llama backbone falls between roughly 40% and 70% of the total depth L, sitting within the mid-to-late depth band that prior probing work identifies as semantically integrated (Azaria & Mitchell, 2023; Burns et al., 2022). Thus, SABER is not sensitive to fine-grained layer tuning and uses depths known to be informative for internal belief estimation.

Cross-dataset generalization. We further evaluate SABER on datasets unseen during training under a leave-one-dataset-out protocol, together with few-shot target-domain adaptation and a data-

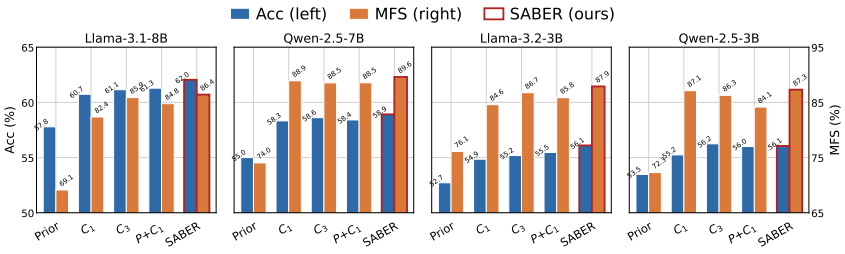

Figure 4: Ablation study with four variants across four LLMs over the whole benchmark. Each panel reports answer accuracy and MFS. The SABER bars are outlined in red.

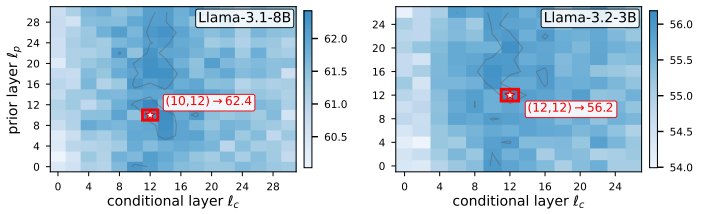

Figure 5: SABER layer sensitivity on Llama. Heatmaps show answer accuracy (%) across (ℓ_p, ℓ_c) at stride 2. Darker cells indicate higher accuracy, grey contour traces the 90th-percentile region, and red squares mark selected layer pairs. Qwen results appear in Appendix E.4.

efficiency study, as illustrated in Appendix F. SABER retains most of its performance on datasets whose contexts are derived from real passages, and a few hundred labeled target-domain instances recover a large part of the gap on the others.

5 Related Work

Knowledge Conflict in RAG. Prior benchmarks formalize the CK-PK relation as supportive, contradictory, or irrelevant (Mallen et al., 2023; Xie et al., 2024; Wu et al., 2024; Bi et al., 2025; Su et al., 2024b; Longpre et al., 2021; Hou et al., 2024; Xu et al., 2024). This data-level view helps reveal PK-CK mismatch, but overlooks each model's actual answers and their correctness. Along this line, some works infer conflicts from token likelihoods or distributional uncertainty (Wu et al., 2024), some steer generation through decoding interventions or verbal source judgments (Shi et al., 2024; Huang et al., 2025; Wang et al., 2024), and others revise retrieved evidence or adaptively decide when retrieval is needed (Yu et al., 2024; Zhang et al., 2025; Gao et al., 2026; Asai et al., 2024; Jeong et al., 2024; Su et al., 2024a; Yao et al., 2025). These methods improve source coordination but may still force unsupported answers without correctness estimates. We instead label PK- and CK-path correctness against ground truth and abstain when neither path is reliable.

Self-Awareness in LLMs. A growing body of work studies whether LLMs can recognize the limits of their own knowledge and reasoning. On the knowledge side, LLMs expose partial self-knowledge through calibrated confidence, verbalized uncertainty, answerability estimation, and hidden-state probes of latent truthfulness or hallucination risk (Kadavath et al., 2022; Lin et al., 2022; Chen et al., 2026; Burns et al., 2022; Azaria & Mitchell, 2023; Yin et al., 2023; Chen et al., 2024). On the reasoning side, recent work examines whether models can evaluate intermediate reasoning states or anticipate response-level failures from internal representations (Ghasemabadi & Niu, 2026). These studies typically extract a single self-awareness signal for post-hoc diagnosis, such as calibration, hallucination detection, or selective prediction on the model's own answer. SABER instead combines a knowledge-boundary self-prior with reasoning-reliability self-evaluation into paired PK- and CK-side beliefs, and uses them at inference time to choose PK, choose CK, or abstain when both paths fail (Geifman & El-Yaniv, 2017; Zhang et al., 2024; Wen et al., 2025; Joren et al., 2025).

6 Conclusion

We presented SABER, a Self-Aware Belief Estimator that resolves RAG knowledge conflicts by estimating PK- and CK-side correctness beliefs from frozen LLM hidden states. We also constructed a model-specific benchmark labeling PK-only and CK-conditioned answer correctness across four Llama and Qwen backbones. SABER improves accuracy and faithfulness over strong baselines, while its reliability-based abstention signal Pareto-dominates prompt-based abstainers on every backbone. Ablations show that the self-prior and source-conditional reasoning representations provide complementary signals, with gains robust across mid-to-late layers. These results suggest that self-awareness can support faithful RAG by identifying unsupported cases and abstaining.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code implementation, literature review, and editing of the manuscript text. All research ideas, method design, experimental analysis, and final claims are the authors' own, and all AI-assisted content was reviewed and verified by the authors.

REPRODUCIBILITY STATEMENT

We document the benchmark construction protocol, prompts, and labeling procedure in Section 2 and Appendix A, the probe architecture, training hyperparameters, and prompts of SABER in Appendix B, and the metric definitions in Appendix D. Our code and benchmark construction scripts are released at the repository linked in the abstract.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. *arXiv preprint arXiv:2304.13734*, 2023.

Baolong Bi, Shenghua Huang, Yiwei Wang, Tianchi Yang, Zhongyu Zhang, Haizhou Huang, Lijie Mei, Junfeng Fang, Zehao Li, Furu Wei, et al. Context-DPO: Aligning language models for context-faithfulness. *In Findings of the Association for Computational Linguistics: ACL*, 2025.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *International Conference on Learning Representations*, 2024.

Lihu Chen, Gerard de Melo, Fabian M. Suchanek, and Gaël Varoquaux. Query-level uncertainty in large language models. In *International Conference on Learning Representations*, 2026.

Linfeng Gao, Qinggang Zhang, Baolong Bi, Bo Zeng, Zheng Yuan, Zerui Chen, Zhimin Wei, Shenghua Liu, Linlong Xu, Longyue Wang, Weihua Luo, and Jinsong Su. Beyond black-box interventions: Latent probing for faithful retrieval-augmented generation. *In Findings of the Association for Computational Linguistics: ACL*, 2026.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, 2017.

Amirhosein Ghasemabadi and Di Niu. Can llms predict their own failures? self-awareness via internal circuits. *arXiv preprint arXiv:2512.20578*, 2026.

Minghao Guo, Qingcheng Zeng, Xujiang Zhao, Yanchi Liu, Wenchao Yu, Mengnan Du, Haifeng Chen, and Wei Cheng. DeepSieve: Information sieving via LLM-as-a-knowledge-router. In *Findings of the Association for Computational Linguistics: EACL 2026*, 2026.

Yufang Hou, Alessandra Pascale, Javier Carnerero-Cano, Tigran Tchrakian, Radu Marinescu, Elizabeth Daly, Inkit Padhi, and Prasanna Sattigeri. WikiContradict: A benchmark for evaluating LLMs on real-world knowledge conflicts from Wikipedia. In *Advances in Neural Information Processing Systems*, 2024. Datasets and Benchmarks Track.

Yukun Huang, Sanxing Chen, Hongyi Cai, and Bhuwan Dhingra. To trust or not to trust? enhancing large language models' situated faithfulness to external contexts. In *International Conference on Learning Representations*, 2025.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, 2024.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. In *International Conference on Learning Representations*, 2025.

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.

Ehsan Kamalloo, Nouha Dziri, Charles L. A. Clarke, and Davood Rafiei. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, 2020.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*, 2022.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. *arXiv preprint arXiv:2109.05052*, 2021.

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 *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

Kai Mei, Jiang Guo, Shuaichen Chang, Mingwen Dong, Dongkyu Lee, Xing Niu, and Jiarong Jiang. R-WoM: Retrieval-augmented world model for computer-use agents. *arXiv preprint arXiv:2510.11892*, 2025.

Jingwei Ni, Ekaterina Fadeeva, Tianyi Wu, Mubashara Akhtar, Jiaheng Zhang, Elliott Ash, Markus Leippold, Timothy Baldwin, See-Kiong Ng, Artem Shelmanov, and Mrinmaya Sachan. Reprobe: Efficient test-time scaling of multi-step reasoning by probing internal states of large language models. *arXiv preprint arXiv:2511.06209*, 2026.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Yih. Trusting your evidence: Hallucinate less with context-aware decoding. *arXiv preprint arXiv:2305.14739*, 2024.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. Dragin: Dynamic retrieval augmented generation based on the information needs of large language models. *arXiv preprint arXiv:2403.10081*, 2024a.

Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. ConflictBank: A benchmark for evaluating knowledge conflicts in large language models. *arXiv preprint arXiv:2408.12076*, 2024b.

