Feeling of Knowing in Large Language Models

Zichuan Fu[1,*], Xian Wu[2,†], Jingtong Gao[1,†], Wenlin Zhang[1], Jiaxuan Li[1], Binhao Wang[1], Yimin Deng[1,3], Guojing Li[1,4], Xiaopeng Li[1], Derong Xu[1,5], Yefeng Zheng[6], Xiangyu Zhao[1,†]
[1]City University of Hong Kong, [2]Tencent Jarvis Lab
[3]Xi'an Jiaotong University, [4]Renmin University of China
[5]University of Science and Technology of China, [6]Westlake University
zc.fu@my.cityu.edu.hk, kevinxwu@tencent.com,
jt.g@my.cityu.edu.hk, xianzhao@cityu.edu.hk

Abstract

Large language models (LLMs) are frequently updated through post-training techniques to maintain up-to-date knowledge in real-world deployment. Their reliability depends not only on what the LLM knows, but also on whether it knows what it knows—which we study as functional *feeling of knowing* (FoK). FoK serves as the underlying signal for selective generation, retrieval triggering, and model routing. Miscalibrated FoK makes systems overconfident on unknown questions or unnecessarily refuse questions they could have answered. As LLMs are updated through post-training, however, we discover that what an LLM knows can change without a corresponding update to its FoK. This creates a knowledge–FoK desynchronization challenge, where FoK estimators calibrated to the pre-trained LLM become unreliable as the model's knowledge state evolves. To address this challenge, we propose a two-channel design: a knowledge channel that changes what the LLM knows and a FoK channel that judges whether the current knowledge state supports answering. We train these channels with a three-stage procedure: on-policy supervised fine-tuning (OP-SFT) updates the knowledge channel from corrected on-policy answers, knowledge-delta FoK training (KΔ-FT) trains the FoK channel over intermediate knowledge states, and FoK-guided policy optimization (FGPO) uses the judgment to generate the corresponding answer or explanation. Experiments on four datasets show that explicit FoK alignment substantially improves judgment quality on updated knowledge. The code is available at `Applied-Machine-Learning-Lab/NeurIPS2026_Know-thyself`.

1 Introduction

Large language models (LLMs) serve as assistants, tool-using and search agents [22, 42, 64, 70], and recommendation and decision-support models [12, 45, 46]. Their reliability depends on answer accuracy and on judging whether to answer or seek external support. Cognitive science calls this metacognitive judgment *feeling of knowing* (FoK) [17, 30]. We use *functional FoK* for this behavior in LLMs: answerability judgments modeled after human FoK. This scalar judgment supports selective prediction, model routing, and retrieval triggering [15, 24, 26, 41, 67]. If it fails to reflect updated knowledge, a system may confidently answer unknown questions or refuse questions the updated model could solve.

Existing LLM FoK methods mainly take three forms. Output-based methods read uncertainty from the model's own generations, such as the probability of a "true" token or the entropy of sampled

[*]Work done during an internship at Tencent Jarvis Lab.
[†]Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

answers [9, 26, 33]. Probe-based methods train a lightweight classifier on hidden states [1, 3]. Elicitation-based methods supervise the model to verbalize its confidence directly [52, 62, 66]. However, these methods are usually calibrated and evaluated under static model parameters. This leaves open what happens when the model is later updated by supervised fine-tuning (SFT), low-rank adaptation (LoRA), knowledge editing, or continual learning [13, 21, 48, 49]. After an update, the LLM's knowledge may change, while a FoK estimator trained or calibrated for the base model may still judge answerability based on the base model's knowledge boundary. **The problem of keeping FoK aligned under changing model knowledge remains underexplored.**

Koriat's account motivates this study: human FoK is inferred from accessible cues rather than directly read from memory [30]. Our correctness-supervised objective does not instantiate or validate that cognitive process. In LLMs, prior causal tracing locates factual associations mainly in middle MLP layers [49], whereas our FoK recovery peaks occur later (Section 2.2). Two challenges follow: 1) **Knowledge–FoK desynchronization:** knowledge training changes answerability without keeping the FoK judgment aligned; and 2) **Context-dependent FoK:** the judgment varies with the available context and generated reasoning. We therefore train a FoK channel to judge the current knowledge state and to guide the subsequent response.

We address these challenges with two trainable channels. On-policy supervised fine-tuning (OP-SFT) updates the knowledge channel from the model's own answers, supervising only corrected answer-relevant suffixes to limit disruption to FoK. Knowledge-delta FoK training (KΔ-FT) trains the FoK channel across knowledge deltas, allowing judgment labels to reflect changing answerability. For context-dependent FoK, FoK-guided policy optimization (FGPO) trains the model to generate context, judge answerability, and then answer or explain. The paper makes three contributions:

- We study functional FoK after knowledge updates, finding later-layer FoK recovery, sensitivity to accessible context, and that knowledge updating alone fails to synchronize answerability judgments.
- We introduce a two-channel framework with OP-SFT, KΔ-FT, and FGPO to update knowledge, synchronize FoK, and train the model to answer or explain according to its FoK judgment.
- We evaluate FoK on factual QA with cross-dataset transfer, three sequential updates, and seed variation for the main method. MMLU and ARC-Easy measure capability preservation.

2 Mechanism of LLM FoK

2.1 Problem formulation

Let M_θ be an LLM at parameter state θ, and let q be a question. We use $Y_\theta(q) \in \{0, 1\}$ for whether the model's greedy answer is correct, and $J_\theta(q) \in [0, 1]$ for the model's functional FoK score, where 0 means the model judges that it cannot answer and 1 means it judges that it can answer. The quality of FoK is reflected in the alignment between $J_\theta(q)$ and $Y_\theta(q)$ under the same decoding rule. For a generative yes/no judgment prompt, we instantiate this score as $J_\theta(q) = \frac{P_\theta(\text{yes}|q)}{P_\theta(\text{yes}|q)+P_\theta(\text{no}|q)}$.