Cunxiang Wang, Sirui Cheng, Qipeng Guo, Yuanhao Yue, Bowen Ding, Zhikun Xu, Yidong Wang, Xiangkun Hu, Zheng Zhang, and Yue Zhang. Evaluating open-QA evaluation. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023a.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Adacad: Adaptively decoding to balance conflicts between contextual and parametric knowledge. *arXiv preprint arXiv:2409.07394*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023b.

Ziqi Wang, Xi Zhu, Shuhang Lin, Haochen Xue, Minghao Guo, and Yongfeng Zhang. RAGRouter-Bench: A dataset and benchmark for adaptive RAG routing. *arXiv preprint arXiv:2602.00296*, 2026.

Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang. Know your limits: A survey of abstention in large language models. *arXiv preprint arXiv:2407.18418*, 2025.

Kevin Wu, Eric Wu, and James Zou. Clasheval: Quantifying the tug-of-war between an llm's internal prior and external evidence. In *Advances in Neural Information Processing Systems*, 2024. Datasets and Benchmarks Track.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *International Conference on Learning Representations*, 2024.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. Knowledge conflicts for LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. SeaKR: Self-aware knowledge retrieval for adaptive retrieval augmented generation. *arXiv preprint arXiv:2406.19215*, 2025.

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 the Association for Computational Linguistics: ACL 2023*, 2023.

Tian Yu, Shaolei Zhang, and Yang Feng. Truth-aware context selection: Mitigating hallucinations of large language models being misled by untruthful contexts. *arXiv preprint arXiv:2403.07556*, 2024.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R. 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*, 2024.

Qinggang Zhang, Zhishang Xiang, Yilin Xiao, Le Wang, Junhui Li, Xinrun Wang, and Jinsong Su. Faithfulrag: Fact-level conflict modeling for context-faithful retrieval-augmented generation. *arXiv preprint arXiv:2506.08938*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

CONTENTS

1 Introduction **1**

2 A Knowledge Conflict Benchmark for LLM Self-Awareness **3**

3 Method: SABER **3**
- 3.1 Self-Prior Representation . . . 3
- 3.2 Conditional Reasoning Representation . . . 4
- 3.3 Belief Estimation and Decision with Abstention . . . 4

4 Experiments **5**
- 4.1 Main Results . . . 5
- 4.2 Selective Answering with Abstention . . . 6
- 4.3 Additional Analysis and Discussions . . . 7

5 Related Work **9**

6 Conclusion **9**

A Benchmark Construction **14**
- A.1 Source datasets . . . 14
- A.2 Benchmark statistics and processing . . . 14
- A.3 Benchmark construction prompts . . . 16

B Implementation Details **16**
- B.1 Architecture and training . . . 16
- B.2 Multi-trace generation . . . 16
- B.3 Self-prior prompt . . . 17
- B.4 PK-side reasoning prompt . . . 17
- B.5 CK-side reasoning prompt . . . 18
- B.6 Self-evaluation prompt . . . 18

C Baselines **18**
- C.1 Baseline descriptions . . . 18
- C.2 Baseline prompts and abstention modifications . . . 19

D Evaluation Metrics **19**

E Additional Results on the Main Experiments **21**
- E.1 Per-sub-task breakdown of main results . . . 21
- E.2 Threshold sweep on the abstention experiment . . . 21
- E.3 Controls for test-time compute and training-data access . . . 21
- E.4 Layer-grid heatmap on the Qwen family . . . 23

F Generalization and Data Requirements **24**
- F.1 Cross-dataset (leave-one-dataset-out) generalization . . . 24
- F.2 Data efficiency . . . 26

G Computational Cost **27**

H Limitations **28**

I Broader Impacts **29**

Appendix

This appendix is organized as follows. Appendix A documents the construction of the benchmark, Appendix B the implementation details including the prompts of SABER, Appendix C the baselines, and Appendix D the metric definitions. Appendix E supplements the main experiments, Appendix F studies cross-dataset generalization and data requirements, and Appendix G reports computational cost, followed by limitations and broader impacts.

A Benchmark Construction

A.1 Source datasets

Our benchmark draws from five publicly released QA datasets, each exposing a controlled form of PK–CK conflict.

- **ConFiQA** (Bi et al., 2025): A QA benchmark where, in the conflicting variant, the retrieved context is modified by counterfactual substitution to assert an answer that contradicts the ground truth. We use all three sub-tasks (QA, MR, MC), which differ in question structure but share the same conflict-construction pipeline.
- **ConflictQA-PopQA** (Xie et al., 2024): Built from PopQA queries on which the LLM has previously given a wrong answer; the conflicting passage is constructed to support the model's incorrect prediction. This directly probes the model's tendency to trust a misleading document over its own (correct) parametric knowledge.
- **ConflictBank** (Su et al., 2024b): A synthesised benchmark of factual conflicts covering multiple categories such as entity substitution, temporal mismatch, and attribute alteration. We use a pilot subset that spans the main categories.
- **TriviaQA**: We use the adversarial-context subset prepared by Huang et al. (2025), in which the conflicting context is obtained by perturbing the original supporting passage to assert a wrong answer, applying the same construction style as ConflictQA-PopQA. It tests robustness to misleading context on questions about widely-known entities.
- **NQ**: We use the adversarial-context subset of Natural Questions prepared by Huang et al. (2025), applying the same construction as the TriviaQA subset above on naturally-occurring user queries. It complements TriviaQA by covering a different query distribution.

A.2 Benchmark statistics and processing

Table 4: Constructed benchmark sizes per backbone. The same partition is used for every backbone.

Sub-dataset	N_{train}	N_{val}	N_{test}	**Total**
ConFiQA-QA	9,600	1,200	1,200	12,000
ConFiQA-MR	9,600	1,200	1,200	12,000
ConFiQA-MC	9,598	1,202	1,200	12,000
ConflictQA-PopQA	12,834	1,608	1,612	16,054
ConflictBank	9,611	1,197	1,192	12,000
TriviaQA	1,920	240	240	2,400
NQ	1,934	242	242	2,418
Total per backbone	**55,097**	**6,889**	**6,886**	**68,872**

Selection and sampling. We use the publicly released instances in full for ConFiQA, ConflictQA-PopQA, TriviaQA, and NQ. For ConflictBank, which is substantially larger than the other datasets, we sample a subset of approximately 12,000 instances that covers its misinformation, temporal, and semantic conflict categories. Every source pairs each question with a context consistent with the gold answer and one or more conflicting contexts, and we keep all of them. The seven datasets are then pooled into a single ground-truth-aligned benchmark that is constructed independently for each backbone.

Labeling answer paths. For every query-context instance, we run the backbone under both the PK-only and the CK-conditioned answer paths and label each resulting answer against the gold answer in two stages. The alias-aware match cascade $\mathrm{match}(\cdot,\cdot)$ described next decides whether the answer matches the gold answer or one of its aliases. Every positive match is then verified by an LLM judge (GPT-4o-mini, prompt in Appendix A.3) that decides whether the candidate expresses the same fact as any gold answer, and the side label y_s is 1 only if both stages accept the answer. The two stages are complementary. Lexical matching alone mislabels a sizeable fraction of open-domain QA answers, whereas LLM judges agree substantially better with human judgment (Bulian et al., 2022; Kamalloo et al., 2023; Wang et al., 2023a; Zheng et al., 2023). Restricting the judge to alias-positive answers keeps every label anchored to the gold string, since the judge can only reject a match and never create one, and it removes the false positives of containment matching, such as a response that lists several entities or embeds the gold string in a contradictory statement. Depending on the backbone, the judge overturns 11% to 17% of the alias-positive PK answers and 7% to 10% of the alias-positive CK answers. The resulting pair $(y_{\mathrm{PK}}, y_{\mathrm{CK}})$ is the 4-cell reliability outcome of the instance, indicating which of the two paths produced a correct answer.

Alias-aware match cascade. The match function compares a generated answer string a against the gold answer, or against any of its accepted aliases when the dataset provides a list of equivalents. It first applies Unicode normalization, casefolding, and punctuation stripping, then tries an exact string match, and finally falls back to a token-level containment check. The cascade returns 1 if any of these checks succeeds for a_s and 0 otherwise. This alias-aware, containment-based matching follows common practice in the literature, including the original papers of several of our baselines. Mallen et al. (2023) count a prediction as correct if any substring of it exactly matches a gold answer, and R-Tuning (Zhang et al., 2024) checks whether the output contains the ground-truth answer instead of requiring an exact match. The evaluation of Huang et al. (2025), from which five of our baselines are drawn, likewise applies relaxed exact matching over synonym-augmented gold answers on TriviaQA and PopQA, and ConFiQA (Bi et al., 2025) scores responses by matching the answer or any of its aliases. We apply the same matching function to SABER and to every baseline.

Scoring and cell membership in evaluation. All reported metrics score the answer of every method, including SABER and the trained baselines, with the alias-aware match cascade alone, without the judge. This keeps the scoring identical for all methods and backbones, follows the practice of the datasets and baselines cited above, and keeps the comparison free of judge variance. Cell membership in the evaluation, which determines CF, KF, and the abstention gold set, uses the judge-verified benchmark labels. Appendix D states the consequences of this combination.

Splits. We split each dataset by question into a 80% / 10% / 10% train / validation / test partition, keeping all contexts of a question in the same partition, and aggregate the per-dataset partitions into the final benchmark whose statistics are reported in Table 4.