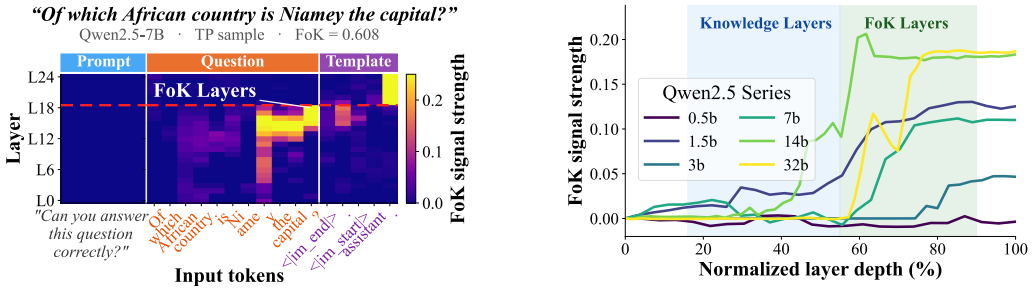
When an LLM learns new knowledge through post-training, its parameters change from the base state θ to an updated state $\tilde{\theta}$. We call the induced difference $\kappa = \tilde{\theta} - \theta$ the *knowledge delta*. The target is to keep $J_{\tilde{\theta}}$ aligned with $Y_{\tilde{\theta}}$ as knowledge changes. Our method trains FoK along the update trajectory.

In evaluation, we focus on two directional failures of the FoK judgment. An *illusion* occurs when the model judges that it can answer but is wrong; an *omission* occurs when it judges that it cannot answer although its answer would be correct. Metrics and splits are detailed in Section 4.1.

2.2 FoK localization in model depth

To localize FoK ability, we first adopt ROME's causal-tracing method [49]. For each factual question, we corrupt the subject token, restore hidden states one layer at a time, and measure whether the yes/no FoK judgment is recovered. We repeat the procedure across Qwen2.5 model sizes, using normalized depth to compare corresponding layers across architectures rather than raw layer indices.

As shown in Figure 1a, Qwen2.5-7B places the strongest FoK signal in late layers, around two-thirds of model depth. Figure 1b shows that the same late-layer pattern remains after normalizing depth across Qwen2.5 scales from 0.5B to 32B. This is later than the factual association signal reported by



(a) Causal tracing of FoK signal in a 7B LLM.

(b) FoK signal across normalized depth.

Figure 1: FoK signal localizes late under causal restoration. Panel (a) shows a 7B example across tokens and layers; panel (b) aggregates FoK recovery across Qwen2.5 sizes over normalized depth.

Table 1: Knowledge and FoK under separate or sequential SFT updates. Base gives absolute values; the three SFT blocks give changes from Base. Accuracy changes are percentage points (pp); AUROC changes are unitless. ΔAUC denotes the change in AUROC. Higher accuracy and AUROC are better. Full illusion and omission rates are reported in Appendix I.1.

	Base		K-SFT		J-SFT		K→J SFT	
Model	Acc.	AUROC	ΔAcc.	ΔAUC	ΔAcc.	ΔAUC	ΔAcc.	ΔAUC
Qwen2.5-0.5B	20%	0.519	+2	-0.016	-14	+0.213	+2	+0.211
Qwen2.5-1.5B	35%	0.529	+6	-0.035	+2	+0.255	+5	+0.248
Qwen2.5-3B	39%	0.739	+12	-0.220	+3	+0.038	+11	+0.051
Qwen2.5-7B	53%	0.724	+7	-0.042	+2	+0.088	+7	+0.100
Qwen2.5-14B	66%	0.762	+6	+0.004	+1	+0.030	+5	+0.068
Qwen2.5-32B	68%	0.819	+4	-0.026	0	+0.010	+4	+0.020
LLaMA-3.1-8B	74%	0.672	+2	-0.038	+1	+0.045	+2	+0.063

ROME, which is localized in middle MLP layers, around one-third of model depth [49]. Appendix Figure 7 gives a parameter-side check: knowledge SFT changes shallow-to-middle layers more, while judgment SFT changes concentrate in the late layers where FoK signal strength is high. Appendix D extends the causal tracing across scales and outcome groups.

Takeaway 1: Under causal tracing, the FoK signal peaks later in model depth than factual associations, suggesting that FoK acts as a judgment over the model's knowledge state rather than as factual recall itself.

2.3 How FoK depends on updated knowledge

Depth separation does not mean independence. To analyze how factual knowledge and FoK relate, we intervene on one side and evaluate the other under matched data and optimization settings. First, we update factual knowledge with vanilla factual SFT and measure FoK on the updated model. Second, we train FoK alone and measure whether answer accuracy changes. Table 1 summarizes cross-scale results. Appendix G adds exact-match (EM) results across scales and families.

Table 1 shows that knowledge SFT (K-SFT) increases answer accuracy for every model, but FoK AUROC decreases for six of the seven models, with the largest drop on Qwen2.5-3B. Thus, the knowledge update changes the model's answerability, but the judgment boundary does not adjust correctly. The mismatch is not a single error direction: after K-SFT, some models make more omissions, judging that they cannot answer questions they would answer correctly, whereas larger Qwen models better preserve AUROC but shift toward illusion, judging that they can answer even when their answers are wrong. The issue is therefore not simply increased conservatism or confidence, but a weakened alignment between the current knowledge state and FoK. Judgment SFT (J-SFT) provides a reference intervention: directly training judgment improves FoK without compromising answer performance in larger models. K→J SFT further shows that judgment training after a

knowledge update can recalibrate FoK while preserving most of the answer-accuracy gain from K-SFT, indicating that FoK depends on the current knowledge state and should be updated after knowledge changes. Appendix E provides the FoK-recovery and reverse-direction results.