Figure 6 and Table 5 report the resulting 4-cell distribution per (backbone, dataset). The cell composition varies substantially across datasets and backbones, which gives our predictors a diverse mixture of all four reliability outcomes during training.

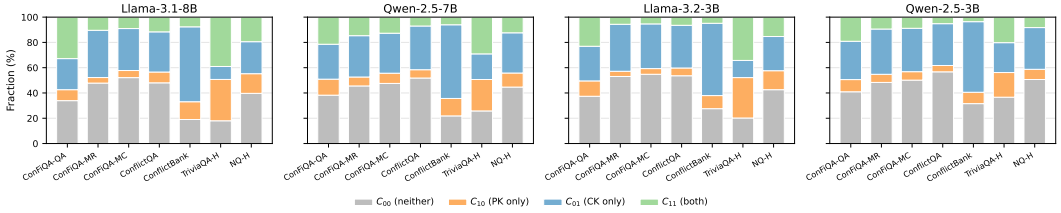

Figure 6: 4-cell reliability distribution of the constructed benchmark. Each stacked bar shows, for one (backbone, dataset) pair, the fraction of instances falling into the four cells C_{00} (neither side correct), C_{10} (only PK correct), C_{01} (only CK correct), and C_{11} (both correct).

Table 5: Per-(backbone, dataset) 4-cell distribution as percentages of dataset size, complementing Figure 6. Rows within each backbone block sum to approximately 100%.

	Llama-3.1-8B				**Qwen-2.5-7B**				**Llama-3.2-3B**				**Qwen-2.5-3B**			
Sub-dataset	C_{00}	C_{10}	C_{01}	C_{11}	C_{00}	C_{10}	C_{01}	C_{11}	C_{00}	C_{10}	C_{01}	C_{11}	C_{00}	C_{10}	C_{01}	C_{11}
ConFiQA-QA	33.9	8.7	24.6	32.8	38.3	12.7	27.6	21.5	37.4	12.1	27.4	23.0	40.9	9.6	30.3	19.1
ConFiQA-MR	47.8	4.4	37.4	10.4	45.6	7.1	32.7	14.7	53.0	4.1	37.2	5.7	48.4	6.4	35.8	9.4
ConFiQA-MC	52.1	5.6	33.2	9.0	47.6	8.1	31.6	12.7	54.8	4.5	35.4	5.4	50.2	6.7	34.3	8.8
ConflictQA-PopQA	48.0	8.6	31.8	11.7	51.8	6.6	34.6	7.0	53.6	6.2	33.8	6.5	56.7	5.0	33.1	5.2
ConflictBank	19.0	14.1	59.3	7.7	21.9	13.8	58.3	6.0	27.6	10.4	57.2	4.9	31.6	9.0	55.8	3.6
TriviaQA	18.0	32.7	10.3	39.0	25.8	24.9	20.2	29.1	20.2	32.0	13.6	34.3	36.7	19.5	23.6	20.2
NQ	39.7	15.5	25.3	19.4	44.7	11.1	31.8	12.4	42.6	15.0	27.0	15.3	50.7	8.1	33.0	8.3

A.3 Benchmark construction prompts

The PK-only and CK-conditioned answer paths used to label the benchmark (Eq. 1) are produced with the two prompts below. The judge prompt that verifies every positive match is shown last. The two answer prompts share the same system instruction and differ only in whether the retrieved context is included before the question.

PK answer prompt (closed-book, benchmark construction)

[System] Answer the question with a short phrase.
[User] Question: {question}
Answer:

CK answer prompt (with retrieved context, benchmark construction)

[System] Answer the question with a short phrase.
[User] Context: {ck_text}
Question: {question}
Answer:

Label verification prompt (LLM judge, benchmark construction)

You are an answer-equivalence judge. Decide whether the CANDIDATE answer expresses the same fact as ANY of the GOLD answers.
Treat as equivalent: paraphrases, alias names, abbreviations/acronyms, partial matches that uniquely identify the entity, language variations, formatting differences. Treat as NOT equivalent: different entities, off-by-one numbers, contradictory facts.
GOLD answers (any one match suffices): {gold_list}
CANDIDATE answer: {candidate}
Reply with exactly one word: YES or NO.

B Implementation Details

B.1 Architecture and training

The two side-specific reliability predictors $f_{\text{pred,PK}}$ and $f_{\text{pred,CK}}$ share the same architecture and training recipe, summarized in Table 6. The LLM backbone is kept frozen throughout. The prior and conditional layers of each backbone are chosen on the validation partitions, and the decision threshold is fixed at $\tau{=}0.5$ rather than tuned, so the test partitions are used only for the results reported in the paper.

B.2 Multi-trace generation

We sample $K{=}3$ reasoning traces per side using the parameters in Table 7. The same parameters are applied to both PK-side and CK-side trace generation across all four backbones.

Table 6: Probe architecture and training hyperparameters for the two reliability predictors $f_{\text{pred,PK}}$ and $f_{\text{pred,CK}}$, which share the same architecture and training recipe.

Component	Setting
Hidden sizes	$(256, 128)$
Activation	ReLU
Dropout	0.2
Input normalization	Per-feature standardization (fit on train)
Optimizer	AdamW
Learning rate	1×10^{-3}
Weight decay	1×10^{-4}
Batch size	256
Loss	Binary cross-entropy with logits
Max epochs	50
Early stopping	On the validation split, patience 8
Seed	42

Table 7: Sampling parameters for multi-trace reasoning generation. Each trace is sampled independently with a different seed, and the same parameters are used on both PK and CK sides across all four backbones.

Parameter	Value
Number of traces K	3
Decoding	Stochastic sampling
Temperature	0.9
Top-p	1.0
Repetition penalty	1.05
Max new tokens	256
Per-trace seeds	$\{0, 1, 2\}$

B.3 SELF-PRIOR PROMPT

The self-prior representation h_q in Eq. 3 is read from a forward pass on the raw query string alone, with no system message and no instruction template, so that the hidden state reflects the model's pre-context belief without being biased by surface phrasing.

Self-prior prompt

{question}

B.4 PK-SIDE REASONING PROMPT

PK reasoning generation prompt

You will be given a question. Assume your own knowledge is correct and complete for this question. Reason step by step using ONLY your prior knowledge — do NOT use any external sources. End with one final answer on the last line.

Question: {question}

Reasoning:

B.5 CK-SIDE REASONING PROMPT

CK reasoning generation prompt

You will be given a question and a document. Assume the document is reliable and contains the correct facts needed to answer the question. Reason step by step using ONLY the information in the document. End with one final answer on the last line.

Question: {question}
Document: {ck_text}
Reasoning:

B.6 SELF-EVALUATION PROMPT

Self-evaluation prompt

You are evaluating a proposed reasoning for a question. Use your own knowledge to judge whether the reasoning is factually reliable.

Question: {question}

Proposed reasoning: {reasoning}

Is the reasoning above factually reliable?
(A) Yes
(B) No

Answer:

C BASELINES

C.1 BASELINE DESCRIPTIONS

We compare against ten baselines that span four design philosophies for handling PK–CK conflict: vanilla retrieve-or-not, verbal self-evaluation, probability- or probe-based heuristics, and fine-tuning.

- **Closed-Book**: The LLM answers from its parametric memory alone, with no retrieved context. We include it as a non-RAG reference point that characterizes the intrinsic knowledge of each backbone before any document is supplied.
- **Vanilla RAG** (Lewis et al., 2020): The standard retrieve-then-read setup, where the LLM answers conditioned on a top-retrieved passage without any further filtering or confidence signal. As the most widely deployed RAG variant, it serves as the natural reference point for any method that adds reliability awareness on top of retrieval.
- **Internal-Eval** (Huang et al., 2025): After producing an internal (PK) answer, the LLM is prompted to judge whether that answer is correct, and the routing decision uses this self-judgment to choose between the PK answer and the retrieved context. It represents a self-confidence-based approach grounded in the model's own assessment of its parametric knowledge.
- **Context-Eval** (Huang et al., 2025): A counterpart to Internal-Eval on the context side: the LLM is asked whether the retrieved passage is reliable enough to be trusted before producing the answer. It isolates the contextual-reliability axis of the routing decision.
- **Implicit-SCR** (Huang et al., 2025): Instructs the LLM to implicitly assess the reliability of the retrieved document and combine this assessment with its own knowledge in a single generation pass, following the self-guided confidence reasoning setup of Huang et al. (2025). We use the original instruction with 3-shot demonstrations and a single return sequence, matching the reference implementation. It is among the strongest prompt-only baselines reported in prior work on knowledge conflict.
- **Explicit-SCR** (Huang et al., 2025): Prompts the LLM to directly compare its PK answer against the CK answer and pick the more reliable one in a single call, the most direct prompting strategy for source arbitration without any auxiliary model or training.