Takeaway 2: In current LLMs, knowledge updates change answerability but do not automatically update FoK. FoK depends on features of the current knowledge state, so changing knowledge can break the mapping from knowledge to self-judgment unless FoK is explicitly updated.

2.4 Reasoning before judgment increases FoK certainty

Koriat's FoK theory suggests that the amount and intensity of accessible information increase FoK judgments [30]. We test the LLM analogue by asking whether self-generated reasoning affects FoK.

We evaluate Qwen2.5-7B on MATH [19] problems. For each problem, the model first generates a solution trace. We then re-query the same model at increasing prefixes of that trace and compute $J = P(\text{“Yes”})/[P(\text{“Yes”}) + P(\text{“No”})]$ for the question "Can you solve this problem correctly?" Figure 2 shows that the FoK signal becomes more separated over the generated trace. As more reasoning tokens are available, the mean FoK scores for final "Yes" and "No" judgments move toward $J = 1$ and $J = 0$, respectively; the AUROC of J for the final Yes/No FoK token rises as more reasoning tokens are available. This indicates that FoK is sensitive to the model's current context, not only to the original question.

Takeaway 3: FoK should be judged from the model's current context, not from the original question alone. This motivates a response-generation procedure in which the model first accumulates task-relevant context, then evaluates answerability before producing the final answer.

3 Methodology

Our methodology aligns FoK with updated knowledge through a two-channel model structure and a three-stage procedure that updates knowledge, synchronizes FoK, and trains FoK-guided responses.

3.1 Framework overview

Starting from a base LLM M, we represent post-training changes as parameter deltas [12–14, 71]. The knowledge delta κ forms the knowledge-retrieval channel, which changes what the model knows and retrieves. The FoK delta φ forms the FoK channel, which judges whether the current knowledge state supports a correct answer. The training procedure has three parts:

- **On-policy supervised fine-tuning (OP-SFT).** OP-SFT updates κ on the model's own sampled responses and supervises only the corrected answer-relevant suffix. This part learns the target facts while saving a trajectory of knowledge states $\{\kappa_0, \kappa_1, \ldots, \kappa_T\}$ that later serves as supervision for FoK adaptation under the same base model.
- **Knowledge-delta FoK training (KΔ-FT).** KΔ-FT trains φ over the OP-SFT checkpoint trajectory κ_t. This lets FoK judgments follow the model's knowledge as it evolves, rather than remaining tied to a fixed knowledge boundary.
- **FoK-guided policy optimization (FGPO).** Finally, FGPO trains the model to generate a reasoning trace, make a yes/no FoK judgment, and produce either an answer or an explanation of why it cannot answer. Its reward evaluates the judgment and response together, so FoK is optimized as part of the generated decision rather than as a separate post-processing score.

With this two-channel update, FoK guides the LLM before answering. The model can first judge whether its current knowledge is sufficient, then either generate an answer or explain what knowledge is missing, which can support routing to a stronger model or retrieval-augmented generation.

3.2 On-policy supervised fine-tuning (OP-SFT)

Section 2.3 shows that standard knowledge training does not synchronize FoK with the updated knowledge state, and can severely interfere with the model's original FoK ability. To reduce this

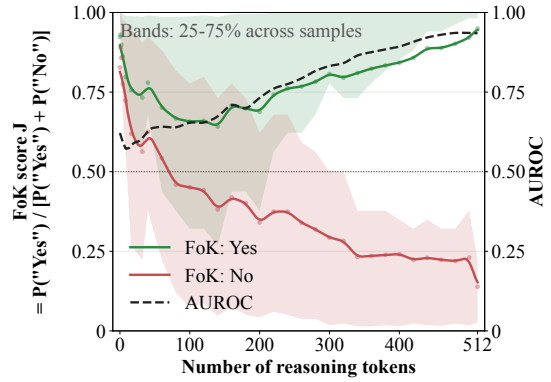


Figure 2: FoK score and AUROC over pre-judgment reasoning tokens. Green/red curves show mean FoK score for examples whose final FoK token is Yes/No. $J = 0.5$ denotes FoK uncertainty, while $J \approx 1$ or $J \approx 0$ denotes high certainty. The black curve reports AUROC of prefix-level J for predicting whether the final FoK token is Yes or No.

interference, we introduce on-policy supervised fine-tuning (OP-SFT). Specifically, OP-SFT keeps the model's own generated context and applies gradients only to the corrected answer-relevant span.

For each question q with gold answer set $\mathcal{A}_q$, we collect k complete sampled responses from the current LLM. For a sampled response $r = (r_{<s}, r_{[s:e]}, r_{>e})$, $r_{[s:e]}$ denotes the generated answer span. We then handle two cases. If the sampled response already answers correctly, we keep this answer span. If it answers incorrectly, we replace only this answer span with a gold answer $a^*(q) \in \mathcal{A}_q$ and leave the rest of the generated response unchanged. The repaired training response is

$$\bar{r}(q,r) = \big(r_{<s}, \bar{a}(q,r), r_{>e}\big), \qquad \bar{a}(q,r) = \begin{cases} r_{[s:e]} & r \text{ answers correctly,} \\ a^*(q) & r \text{ answers incorrectly.} \end{cases} \tag{1}$$

The training sequence is therefore not a static QA pair: it is sampled from the current LLM, changes only the wrong answer span, and uses loss only on $\bar{a}(q,r)$. For example, for $q =$"Who wrote *Pride and Prejudice*?" and $\mathcal{A}_q = \{\text{Jane Austen}\}$, OP-SFT repairs one sampled response as

$$\begin{aligned} q &: \text{Who wrote } \textit{Pride and Prejudice}?, \qquad \mathcal{A}_q = \{\text{Jane Austen}\}, \\ \bar{r}(q,r) &: \text{The book is a nineteenth-century novel. I think the author was} \\ &\quad \sout{\text{Charlotte Brontë}}\ \text{Jane Austen}, \text{ because she wrote English classics.} \end{aligned} \tag{2}$$

The red span is the first answer-relevant error in the sampled response; the green span is the supervised correction. The remaining generated text stays in the response, but receives no loss. The OP-SFT objective is therefore the local corrected-span loss:

$$\mathcal{L}_{\text{OP-SFT}} = -\,\mathbb{E}_{q,r} \sum_{u=1}^{|\bar{a}(q,r)|} \log P_{M_\kappa}\big(\bar{a}(q,r)_u \mid q, r_{<s}, \bar{a}(q,r)_{<u}\big). \tag{3}$$

We save checkpoints $\{\kappa_0, \kappa_1, \ldots, \kappa_T\}$ during OP-SFT, keeping a sequence of knowledge states.