- **TPC** (Wu et al., 2024): Follows the answer path whose token-level probability is higher under the LLM's distribution, treating output likelihood as a confidence signal. It represents a probability-based heuristic that operates at decoding time without any extra prompting or training.
- **TACS-P** (Yu et al., 2024): A prompt-based two-step adaptation of truth-aware context selection. The LLM is first instructed to remove unreliable content from the retrieved document; it then answers based on the filtered document. We use the suffix "-P" to indicate that our adaptation performs the filtering through prompting, whereas the original TACS trains a truthfulness classifier on internal representations to filter context at the token level. This baseline tests whether context filtering alone can cover the routing benefits we aim for.
- **CR-DPO** (Huang et al., 2025): Fine-tunes the LLM backbone using Direct Preference Optimization on conflict-resolution preference pairs, extending Explicit-SCR with parametric updates. It serves as a fine-tuning-based comparison point that bounds the gain achievable when the backbone itself is allowed to be retrained.
- **R-Tuning** (Zhang et al., 2024): Fine-tunes the LLM backbone on an instruction-tuning dataset that explicitly teaches it to respond *I don't know* on questions whose answers it does not reliably know, rather than confabulating one. Although introduced in a non-RAG setting, R-Tuning is a natural training-based comparison point for SABER's reliability-based abstention: both aim at honest refusal when the model lacks reliable knowledge, but R-Tuning achieves it through parametric updates while SABER does so through a frozen-backbone hidden-state probe.

C.2 Baseline prompts and abstention modifications

We adopt the original baseline prompts from their respective sources (Huang et al., 2025; Wu et al., 2024; Yu et al., 2024) and apply only minimal edits to align their input format with our (question, document, gold answer) schema. The modifications described in this subsection are used *only* for the abstention experiment (§4.2); the main-results comparison (§4.1) uses each baseline's original prompt without any abstention suffix. For the abstention experiment, each prompt-only baseline is augmented with a single trailing instruction that allows the model to refuse to answer; the rest of the original prompt, including any in-context examples, is kept verbatim. For instance, the Implicit-SCR system instruction is modified as shown below.

Implicit-SCR system instruction (original (Huang et al., 2025))

You will be given a question and a document. The document may not be trustworthy. Use your judgment to assess the reliability of the document. Then, based on both your assessment and your own knowledge, provide the best possible answer.

Implicit-SCR system instruction (our abstention variant)

You will be given a question and a document. The document may not be trustworthy. Use your judgment to assess the reliability of the document. Then, based on both your assessment and your own knowledge, provide the best possible answer. **If you cannot answer the question accurately based on either your own knowledge or the provided document, respond exactly with: I don't know.**

The same single-sentence abstention suffix is appended to the system instructions of all other prompt-only baselines.

D Evaluation Metrics

All metrics are computed on the same test set of N instances. Each instance belongs to one of four reliability cells $C_{00}, C_{01}, C_{10}, C_{11}$ according to its judge-verified benchmark labels $(y_{\mathrm{PK}}, y_{\mathrm{CK}})$ of Appendix A, where C_{ab} collects the instances with $y_{\mathrm{PK}} = a$ and $y_{\mathrm{CK}} = b$.

Main-results metrics (§4.1; Tables 1, 2, 8, 9). For each instance i, let $c_i \in \{0, 1\}$ indicate whether the method's answer alias-matches the gold. The indicator c_i uses the alias cascade alone, whereas cell membership is judge-verified. Closed-Book and Vanilla RAG return a_{PK} and a_{CK} themselves,

so their KF and CF respectively equal 100 by construction, and the accuracy of a method is not bounded by $1 - |C_{00}|/N$. The four main-results metrics describe a method along two axes: how often it produces the right answer overall, and how it handles the two single-source cells where the right answer is reachable from only one of {PK, CK}.

- **Acc (Answer accuracy).** The fraction of instances on which the method's answer alias-matches the gold. This is the bottom-line correctness rate; it does not distinguish how the method arrived at the answer or which cell the instance came from.

$$\mathrm{Acc} \;=\; \frac{1}{N}\sum_{i=1}^{N} c_i. \tag{8}$$

- **CF (Context Faithfulness).** Accuracy restricted to the cell C_{01}, where only the context carries the correct answer. CF asks: when the model would be wrong from its parametric knowledge alone but the retrieved context contains the right answer, does the method actually follow the context? A high CF indicates that the method effectively absorbs information from a trustworthy context.

$$\mathrm{CF} \;=\; \frac{1}{|C_{01}|}\sum_{i\in C_{01}} c_i. \tag{9}$$

- **KF (Knowledge Faithfulness).** The mirror of CF, restricted to C_{10}, where only parametric knowledge is correct and the retrieved context is misleading. KF asks: when the context would lead the model astray, does the method preserve its internally-correct PK answer? A high KF indicates that the method is robust to a misleading context.

$$\mathrm{KF} \;=\; \frac{1}{|C_{10}|}\sum_{i\in C_{10}} c_i. \tag{10}$$

- **MFS (Macro Faithfulness Score).** The unweighted mean of CF and KF. MFS balances the two failure modes, over-trusting context and over-trusting memory, into a single number; a method that inflates Acc by collapsing onto one source will register near-zero on either CF or KF and so drag MFS down. We treat MFS as the primary single-number measure of conflict-resolution quality.

$$\mathrm{MFS} \;=\; \tfrac{1}{2}\big(\mathrm{CF} + \mathrm{KF}\big). \tag{11}$$

Selective-answering metrics (§4.2; Table 3). Each instance falls into exactly one of three outcomes – answered correctly, answered wrongly, or abstained – with counts N_+, N_-, N_0 satisfying $N_+ + N_- + N_0 = N$. Let $\mathcal{A}$ denote the set of instances on which the method abstained, and let $G = C_{00}$ denote the gold-abstain set (instances on which neither PK nor CK is correct).

- **Score.** The mean per-instance utility under the selective-prediction scheme of Geifman & El-Yaniv (2017): a correct answer is worth $+1$, a wrong answer -1, and an abstention 0. Equivalently, Score is the gap between the number of correct and wrong answers normalized by the test-set size. Concretely, define the per-instance utility

$$u_i \;=\; \begin{cases} +1 & \text{if instance } i \text{ is answered correctly,} \\ -1 & \text{if instance } i \text{ is answered wrongly,} \\ \;\;0 & \text{if instance } i \text{ is abstained.} \end{cases} \tag{12}$$

  The aggregate Score is then

$$\mathrm{Score} \;=\; \frac{1}{N}\sum_{i=1}^{N} u_i \;=\; \frac{N_+ - N_-}{N}. \tag{13}$$

  Score rewards a method only when correct answers outweigh wrong ones; because abstaining contributes zero, a method that refuses everywhere collapses to $\mathrm{Score} = 0$ rather than going negative, which prevents trivial gaming by silence.

- **Cov (Coverage).** The fraction of instances on which the method commits to an answer. Coverage measures *willingness to answer*; we report it as a diagnostic axis rather than a goal, since it can be raised arbitrarily by simply refusing to abstain.

$$\text{Cov} = \frac{N_+ + N_-}{N}. \tag{14}$$

- R_C **(Risk on answered).** The error rate among the answered instances, ignoring abstentions. R_C measures the *conditional quality* of the answers a method does commit to; reporting it together with Cov exposes the trade-off between answering more and answering reliably.

$$R_C = \frac{N_-}{N_+ + N_-}. \tag{15}$$

- F_1 **(Abstention F1).** The F1 of the abstain decisions against the gold-abstain set $G = C_{00}$, treating abstain as the positive class. F_1 measures how *targeted* the refusals are: a method that abstains on everything reaches near-perfect recall but very low precision, while a method that never abstains scores zero on both, so F_1 penalises both blanket caution and reckless answering.

$$F_1 = \frac{2\,|\mathcal{A} \cap G|}{|\mathcal{A}| + |G|}. \tag{16}$$

Together, Score, Cov, R_C, and F_1 span three complementary axes, including net utility (Score), the answer-versus-refuse trade-off (Cov, R_C), and abstain targeting (F_1), that resist gaming by any single failure mode.

E Additional Results on the Main Experiments

E.1 Per-sub-task breakdown of main results

Tables 1 and 2 in the main paper report ConFiQA as a single column computed on the union of its three sub-tasks (QA, MR, MC). Tables 8 and 9 below give the per-sub-task breakdown for those three sub-tasks; numbers on the other four datasets (ConflictQA, ConflictBank, TriviaQA, NQ) are unchanged and can be read directly from the main-paper tables.

E.2 Threshold sweep on the abstention experiment

Table 10 reports the full threshold sweep used to produce Figure 3. For each backbone, we sweep τ over $[0.05, 0.95]$ in steps of 0.05 and report Score, Coverage (Cov), Risk on answered (R_C), and abstention F1 (F_1), all in percent.

E.3 Controls for test-time compute and training-data access

SABER generates multiple reasoning traces and trains predictors on labeled instances, giving it two potential advantages over prompt-based baselines beyond its core mechanism. We isolate these factors with two controls. *Self-consistency* samples six CK-conditioned traces per query using SABER's sampling parameters and generation budget, then returns the majority answer. *Few-shot prompting* adds $k \in \{4, 8, 16\}$ labeled training examples to Vanilla RAG, reporting the best k for each backbone–dataset pair. Both controls use the main experiment's backbones, test splits, and metrics. Table 11 reports Acc and MFS averaged over seven datasets.

Neither control matches SABER. SABER substantially exceeds self-consistency in Acc and MFS on every backbone, as majority voting retains the traces' tendency to follow context. Few-shot prompting yields little improvement and can degrade the smaller Llama backbone. These results suggest that SABER's gains come from estimating path-level reliability from hidden states, rather than from additional sampling or labeled examples alone. Appendix F.2 further examines how much labeled data the predictors require.