3.3 Knowledge-delta FoK training (KΔ-FT)

After a knowledge update, the LLM's answerability changes, but its FoK does not automatically synchronize with that change. Training FoK only on the base or final LLM gives a state-specific answerability boundary that generalizes poorly across knowledge deltas. Final-state training also creates label imbalance because many updated questions become answerable.

To make FoK generalize across knowledge deltas, we introduce knowledge-delta FoK training (KΔ-FT), which uses the OP-SFT checkpoint trajectory $\{\kappa_0, \kappa_1, \ldots, \kappa_T\}$ as supervision states rather than supervising only one endpoint [76]. For each checkpoint κ_t, we combine it with the same FoK delta φ and train φ to predict whether M_{κ_t} can answer question q with the following loss:

$$\ell_\varphi(q, \kappa_t) = -\log P_{M_{\kappa_t,\varphi}}\left(\mathbf{1}[\text{ans}(M_{\kappa_t}, q) \in \mathcal{A}_q] \mid T_J(q)\right). \tag{4}$$

Here $\mathbf{1}[\cdot]$ denotes the indicator function: it equals 1 when the answer generated by M_{κ_t} belongs to $\mathcal{A}_q$, and equals 0 otherwise. $T_J(q)$ wraps q in the fixed Yes/No prompt "Can you answer this question correctly?" $M_{\kappa_t,\varphi}$ is the base model with the frozen knowledge delta and trainable FoK delta. For $y \in \{0,1\}$, $P_{M_{\kappa_t,\varphi}}(y \mid T_J(q))$ is the normalized Yes/No probability: $J(q)$ for $y = 1$ and $1 - J(q)$ for $y = 0$. The probability is evaluated under $M_{\kappa_t,\varphi}$, so FoK is trained with the same combined knowledge-and-FoK state used for judgment. The overall objective averages this checkpoint-specific loss over all questions and all OP-SFT checkpoints:

$$\mathcal{L}_{\mathrm{K}\Delta\text{-FT}}(\varphi) = \frac{1}{|\mathcal{Q}|(T+1)} \sum_{q \in \mathcal{Q}} \sum_{t=0}^{T} \ell_\varphi(q, \kappa_t). \tag{5}$$

Only φ is optimized in this stage; the knowledge checkpoints are fixed supervision states. Because the same question is evaluated under multiple κ_t, KΔ-FT trains FoK to follow answerability as it changes from pre-update to post-update knowledge, rather than to memorize a static question-level label detached from the current model state during updating.

3.4 FoK-guided policy optimization (FGPO)

After OP-SFT and KΔ-FT, a continuous FoK score still requires a decision threshold. FoK-guided policy optimization (FGPO) trains the judgment and subsequent response together, allowing the model to commit to answering or explaining within one generation [47, 72–74]. It generates a short context trace, makes a FoK judgment, and produces an answer or explanation:

```
<think>...reasoning...</think> <answer>yes|no</answer>
      <response>answer or explanation</response>
```

For a sampled trajectory $z = (h, j, u)$, where h is the reasoning trace, $j \in \{\mathrm{yes}, \mathrm{no}\}$ is the FoK judgment, and u is the response, we decompose the reward into FoK, answer, and format terms:

$$R(q, z) = R_{\mathrm{FoK}}(Y(q), j) + R_{\mathrm{ans}}(q, Y(q), j, u) + R_{\mathrm{fmt}}(z) - \lambda|h|. \tag{6}$$

Here $Y(q)$ is the answerability target for the current training example. Let $C(q, u) = \mathbf{1}[u \in \mathcal{A}_q]$ denote answer correctness, $E(q, u)$ denote whether an LLM judge accepts u as a valid explanation for not answering, and $F(z), S(z) \in \{0, 1\}$ denote valid format and proper sequence termination. The FoK term directly supervises the judgment token:

$$R_{\mathrm{FoK}}(Y, j) = \alpha_+ \mathbf{1}[\iota(j) = Y] - \alpha_- \mathbf{1}[\iota(j) \neq Y], \tag{7}$$

where $\iota(\mathrm{yes}) = 1$ and $\iota(\mathrm{no}) = 0$. The answer term scores the response under the chosen judgment:

$$\begin{aligned} R_{\mathrm{ans}}(q, Y, j, u) = {} & \mathbf{1}[j = \mathrm{yes}, Y = 1]\big(\beta_+ C(q, u) - \beta_-(1 - C(q, u))\big) \\ & - \mathbf{1}[j = \mathrm{yes}, Y = 0]\big(\bar{\beta}_+ C(q, u) + \bar{\beta}_-(1 - C(q, u))\big) \\ & + \mathbf{1}[j = \mathrm{no}, Y = 0]\big(\gamma_+ E(q, u) - \gamma_-(1 - E(q, u))\big) \\ & - \mathbf{1}[j = \mathrm{no}, Y = 1]\gamma_{\mathrm{skip}}. \end{aligned} \tag{8}$$

Thus, a "yes" judgment must be followed by a correct answer, while a "no" judgment must be justified by a valid explanation and is penalized on answerable examples. The format term is

$$R_{\mathrm{fmt}}(z) = \eta_F F(z) - \bar{\eta}_F(1 - F(z)) + \eta_S S(z) - \bar{\eta}_S(1 - S(z)). \tag{9}$$

Table 6 in Appendix C lists the implemented values. In implementation, wrong answers and malformed outputs receive larger penalties than valid "no" explanations receive positive reward, so the model is not rewarded for avoiding answerable questions by choosing the negative judgment.

4 Experiments

We ask whether FoK can be synchronized after an update (RQ1), which training stages contribute (RQ2), and whether the result persists across datasets, sequential updates, and seeds (RQ3). Sections 4.2, 4.3, and 4.5 address RQ1–RQ2; Section 4.4 and Appendix H address RQ3.

Table 2: Overall performance after one TriviaQA knowledge update on Qwen2.5-7B. Train columns use questions seen during the update; Test columns use held-out TriviaQA questions; General reports MMLU and ARC-Easy. ↑: higher is better; ↓: lower is better; $\rightarrow 0$: closer to zero is better. Bold and underline indicate the best and second-best results.

	Train			Test			General	
Method	Acc.↑	AUROC↑	Bias$\rightarrow 0$	Acc.↑	AUROC↑	Bias$\rightarrow 0$	MMLU↑	ARC↑
Base	0.680	0.702	0.381	<u>0.672</u>	0.688	0.402	0.687	0.811
+ Prob	0.680	0.723	0.409	0.672	0.730	0.407	0.687	0.811
+ N-Prob	0.680	0.723	0.409	0.672	0.730	0.407	0.687	0.811
+ EliCal	0.637	<u>0.808</u>	<u>0.263</u>	0.622	**0.810**	**0.254**	0.709	0.504
Vanilla K-SFT	<u>0.874</u>	0.458	0.500	0.546	0.519	0.500	<u>0.713</u>	**0.840**
+ Verbal-0	0.874	0.491	0.543	0.546	0.478	0.551	0.713	0.840
+ Verbal-10	0.874	0.567	0.456	0.546	0.529	0.467	0.713	0.840
+ Prob	0.874	0.590	0.495	0.546	0.541	0.498	0.713	0.840
+ N-Prob	0.874	0.590	0.495	0.546	0.542	0.498	0.713	0.840
+ Consis-Lex	0.874	0.539	0.482	0.546	0.531	0.486	0.713	0.840
+ Consis-Sem	0.874	0.553	0.472	0.546	0.567	0.467	0.713	0.840
+ SemEnt	0.874	0.538	0.481	0.546	0.526	0.504	0.713	0.840
+ EliCal	0.648	0.646	0.428	0.587	0.644	0.447	**0.723**	0.688
Ours	**0.961**	**0.817**	**0.245**	**0.692**	<u>0.809</u>	<u>0.308</u>	0.711	<u>0.830</u>

4.1 Evaluation setup

Datasets. We use TriviaQA [25] as the main QA dataset, with 5,000 update questions and 1,000 held-out questions. Additional FoK evaluations use SelfAware [68], Natural Questions [36], and WebQuestions [2]. MMLU [18] and ARC-Easy [6] evaluate general capability.

Baselines. We compare direct Yes/No FoK judgment, Vanilla K-SFT, verbalized confidence and 10-bin verbalized confidence [62], sequence probability and length-normalized sequence probability, lexical and semantic self-consistency, semantic entropy [9, 33], and EliCal [52].

Metrics. We report accuracy on TriviaQA, MMLU, and ARC-Easy. FoK quality is measured by AUROC [10] using the normalized Yes probability, plus two errors thresholded at $J > 0.5$: illusion rate (illu., $\Pr(\hat{J}=1 \mid Y=0)$) and omission rate (om., $\Pr(\hat{J}=0 \mid Y=1)$). We combine them as:

$$\text{Bias} = \begin{cases} \frac{\text{illu.}+\text{om.}}{2}, & \text{if illu.} \geq \text{om.}, \\ -\frac{\text{illu.}+\text{om.}}{2}, & \text{if illu.} < \text{om.}. \end{cases} \qquad \text{Bias} \in [-1, 1].$$

Implementation details. We evaluate on Qwen2.5-7B-Instruct [60] for the main experiments and on Qwen3-8B [61] as a thinking-mode backbone. We refer to them as Qwen2.5-7B and Qwen3-8B. We instantiate the three stages of Section 3 as LoRA adapters on top of the base model. OP-SFT updates the knowledge channel from corrected sampled responses, KΔ-FT trains the FoK channel over the OP-SFT checkpoint trajectory, and FGPO optimizes response generation using both channels. The full data construction, baseline definitions, evaluation protocol, optimizer settings, checkpoint schedule, and inference configuration are listed in Appendix C.

4.2 Overall performance

We compare our two-channel update with vanilla K-SFT and a suite of existing FoK estimators after a single TriviaQA knowledge update on Qwen2.5-7B. Table 2 gives the main numbers; Figure 3 decomposes the thresholded errors into illusion and omission. After vanilla K-SFT, in-domain accuracy rises sharply but held-out accuracy falls below the base model, so the update does not simply add knowledge. All existing estimators fall well below their base-model level, and most collapse into one-sided behavior: vanilla K-SFT and confidence-based estimators are dominated by illusion, judging almost every question answerable, while consistency-based estimators trade some illusion for a correspondingly large omission, leaving both error sources substantial. A likely cause is that the

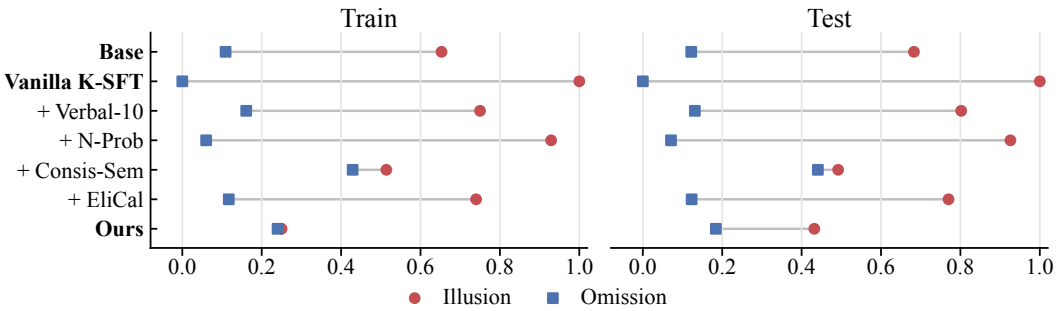


Figure 3: FoK error decomposition for Table 2. Red circles show illusion, the conditional rate of false "can answer" judgments; blue squares show omission, the conditional rate of false "cannot answer" judgments. The horizontal gap shows which error source determines the signed Bias value.