Table 8: Per-sub-task breakdown of ConFiQA on the Llama family. We report Acc (↑), CF, KF, and MFS (↑), all as percentages, on the three ConFiQA sub-tasks (QA / MR / MC). For Acc and MFS, within each (backbone, sub-task) cell the best value is highlighted with a **blue** background and the second-best with a green background.

Method	ConFiQA-QA				ConFiQA-MR				ConFiQA-MC			
	Acc↑	CF	KF	MFS↑	Acc↑	CF	KF	MFS↑	Acc↑	CF	KF	MFS↑
Llama-3.1-8B												
Closed-Book	51.2	17.2	100.0	58.6	22.3	3.7	100.0	51.8	20.0	5.8	100.0	52.9
Vanilla RAG	65.8	100.0	21.2	60.6	**57.3**	100.0	18.6	59.3	47.8	100.0	10.8	55.4
Internal-Eval	60.3	56.4	79.8	68.1	40.6	54.3	45.8	50.1	35.2	55.6	52.3	53.9
Context-Eval	57.4	44.0	94.9	69.5	32.8	31.3	81.4	56.3	27.6	31.2	75.4	53.3
Implicit-SCR	66.8	83.8	63.6	73.7	52.2	75.9	39.0	57.4	42.1	77.9	18.5	48.2
Explicit-SCR	55.3	56.4	79.8	68.1	33.6	40.2	62.7	51.5	31.2	43.9	80.0	62.0
TPC	66.4	88.3	67.7	78.0	52.9	85.4	47.5	66.4	45.9	86.8	61.5	74.2
TACS-P	61.5	89.7	36.4	63.0	39.8	52.6	37.3	44.9	29.5	45.7	15.4	30.5
R-Tuning	32.4	10.7	41.3	26.0	14.8	8.6	23.9	16.3	14.7	8.6	29.1	18.9
CR-DPO	58.6	51.2	87.9	69.5	46.4	63.7	74.6	69.1	38.6	62.4	63.1	62.8
SABER (Ours)	**68.2**	89.0	74.7	**81.9**	57.0	92.8	66.1	**79.5**	**50.2**	95.4	73.8	**84.6**
Llama-3.2-3B												
Closed-Book	40.5	10.4	100.0	55.2	16.8	4.2	100.0	52.1	14.5	2.8	100.0	51.4
Vanilla RAG	56.0	100.0	10.1	55.0	49.2	100.0	7.5	53.8	45.8	100.0	10.7	55.4
Internal-Eval	51.8	59.5	74.1	66.8	37.2	62.2	43.4	52.8	33.6	59.9	53.6	56.7
Context-Eval	52.6	54.3	88.5	71.4	44.8	86.9	30.2	58.5	41.6	83.2	42.9	63.0
Implicit-SCR	58.0	93.9	28.8	61.4	38.0	62.0	22.6	42.3	32.3	58.3	23.2	40.7
Explicit-SCR	44.3	47.7	79.1	63.4	22.6	30.0	67.9	49.0	17.3	23.8	53.6	38.7
TPC	60.9	93.9	61.9	77.9	47.2	89.3	41.5	65.4	43.9	89.3	58.9	74.1
TACS-P	51.8	86.4	21.6	54.0	36.2	58.2	22.6	40.4	31.6	56.2	17.9	37.0
R-Tuning	4.8	0.3	10.6	5.5	2.2	1.1	4.8	3.0	1.9	1.1	2.5	1.8
CR-DPO	46.8	43.6	89.9	66.8	29.6	40.4	69.8	55.1	27.5	41.5	71.4	56.5
SABER (Ours)	**62.7**	91.0	83.5	**87.2**	**51.3**	95.8	73.6	**84.7**	**48.2**	96.7	75.0	**85.9**

Table 9: Per-sub-task breakdown of ConFiQA on the Qwen family, with the same setup as Table 8.

Method	ConFiQA-QA				ConFiQA-MR				ConFiQA-MC			
	Acc↑	CF	KF	MFS↑	Acc↑	CF	KF	MFS↑	Acc↑	CF	KF	MFS↑
Qwen-2.5-7B												
Closed-Book	40.2	6.0	100.0	53.0	27.2	4.2	100.0	52.1	22.8	2.3	100.0	51.2
Vanilla RAG	54.8	100.0	6.3	53.1	52.8	100.0	10.9	55.4	48.8	100.0	3.3	51.6
Internal-Eval	55.8	67.0	83.6	75.3	50.4	90.5	25.0	57.8	48.5	90.5	37.4	63.9
Context-Eval	57.5	69.9	89.3	79.6	40.2	47.8	82.6	65.2	36.8	45.1	95.6	70.4
Implicit-SCR	**67.0**	93.5	78.6	86.0	**59.0**	88.7	40.2	64.4	52.4	89.5	47.3	68.4
Explicit-SCR	45.9	57.7	45.3	51.5	39.8	56.7	44.6	50.6	36.2	57.4	44.0	50.7
TPC	57.5	93.5	42.1	67.8	51.3	91.6	37.0	64.3	49.6	93.1	53.8	73.5
TACS-P	52.8	79.2	39.6	59.4	45.2	60.2	39.1	49.6	38.2	63.3	23.1	43.2
R-Tuning	44.4	14.0	88.2	51.1	36.3	21.2	69.2	45.2	33.3	19.2	70.0	44.6
CR-DPO	55.8	75.9	67.3	71.6	51.2	86.5	47.8	67.2	47.8	86.7	51.6	69.2
SABER (Ours)	61.8	87.2	87.4	**87.3**	55.7	93.4	89.1	**91.3**	**53.4**	94.1	95.6	**94.9**
Qwen-2.5-3B												
Closed-Book	32.5	5.4	100.0	52.7	22.8	4.0	100.0	52.0	19.2	4.7	100.0	52.3
Vanilla RAG	55.5	100.0	10.8	55.4	52.7	100.0	5.2	52.6	50.0	100.0	11.5	55.8
Internal-Eval	48.3	61.1	72.5	66.8	45.9	79.2	18.8	49.0	43.2	80.1	24.4	52.2
Context-Eval	39.9	27.4	98.0	62.7	28.2	20.4	93.8	57.1	24.2	19.0	98.7	58.8
Implicit-SCR	57.2	91.6	42.2	66.9	53.8	83.6	31.2	57.4	48.7	84.3	23.1	53.7
Explicit-SCR	45.4	77.2	40.2	58.7	37.3	63.2	26.0	44.6	36.7	68.4	41.0	54.7
TPC	54.3	91.3	32.4	61.8	52.2	93.2	34.4	63.8	50.6	95.1	53.8	74.5
TACS-P	50.5	90.2	15.7	53.0	45.9	78.9	20.8	49.9	41.3	76.1	15.4	45.7
R-Tuning	39.3	17.7	79.6	48.7	45.4	38.7	80.0	59.4	40.7	34.9	66.3	50.6
CR-DPO	48.6	73.4	46.1	59.7	49.0	89.7	26.0	57.9	47.4	91.3	30.8	61.1
SABER (Ours)	**59.1**	93.5	74.5	**84.0**	**56.1**	94.8	69.8	**82.3**	**52.7**	93.7	87.2	**90.4**

Table 10: Full τ-sweep for SABER's selective-answering decision rule across four backbones.

	Llama-3.1-8B				Qwen-2.5-7B				Llama-3.2-3B				Qwen-2.5-3B			
τ	Score	Cov	R_C	F_1	Score	Cov	R_C	F_1	Score	Cov	R_C	F_1	Score	Cov	R_C	F_1
0.05	38.4	84.0	27.1	53.4	32.2	85.1	31.1	50.7	29.0	81.6	32.2	54.8	29.4	83.0	32.3	51.3
0.10	40.2	81.2	25.2	58.8	34.8	80.9	28.5	58.2	32.4	76.6	28.8	62.8	32.2	79.3	29.7	58.1
0.15	41.1	79.2	24.0	61.5	37.0	77.6	26.2	63.8	34.4	73.5	26.6	67.0	34.0	76.1	27.7	62.7
0.20	41.7	77.8	23.2	63.3	39.0	74.3	23.7	68.5	35.5	71.0	25.0	69.8	35.6	73.5	25.8	66.2
0.25	42.4	76.0	22.1	65.1	40.3	71.9	22.0	71.2	36.4	68.5	23.4	71.6	36.9	71.2	24.1	69.2
0.30	43.1	74.3	21.0	66.8	41.7	69.6	20.0	73.4	37.2	66.2	21.9	73.3	37.8	69.0	22.6	71.2
0.35	43.6	72.5	19.9	68.6	42.5	67.4	18.5	75.0	37.7	64.4	20.8	74.6	38.7	67.0	21.1	73.1
0.40	44.0	70.5	18.8	70.3	43.0	64.9	16.9	76.3	38.0	62.5	19.6	75.6	39.2	65.2	19.9	74.4
0.45	44.0	68.3	17.8	71.2	43.4	62.3	15.2	77.3	38.6	60.5	18.1	76.4	39.7	62.6	18.3	75.6
0.50	**43.9**	**65.1**	**16.3**	**72.5**	**43.4**	**58.5**	**13.0**	**78.0**	**38.5**	**58.6**	**17.2**	**76.8**	**39.6**	**60.0**	**17.0**	**76.6**
0.55	43.7	62.6	15.1	72.9	42.7	56.2	12.0	77.8	38.0	56.0	16.1	76.9	39.5	56.2	14.9	77.1
0.60	43.3	60.8	14.3	73.1	42.4	54.3	10.9	78.0	38.0	53.9	14.8	77.4	39.3	53.9	13.6	77.8
0.65	42.9	58.9	13.6	73.2	41.9	52.2	9.9	78.0	37.4	51.9	14.0	77.5	38.8	51.6	12.4	78.1
0.70	42.5	57.0	12.7	73.6	41.2	50.3	9.0	77.7	37.2	50.0	12.8	77.8	38.5	49.4	11.0	78.2
0.75	41.9	55.2	12.0	73.5	40.0	48.2	8.4	77.2	36.4	47.9	11.9	77.7	37.6	47.3	10.2	78.1
0.80	41.4	53.0	11.0	73.3	39.3	45.9	7.1	76.8	35.7	45.4	10.6	77.7	36.4	44.4	8.9	77.9
0.85	40.2	50.2	10.0	73.2	38.0	43.3	6.1	76.2	34.6	42.9	9.7	77.3	34.8	41.2	7.7	77.6
0.90	38.3	46.9	9.1	72.5	35.7	40.1	5.5	74.9	33.2	40.0	8.5	76.8	33.4	37.9	6.0	77.2
0.95	34.8	41.2	7.8	71.0	32.3	35.5	4.5	73.0	30.4	35.8	7.5	75.4	29.8	32.4	4.1	75.6