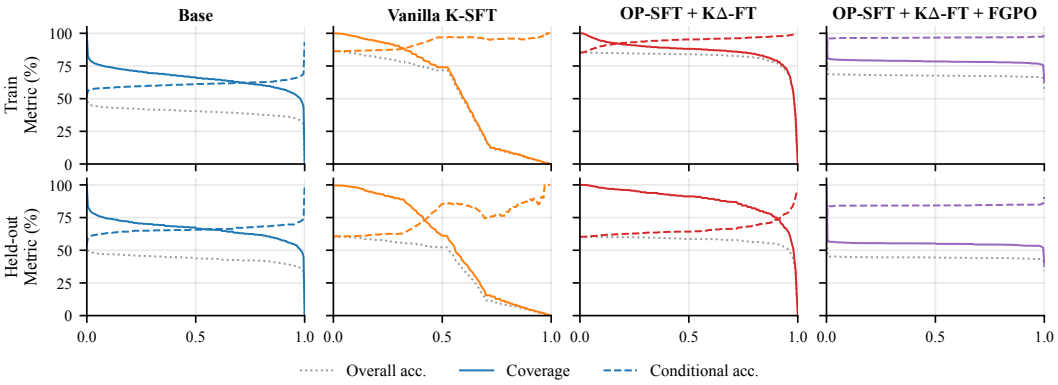


Figure 4: Coverage–accuracy trade-off on Qwen3-8B (thinking mode). Each panel varies the FoK threshold and plots coverage (solid, the fraction of questions the model judges itself able to answer), conditional accuracy on that answered subset (dashed), and overall accuracy with unanswered examples counted as incorrect (dotted).

knowledge update shifts the output distributions and sampling behavior that these estimators rely on, so a decision boundary calibrated before the update no longer carries the same meaning. This is what Takeaway 2 predicts: under a conventional knowledge update, existing FoK estimators are not a robust way to keep answerability judgments aligned with the updated model.

Our two-channel update shows a different profile. Held-out accuracy now rises above the base model, so the knowledge update generalizes instead of overfitting the update shard. FoK AUROC on the updated model reaches the level of the strongest existing estimator applied to the base model, and Bias is the lowest on the update set and the second lowest on held-out questions, although illusion still exceeds omission on held-out questions (Figure 3). We attribute this to two design choices. OP-SFT supervises only the corrected answer-relevant span on the model's own sampled responses, which only mildly disturbs FoK: held-out AUROC is 0.603 after OP-SFT alone, compared with 0.519 after vanilla K-SFT (Tables 2 and 4); KΔ-FT then trains the FoK channel directly on the changed answerability, instead of reusing a judgment calibrated before the update. Appendix F confirms this along the OP-SFT trajectory.

4.3 Coverage-accuracy trade-off under FoK judgment

Table 2 shows that FoK quality is recoverable; it remains to test whether the recovered FoK actually drives the decision of when to answer. We therefore vary the FoK threshold on Qwen3-8B and plot the coverage-accuracy curves in Figure 4. For Base and Vanilla K-SFT the curves are nearly flat on both splits: lowering coverage does not raise accuracy, so their FoK scores do not rank answerability. OP-SFT + KΔ-FT restores a monotonic trade-off, confirming that the learned FoK channel is consistent with answerability after the update. FGPO flattens the curves again, but for a

Table 3: Additional evaluation on Qwen2.5-7B. Left: all methods start from the same OP-SFT knowledge checkpoint; KΔ-FT adds the FoK adapter, so its combined answering state differs. Right: φ_1 is frozen after round 1; φ_i is retrained over each round's knowledge trajectory. Each round adds a disjoint shard from 5,000 update questions; all test columns use the same 1,000 held-out questions.

(a) FoK estimators after OP-SFT.

Estimator	Test Acc.	AUROC
Direct judgment	0.679	0.603
Prob	0.679	0.571
EliCal	0.679	0.618
KΔ-FT	0.699	0.770

(b) Three sequential TriviaQA updates.

Round	Test Acc.	φ_1 AUC	φ_i AUC
1	0.659	0.798	0.798
2	0.644	0.776	0.800
3	0.661	0.754	0.797

Table 4: Held-out stage-wise ablation on Qwen2.5-7B. OP-SFT updates knowledge, KΔ-FT adapts FoK, and FGPO optimizes judgment-guided generation. Higher accuracy and AUROC and lower error rates are better; Bias closer to zero is better. Update-set results are in Table 26.

Stage	Acc.$\uparrow$	AUROC$\uparrow$	Illu.$\downarrow$	Om.$\downarrow$	Bias$\rightarrow 0$
Base	0.672	0.688	0.683	0.122	0.402
Stage 1: + OP-SFT	0.679	0.603	0.601	0.283	0.442
Stage 2: + KΔ-FT	0.699	0.770	0.478	0.187	0.333
Stage 3: + FGPO	0.692	0.809	0.432	0.184	0.308

different reason: the scores collapse to 0 or 1, and the threshold becomes irrelevant once the policy has committed to answer or abstain. KΔ-FT thus provides a tunable coverage–accuracy trade-off, while FGPO answers more selectively, trading coverage for accuracy on answered questions.