Table 11: Controls for test-time compute and training-data access. Self-consistency uses SABER's six generations per query, while few-shot prompting uses labeled training examples with the best number selected for each cell.

	Self-consistency		Few-shot prompting		SABER	
Backbone	Acc	MFS	Acc	MFS	Acc	MFS
Llama-3.1-8B	52.1	51.5	55.4	52.9	**64.2**	**81.4**
Qwen-2.5-7B	50.6	49.3	53.6	52.5	**60.3**	**88.8**
Llama-3.2-3B	49.9	52.3	38.0	39.4	**58.3**	**83.5**
Qwen-2.5-3B	48.8	50.5	51.4	50.6	**57.3**	**85.5**

E.4 Layer-grid heatmap on the Qwen family

Figure 7 shows the Qwen counterpart of Figure 5 on the two Qwen backbones. Both panels exhibit a broad high-accuracy plateau around the chosen layer pair, with the optimum on Qwen-2.5-7B falling in the mid-to-late depth band as on the Llama family, while Qwen-2.5-3B selects an unusually shallow prior layer (ℓ_p=2) that nevertheless lies inside its own plateau region.

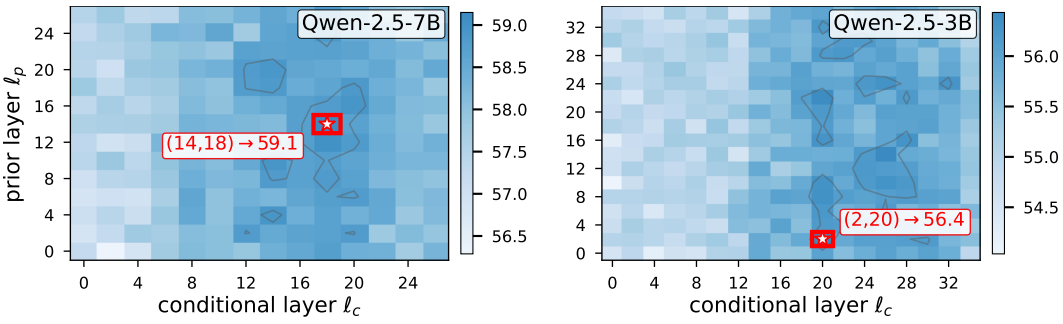

Figure 7: Layer sensitivity of SABER on the Qwen family (counterpart of Figure 5). Each panel shows end-to-end answer accuracy (%) over the (ℓ_p, ℓ_c) grid at stride 2. Darker cells denote higher accuracy and the grey contour traces the 90th-percentile region. The red square marks the selected layer pair.

F Generalization and Data Requirements

F.1 Cross-dataset (leave-one-dataset-out) generalization

The main results train SABER on the combined training partitions of seven datasets and evaluate it on each test split. Here, we assess transfer to a dataset excluded from training. Under a leave-one-dataset-out (LODO) protocol, we retrain SABER on the training, validation, and test instances of the other six datasets, then evaluate it on the held-out test split. Each fold contains roughly 53K–67K training instances, comparable to the main training pool. We report results for every backbone and held-out dataset.

Transfer pattern. Table 12 compares IID and LODO Acc and MFS for each cell, with the same pairs plotted in Figure 8. The datasets differ in how conflicting contexts are constructed. ConFiQA, TriviaQA, and NQ edit real Wikipedia passages to assert incorrect answers, whereas ConflictBank uses templates and ConflictQA uses LLM-generated counter-memory passages. SABER retains most of its IID performance on passage-derived datasets across backbones: Acc changes little in most cells, and most points lie near the diagonal in Figure 8. ConFiQA-QA is an exception on most backbones, as its single-hop questions differ from the multi-hop ConFiQA-MR and ConFiQA-MC questions retained in training. The largest gaps occur on the two synthetic datasets, where MFS drops substantially in every cell and Acc drops most on ConflictBank. Yet IID SABER achieves its highest MFS on these datasets, consistently between 86% and 93%. This pattern points to unseen construction protocols, rather than an inherent performance limit, as the main source of the transfer gap.

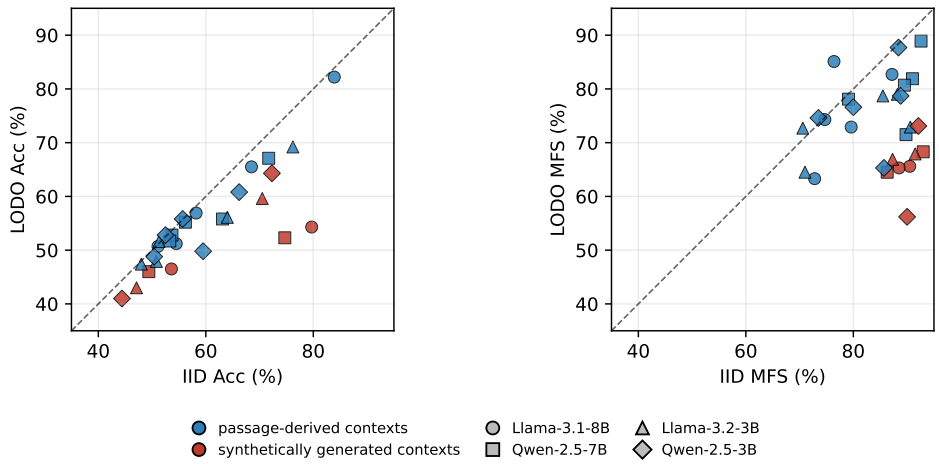

Figure 8: IID versus LODO performance per backbone–held-out dataset cell (Acc, left; MFS, right). Color marks whether the held-out dataset's conflicting contexts are derived from real passages or synthetically generated, and marker shape marks the backbone. The dashed diagonal marks equal IID and LODO performance, so points close to it indicate that SABER keeps its in-distribution performance on a dataset it has never seen.

Few-shot target-domain adaptation. Figure 9 examines how quickly the transfer gap closes after SABER encounters a new construction protocol. We add N labeled instances from the held-out dataset's training partition to the LODO training pool, retrain the predictors, and evaluate on the held-out test split. The benchmark's automatic labeling pipeline eliminates human annotation, while retraining on cached hidden states takes minutes (Appendix G). With just 250 instances, less than 0.5% of the benchmark, MFS rises by over ten points on ConflictBank and also improves on ConflictQA across all backbones. With 2,000 instances, both datasets come within a few points of their in-distribution performance. Thus, the LODO gaps on these datasets reflect the cost of adapting to a new construction protocol rather than a performance ceiling.

Comparison against the strongest zero-shot baseline. We compare LODO SABER with the prompt-based baseline achieving the highest MFS in each (backbone, held-out dataset) cell, using

Table 12: Cross-dataset (LODO) generalization of SABER. For each held-out dataset, SABER is trained on the union of the other six datasets' train, validation, and test instances and evaluated on the held-out test split. IID denotes SABER trained on the training partitions of all seven datasets, and Δ reports the gap from IID to LODO in percentage points. IID numbers come from a separate run under the same protocol as the LODO folds and can differ from the main-results tables.