4.4 Controlled comparison and sequential updates

Accuracy-preserving update. Table 3a tests whether avoiding destructive K-SFT alone suffices. After OP-SFT, direct judgment, Prob, and EliCal reach AUROC 0.603, 0.571, and 0.618, respectively; KΔ-FT reaches 0.770. Thus, FoK adaptation helps even when held-out accuracy is preserved. When no update is applied, strong unadapted estimators remain competitive and the accuracy gain of the update itself is modest: Base + EliCal reaches 0.810 AUROC in Table 2, compared with 0.809 for the full method. Once an update occurs, the unadapted estimators desynchronize from the new knowledge state, and the continually trained FoK channel is what preserves judgment quality.

Sequential updates and broader evaluation. Across three updates, the per-round FoK channel stays near 0.80 AUROC while the frozen channel falls to 0.754 (Table 3b). Cross-dataset tests on SelfAware, Natural Questions, and WebQuestions, plus a Natural Questions replication, extend the factual-QA evaluation. Independent Stage-1+2 runs yield test AUROC 0.777 ± 0.009 (four seeds); FGPO yields 0.811 ± 0.003 (three seeds from one shared initialization). Values are means and sample SDs. Appendix H reports all runs, cross-dataset results, and a MEMIT desynchronization test.

4.5 Ablation study

To locate where the FoK improvement in Table 2 comes from, we ablate the three stages cumulatively: OP-SFT alone, then OP-SFT + KΔ-FT, then the full pipeline with FGPO (Table 4). OP-SFT alone raises update-set accuracy from 0.680 to 0.966 (Table 26) but leaves the FoK judgment misaligned with the updated knowledge state. Adding KΔ-FT is the main source of the FoK gain: AUROC rises and Bias drops while held-out accuracy changes little (0.679 to 0.699), supporting the design choice of training a dedicated FoK channel over knowledge-delta states rather than relying on knowledge learning alone. FGPO raises held-out AUROC from 0.770 to 0.809 and incorporates the judgment into generation. On the separate Qwen3-8B evaluation, its nearly binary judgments explain the flattened threshold curves in Figure 4.

5 Related work

Functional metacognition. Metacognitive sensitivity and calibration are distinct quantities [11]. Human–LLM comparisons therefore require explicit behavioral definitions rather than equating similar reports with identical cognitive processes [58]. Recent work studies competing confidence biases under external advice [35] and how verbal confidence is computed from answer-related representations [34]. Our focus is correctness-supervised answerability judgment under changing model parameters. Koriat's cue-accessibility account and the cue-familiarity account of FoK [30, 31, 55] motivate our context analyses.

LLM uncertainty and hallucination. LLMs often generate fluent but factually false outputs, a failure studied as hallucination and factuality [20, 23]. Theoretical analyses show that this failure cannot be removed by pretraining alone: rare facts impose a Good–Turing-style lower bound on calibrated error, and common training and evaluation procedures can reward guessing rather than explicit uncertainty [27, 28]. This motivates uncertainty estimation and answerability judgment in LLMs [16]. Existing methods estimate uncertainty from generated outputs, including $P(\text{True})$ and $P(\text{IK})$ (probability that the model "knows" the answer), verbalized confidence, explicit "I don't know" (IDK) training, semantic entropy, and self-consistency [8, 9, 26, 32, 33, 44, 62, 63, 66]. Other methods read uncertainty from internal representations, including SAPLMA, CCS, INSIDE, MIND, C^3, and internal-state probes [1, 3, 4, 53, 57, 59]. A related line trains answerability behavior directly: R-Tuning teaches models to state that they do not know [69]; US-Tuning, CRaFT, and GRAIT analyze conflicts between answer learning and negative answerability supervision [38, 77, 78]; EliCal uses annotation-efficient elicitation and calibration for honesty [52]. Inference-time factuality methods such as ITI and DoLa intervene without additional training [5, 40], while TruthfulQA, HaluEval, FActScore, and SimpleQA evaluate factual support in generated answers [39, 43, 51, 65]. These works primarily ask whether a given model can express or reveal uncertainty about its current answers. We instead ask whether the FoK judgment tracks post-training updates to the model's knowledge.

Knowledge editing and continual learning. Post-training updates, knowledge editing, and continual learning change what an LLM knows after pretraining. ROME and MEMIT edit factual associations and evaluate edit success, locality, portability, and retention [13, 49, 50]. Continual-learning methods, including regularization and replay-based approaches, aim to preserve answer accuracy across sequential updates [21, 29, 48]. These methods focus on the update operator and on answer correctness, and usually do not test whether answerability judgment changes with the updated knowledge. We instead treat FoK as a training target co-learned with each update, so the answerability judgment moves with the model rather than lagging behind on a fixed checkpoint.

6 Conclusion

We studied how to keep functional FoK aligned with an LLM's answerability after knowledge updates. In factual QA, knowledge updating alone does not preserve this alignment. Our two-channel framework trains knowledge and FoK separately and then optimizes the judgment together with the subsequent response. Controlled OP-SFT comparisons, cross-dataset evaluation, and three sequential updates support the usefulness of explicit FoK adaptation, with seed variation quantifying its stability in the main setting.

Acknowledgments and Disclosure of Funding

This research was partially supported by National Natural Science Foundation of China (No.62502404), Hong Kong Research Grants Council (Research Impact Fund No.R1015-23, Collaborative Research Fund No.C1043-24GF, RGC Research Fellow Scheme No.RFS2627-1S03, General Research Fund No.11218325, No.11212926), Institute of Digital Medicine of City University of Hong Kong (No.9229503), and Tencent (CCF-Tencent Open Fund, Tencent Rhino-Bird Focused Research Program, Tencent University Cooperation Project).

References

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

[2] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, 2013. URL https://aclanthology.org/D13-1160/.

[3] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations (ICLR)*, 2023.

[4] 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. URL https://openreview.net/forum?id=Zj12nzlQbz.