		IID		LODO		Δ (%)	
Backbone	**Held-out**	Acc	MFS	Acc	MFS	ΔAcc	ΔMFS
Llama-3.1-8B	ConFiQA-QA	68.5	79.6	65.5	72.9	-3.0	-6.6
	ConFiQA-MR	58.2	74.7	56.9	74.3	-1.2	-0.5
	ConFiQA-MC	51.1	87.2	50.7	82.7	-0.3	-4.5
	ConflictQA	53.6	90.5	46.5	65.6	-7.1	-24.9
	ConflictBank	79.7	88.5	54.3	65.3	-25.4	-23.2
	TriviaQA	83.9	76.4	82.2	85.1	-1.7	+8.7
	NQ	54.5	72.8	51.2	63.3	-3.3	-9.5
Qwen-2.5-7B	ConFiQA-QA	63.1	89.8	55.8	71.5	-7.3	-18.3
	ConFiQA-MR	56.2	91.0	55.2	81.9	-1.1	-9.1
	ConFiQA-MC	53.7	92.6	52.8	88.9	-0.9	-3.7
	ConflictQA	49.4	93.0	46.0	68.3	-3.3	-24.7
	ConflictBank	74.7	86.3	52.3	64.5	-22.4	-21.8
	TriviaQA	71.7	89.5	67.1	80.7	-4.6	-8.8
	NQ	53.3	79.1	51.7	78.1	-1.7	-1.0
Llama-3.2-3B	ConFiQA-QA	64.0	90.6	56.1	72.9	-7.9	-17.7
	ConFiQA-MR	51.3	85.5	51.6	78.7	+0.3	-6.8
	ConFiQA-MC	48.0	88.2	47.4	79.0	-0.6	-9.2
	ConflictQA	47.1	91.5	43.0	67.9	-4.2	-23.6
	ConflictBank	70.5	87.3	59.6	66.9	-10.8	-20.4
	TriviaQA	76.2	70.6	69.2	72.7	-7.1	+2.1
	NQ	50.8	71.0	47.9	64.5	-2.9	-6.5
Qwen-2.5-3B	ConFiQA-QA	59.5	85.7	49.8	65.3	-9.7	-20.4
	ConFiQA-MR	55.7	80.0	55.8	76.6	+0.1	-3.4
	ConFiQA-MC	52.5	88.4	52.8	87.7	+0.3	-0.8
	ConflictQA	44.4	92.1	41.0	73.1	-3.4	-19.1
	ConflictBank	72.3	90.0	64.3	56.2	-8.1	-33.9
	TriviaQA	66.2	88.8	60.8	78.7	-5.4	-10.1
	NQ	50.4	73.5	48.8	74.6	-1.7	+1.0

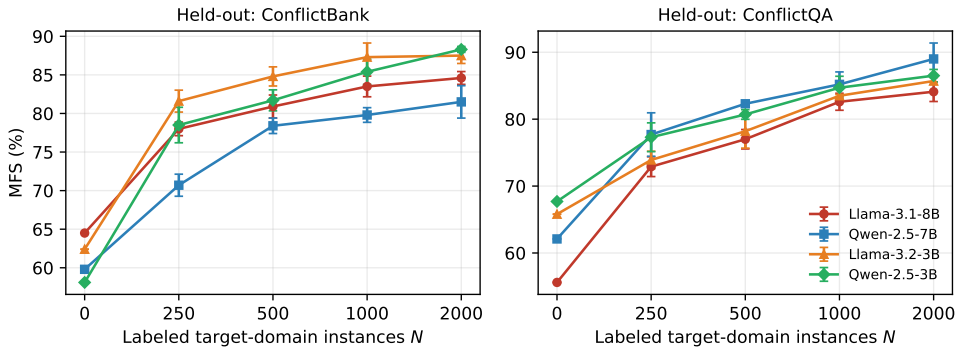

Figure 9: Few-shot target-domain adaptation of LODO SABER on the two synthetically generated datasets. For each held-out dataset, N automatically labeled instances from its training partition are added to the LODO training pool before the predictors are retrained, and the curves show held-out MFS (mean and standard deviation over three seeds). A few hundred instances already recover a large part of the gap. The predictors in this experiment are retrained under a fixed training budget, so the $N{=}0$ point is the zero-shot reference of this protocol and can differ from Table 12.

that baseline's paired (Acc, MFS). This selects one deployable baseline per cell rather than different baselines for the two metrics. Table 13 shows that LODO SABER exceeds this baseline on Acc in 19/28 cells and on MFS in 20/28, often by more than ten MFS points. Some of the largest gains on both metrics occur on ConFiQA-MC and TriviaQA. On the synthetic ConflictQA dataset, SABER leads in Acc on every backbone and in MFS on most. Prompt-based baselines with high MFS, typically Context-Eval and Internal-Eval, often sacrifice Acc, while those with high Acc lag in MFS. SABER performs well on both by estimating path-level reliability. Its weaker cells cluster on ConflictBank and ConFiQA-QA, whose construction patterns are least represented in the six-dataset training pool. As shown above, a few hundred target instances substantially narrow the ConflictBank gap.

Table 13: LODO SABER against the strongest prompt-based zero-shot baseline on each backbone–held-out cell. For each cell, the baseline with the highest MFS among Closed-Book (CB), Vanilla RAG (RAG), Internal-Eval (Int-Eval), Context-Eval (Ctx-Eval), Implicit-SCR (Imp-SCR), and Explicit-SCR (Exp-SCR) is selected, and its paired Acc and MFS are reported. Bold marks SABER wins, and Δ is reported in percentage points. Without seeing the held-out dataset, SABER is still ahead in Acc on 19/28 cells and in MFS on 20/28 cells.

			Top-MFS Baseline		**LODO SABER**		**Δ (%)**	
Backbone	**Held-out**	**Method**	Acc	MFS	Acc	MFS	ΔAcc	ΔMFS
Llama-3.1-8B	ConFiQA-QA	Imp-SCR	66.8	73.7	65.5	72.9	-1.3	-0.8
	ConFiQA-MR	RAG	57.3	59.3	56.9	74.3	-0.4	**+15.0**
	ConFiQA-MC	Exp-SCR	31.2	62.0	50.7	82.7	**+19.6**	**+20.7**
	ConflictQA	Exp-SCR	40.9	63.1	46.5	65.6	**+5.6**	**+2.5**
	ConflictBank	Ctx-Eval	58.9	76.0	54.3	65.3	-4.6	-10.7
	TriviaQA	Exp-SCR	63.3	69.8	82.2	85.1	**+18.9**	**+15.2**
	NQ	Exp-SCR	46.3	63.0	51.2	63.3	**+5.0**	**+0.2**
Qwen-2.5-7B	ConFiQA-QA	Imp-SCR	67.0	86.0	55.8	71.5	-11.2	-14.5
	ConFiQA-MR	Ctx-Eval	40.2	65.2	55.2	81.9	**+15.0**	**+16.7**
	ConFiQA-MC	Ctx-Eval	36.8	70.4	52.8	88.9	**+15.9**	**+18.5**
	ConflictQA	Int-Eval	38.2	67.6	46.0	68.3	**+7.8**	**+0.7**
	ConflictBank	Ctx-Eval	60.7	62.6	52.3	64.5	-8.4	**+1.9**
	TriviaQA	Int-Eval	65.4	78.0	67.1	80.7	**+1.7**	**+2.7**
	NQ	Imp-SCR	52.5	64.5	51.7	78.1	-0.8	**+13.6**
Llama-3.2-3B	ConFiQA-QA	Ctx-Eval	52.6	71.4	56.1	72.9	**+3.5**	**+1.5**
	ConFiQA-MR	Ctx-Eval	44.8	58.5	51.6	78.7	**+6.8**	**+20.2**
	ConFiQA-MC	Ctx-Eval	41.6	63.0	47.4	79.0	**+5.8**	**+15.9**
	ConflictQA	Int-Eval	37.2	70.7	43.0	67.9	**+5.8**	-2.8
	ConflictBank	Ctx-Eval	49.9	71.9	59.6	66.9	**+9.7**	-5.0
	TriviaQA	Int-Eval	72.1	69.9	69.2	72.7	-2.9	**+2.8**
	NQ	Int-Eval	43.8	61.8	47.9	64.5	**+4.1**	**+2.8**
Qwen-2.5-3B	ConFiQA-QA	Imp-SCR	57.2	66.9	49.8	65.3	-7.5	-1.6
	ConFiQA-MR	Imp-SCR	53.8	57.4	55.8	76.6	**+2.0**	**+19.2**
	ConFiQA-MC	Ctx-Eval	24.2	58.8	52.8	87.7	**+28.7**	**+28.8**
	ConflictQA	Int-Eval	30.6	61.4	41.0	73.1	**+10.4**	**+11.7**
	ConflictBank	Ctx-Eval	39.6	59.0	64.3	56.2	**+24.7**	-2.8
	TriviaQA	Int-Eval	63.3	82.6	60.8	78.7	-2.5	-3.8
	NQ	Int-Eval	41.7	71.0	48.8	74.6	**+7.0**	**+3.6**

F.2 Data efficiency

Deploying SABER on a new backbone requires labeling a training set for that backbone, so we ask how much data the final decision needs. Table 14 retrains the predictors on random subsets of increasing size, keeping the architecture, layer pair, decision rule, and early-stopping procedure unchanged. We report Acc, CF, KF, and MFS on the union of the seven test splits. Acc approaches its full-data value with about 2,000 instances and changes little thereafter, whereas MFS continues to improve across the range. The CF and KF components explain this difference. CF is already high with a few hundred instances because deferring to context is correct in the large cell where only the CK path succeeds, and the predictors learn this preference quickly. KF starts much lower and drives

most of the subsequent gains: trusting parametric knowledge despite conflicting context must be learned from the rarest informative cell, which accounts for about 10% of test queries. Acc weights cells by frequency and is therefore dominated by behavior learned early, whereas MFS weights CF and KF equally and keeps rising as KF improves. About 5% the training partition recovers over 90% of full-data MFS on every backbone, and about half brings the gap within a few points.

These results have two implications. For deployment, labels require running the backbone on both answer paths rather than human annotation, so adapting SABER to a new backbone incurs a modest inference cost, and a fraction of the labeled data delivers most of its accuracy and conflict faithfulness. we encourage practitioners who apply SABER to a new source of conflicting contexts to label a few hundred instances from that source before deployment. For the benchmark, continued KF gains show that additional labels still help on the hardest cell. The full benchmark thus supports evaluation and exploration of the predictors' upper performance range, rather than being required to run SABER.

Table 14: Data efficiency of the reliability predictors. Acc, CF, KF, and MFS (%) on the union of the seven test splits when the predictors are trained on a random subset of the training partition (share in parentheses), averaged over three seeds. All other settings follow the main results.

	Llama-3.1-8B				**Qwen-2.5-7B**				**Llama-3.2-3B**				**Qwen-2.5-3B**			
Training instances	Acc	CF	KF	MFS	Acc	CF	KF	MFS	Acc	CF	KF	MFS	Acc	CF	KF	MFS
500 (1%)	58.3	88.9	57.6	73.3	53.3	82.5	69.7	76.1	51.6	85.6	66.2	75.9	51.7	88.0	51.7	69.9
1,000 (2%)	58.1	86.8	63.1	74.9	55.1	86.9	71.2	79.1	53.0	89.4	65.7	77.6	52.3	87.6	61.4	74.5
2,000 (4%)	59.9	90.8	66.4	78.6	56.9	93.0	66.8	79.9	54.0	91.2	68.6	79.9	53.8	92.1	57.4	74.8
5,000 (9%)	60.1	89.8	71.8	80.8	56.9	91.1	73.7	82.4	54.7	92.3	71.6	82.0	54.7	92.6	65.1	78.8
10,000 (18%)	60.7	91.9	69.7	80.8	57.6	92.3	75.9	84.1	55.0	93.9	68.3	81.1	55.0	92.8	68.3	80.6
25,000 (45%)	61.2	91.4	77.1	84.2	58.6	94.2	78.4	86.3	55.6	93.6	74.9	84.3	55.4	93.8	68.8	81.3
Full (≈55K)	62.1	94.5	74.5	84.5	59.0	94.3	82.5	88.4	56.1	93.9	79.5	86.7	56.0	94.2	73.3	83.8

G Computational Cost

We separate SABER's cost into a one-time training-side cost and a per-query inference-side cost. The former is dominated by hidden-state extraction and is comparable to the cheaper of the two fine-tuning baselines, while the latter is a modest, adjustable test-time budget.

Training-side cost. SABER's predictors train roughly an order of magnitude faster than R-Tuning and two orders faster than CR-DPO, using fewer GPUs and less memory. All experiments used NVIDIA A5000 GPUs with 24 GB of memory. Table 15 reports the per-backbone training costs of the three methods. Compared with the two fine-tuning baselines, SABER offers four practical advantages:

- **Smaller predictors and lower memory use.** SABER's MLP predictors have $\sim$5M trainable parameters and use $<$2 GB of peak GPU memory, compared with $\sim$14M LoRA parameters and 5–10 GB of peak memory for R-Tuning and CR-DPO.
- **Single-GPU training.** SABER trains on one A5000 in minutes, whereas our R-Tuning and CR-DPO runs used multi-GPU DDP with 4–8 GPUs. It therefore fits within typical academic compute budgets without specialized infrastructure.
- **Lower training cost.** Across four backbones, SABER's total predictor training cost is $<$0.6 GPU-hours, compared with $\sim$13 GPU-hours for R-Tuning ($\sim$22$\times$ higher) and $\sim$72 GPU-hours for CR-DPO ($\sim$120$\times$ higher). Predictor training and evaluation each take less than 0.2 GPU-hour, even for 8B backbones.
- **Amortized setup and no additional checkpoint.** SABER's main up-front cost is hidden-state extraction, which required approximately 15 GPU-hours of effective compute across four backbones in the production run. Multiple extraction jobs shared each GPU, and inference rarely saturated the device. This cost is paid once per backbone and amortized across probe variants, while the backbone remains frozen and reusable. R-Tuning and CR-DPO incur their full training cost when the backbone, dataset, or hyperparameters change and require a fine-tuned checkpoint alongside the base model.

Including hidden-state extraction, SABER requires approximately 15 GPU-hours of effective compute, about one-fifth of CR-DPO's $\sim$72 GPU-hours and comparable to R-Tuning's $\sim$13 GPU-hours. It also

runs on a single A5000 with substantially lower peak memory use. Once hidden states are cached, each subsequent probe retraining takes minutes rather than hours or even days, widening the compute advantage when multiple configurations are explored. These properties make SABER practical to deploy and iterate on, particularly without large-scale multi-GPU infrastructure. Labeling a training partition for a new backbone requires one PK and one CK generation per instance, plus a judge call for each positive match. Appendix F.2 shows that a fraction of the partition already yields most of the dquality.

Table 15: Per-backbone training cost on NVIDIA A5000 (24 GB). SABER's one-time hidden-state extraction (not shown) costs ∼5 GPU-h per 8B backbone and ∼3 GPU-h per 3B backbone (∼15 GPU-h total of effective compute for the production run, with multiple extraction jobs sharing each GPU), amortized across every variant trained on the cached states.

	CR-DPO	**R-Tuning**	**SABER predictor**
Trainable params	∼14M (LoRA)	∼14M (LoRA)	∼5M (MLP)
Peak GPU memory	∼10 GB	∼5 GB	<2 GB
Hardware (DDP)	8×A5000	4–8×A5000	1×A5000
GPU-h on Llama-3.1-8B	21	5.3	<0.2
GPU-h on Llama-3.2-3B	14	2.3	<0.1
GPU-h on Qwen-2.5-7B	23	2.3	<0.2
GPU-h on Qwen-2.5-3B	14	3.0	<0.1
Total GPU-h	∼72	∼13	<0.6

Inference-side cost. For each query, SABER generates $2K$ reasoning traces, K per side, and performs one query-only forward pass for the self-prior and $2K$ self-evaluation forward passes. These additional passes are prefill-only and require no decoding. At the default $K{=}3$, SABER generates six traces, each averaging 65–116 tokens across our backbones. This amounts to roughly 400–700 generated tokens and seven prefill-only passes per query.

The traces are independent and generated in one batched call, keeping wall-clock latency closer to that of one trace than six sequential traces. The prefill-only passes require no decoding, and the two predictors add negligible cost. The budget is adjustable: K provides a smooth cost–faithfulness trade-off, and $K{=}1$, with two generations per query, retains most of the gain (Section 4.3). At the same six-generation budget, majority voting over the traces falls well behind SABER in both Acc and MFS on every backbone (Appendix E.3). Thus, the gain comes from estimating path-level reliability from hidden states rather than additional sampling. Alongside its modest training-side cost, SABER adds reliability awareness to RAG with an adjustable test-time budget.

H LIMITATIONS

We discuss several limitations of our study, primarily as scope choices that motivate future work.

Backbone scope. Our experiments cover four open-weight instruction-tuned backbones from the Llama family, Llama-3.1-8B and Llama-3.2-3B, and the Qwen family, Qwen-2.5-7B and Qwen-2.5-3B. Extending SABER to substantially larger backbones, where hidden states may exhibit different self-awareness profiles, is left to future work.

Model access. SABER reads hidden states and therefore assumes white-box access to the backbone, so it does not directly apply to models served only through a text interface. This requirement is shared by probing-based methods and is lighter than that of the fine-tuning baselines, R-Tuning and CR-DPO, which must also update and maintain the model weights. SABER keeps the backbone frozen and trains only lightweight predictors on its hidden states, at a small fraction of their training cost (Appendix G). This matches a common deployment setting, in which small open-weight models are chosen to reduce serving cost, and it is also where SABER helps most, since its advantage over CR-DPO and over prompt-based abstention is largest on the 3B backbones (§4.1 and §4.2).

Retrieval scope. Our study assumes that the retrieved context has already been supplied and focuses on the well-defined problem of resolving knowledge conflict between this context and the model's parametric knowledge. We do not address the upstream retrieval process itself, including how candidate passages are selected, ranked, or pre-filtered from a corpus, which is an orthogonal line of work. The contexts in our benchmark come from established conflict-QA datasets rather than from a live retriever, and evaluating SABER on naturally retrieved passages is a natural next step.

Static benchmark. Our benchmark labels four fixed LLMs at a single point in time. As new LLMs are released, per-backbone PK- and CK-path correctness labels need to be re-collected. The labeling pipeline in Eq. 1–2, together with the judge verification of Appendix A, is reproducible, but incurs non-trivial inference cost for each new backbone, although Appendix F.2 shows that a fraction of the labeled instances already yields most of the benefit.

Unseen construction protocols. When the conflicting contexts of a held-out dataset are constructed in a way that is absent from training, a gap to the in-distribution setting remains (Appendix F.1). Even without any target-domain data, SABER stays ahead of the strongest prompt-based baseline in most held-out cells, and a few hundred automatically labeled instances from the new protocol close most of the remaining gap within minutes of predictor retraining. We therefore view this limitation as a small and necessary adaptation cost rather than a limit of the method, and we encourage practitioners who apply SABER to a new source of conflicting contexts to label a few hundred instances from that source before deployment.

I Broader Impacts

SABER aims to make retrieval-augmented generation more honest by enabling reliability-based abstention when neither the parametric nor the contextual answer path is reliable. Its intended positive impact is to reduce confidently wrong outputs in LLM-based question answering. We do not anticipate substantial negative societal impacts specific to our method.