[5] Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. DoLa: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Th6NyL07na.

[6] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge, 2018.

[7] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[8] Roi Cohen, Konstantin Dobler, Eden Biran, and Gerard de Melo. I don't know: Explicit modeling of uncertainty with an [IDK] token. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-0349.

[9] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024. doi: 10.1038/s41586-024-07421-0.

[10] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. doi: 10.1016/j.patrec.2005.10.010.

[11] Stephen M. Fleming and Hakwan C. Lau. How to measure metacognition. *Frontiers in Human Neuroscience*, 8:443, 2014. doi: 10.3389/fnhum.2014.00443.

[12] Zichuan Fu, Xiangyang Li, Chuhan Wu, Yichao Wang, Kuicai Dong, Xiangyu Zhao, Mengchen Zhao, Huifeng Guo, and Ruiming Tang. A unified framework for multi-domain CTR prediction via large language models. *ACM Transactions on Information Systems*, 43(5):117, 2025.

[13] Zichuan Fu, Xian Wu, Guojing Li, Yingying Zhang, Yefeng Zheng, Tianshi Ming, Yejing Wang, Wanyu Wang, and Xiangyu Zhao. Model merging for knowledge editing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, 2025.

[14] Zichuan Fu, Xian Wu, Yejing Wang, Wanyu Wang, Shanshan Ye, Hongzhi Yin, Yi Chang, Yefeng Zheng, and Xiangyu Zhao. Training-free LLM merging for multi-task learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

[15] Zichuan Fu, Xian Wu, Guojing Li, Yejing Wang, Yijun Chen, Zihao Zhao, Yixuan Luo, Hanyu Yan, Yefeng Zheng, and Xiangyu Zhao. Tandem: Riding together with large and small language models for efficient reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 42286–42302, 2026.

[16] Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.366. URL https://aclanthology.org/2024.naacl-long.366/.

[17] Joseph T. Hart. Memory and the feeling-of-knowing experience. *Journal of Educational Psychology*, 56(4):208–216, 1965.

[18] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

[19] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

[20] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 2024. doi: 10.1145/3703155.

[21] Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. Towards continual knowledge learning of language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[22] 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: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.389. URL https://aclanthology.org/2024.naacl-long.389/.

[23] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023. doi: 10.1145/3571730.

[24] Pengyue Jia, Yiding Liu, Xiangyu Zhao, Xiaopeng Li, Changying Hao, Shuaiqiang Wang, and Dawei Yin. MILL: Mutual verification with large language models for zero-shot query expansion. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2498–2518, 2024.

[25] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

[26] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. doi: 10.48550/arXiv.2207.05221.

[27] Adam Tauman Kalai and Santosh S. Vempala. Calibrated language models must hallucinate. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, 2024. doi: 10.1145/3618260.3649777.

[28] Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025. doi: 10.48550/arXiv.2509.04664.

[29] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526, 2017. doi: 10.1073/pnas.1611835114.

[30] Asher Koriat. How do we know that we know? The accessibility model of the feeling of knowing. *Psychological Review*, 100(4):609–639, 1993.

[31] Asher Koriat and Ravit Levy-Sadot. The combined contributions of the cue-familiarity and accessibility heuristics to feelings of knowing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 27(1):34–53, 2001.

[32] Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth A. Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in LLMs. *arXiv preprint arXiv:2406.15927*, 2024. doi: 10.48550/arXiv.2406.15927.

[33] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=VD-AYtP0dve`.

[34] Dharshan Kumaran, Arthur Conmy, Federico Barbero, Simon Osindero, Viorica Patraucean, and Petar Veličković. How do LLMs compute verbal confidence, 2026. URL `https://arxiv.org/abs/2603.17839`.

[35] Dharshan Kumaran, Stephen M. Fleming, Larisa Markeeva, Joe Heyward, Andrea Banino, Mrinal Mathur, Razvan Pascanu, Simon Osindero, Benedetto De Martino, Petar Veličković, and Viorica Patraucean. Competing biases underlie overconfidence and underconfidence in LLMs. *Nature Machine Intelligence*, 8:614–627, 2026. doi: 10.1038/s42256-026-01217-9.

[36] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL `https://aclanthology.org/Q19-1026/`.

[37] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, 2017.

[38] Jiaqi Li, Yixuan Tang, and Yi Yang. Know the unknown: An uncertainty-sensitive method for LLM instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2972–2989. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.153. URL `https://aclanthology.org/2025.findings-acl.153/`.

[39] Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.397. URL `https://aclanthology.org/2023.emnlp-main.397/`.

[40] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, 2023.

[41] Xiaopeng Li, Pengyue Jia, Derong Xu, Yi Wen, Yingyi Zhang, Wenlin Zhang, Wanyu Wang, Yichao Wang, Zhaocheng Du, Xiangyang Li, Yong Liu, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. A survey of personalization: From RAG to agent. *arXiv preprint arXiv:2504.10147*, 2025. doi: 10.48550/arXiv.2504.10147.

[42] Yuchen Li, Hengyi Cai, Rui Kong, Xinran Chen, Jiamin Chen, Jun Yang, Haojie Zhang, Jiayi Li, Jiayi Wu, Yiqun Chen, Changle Qu, Wenwen Ye, Lixin Su, Xinyu Ma, Lingyong Yan, Long Xia, Daiting Shi, Junfeng Wang, Xiangyu Zhao, Jiashu Zhao, Haoyi Xiong, Shuaiqiang Wang, and Dawei Yin. Towards AI search paradigm. *arXiv preprint arXiv:2506.17188*, 2025. doi: 10.48550/arXiv.2506.17188.

[43] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

[44] Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022.

[45] Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Zijian Zhang, Feng Tian, and Yefeng Zheng. Large language model distilling medication recommendation model. *arXiv preprint arXiv:2402.02803*, 2024. doi: 10.48550/arXiv.2402.02803.

[46] Qidong Liu, Xiangyu Zhao, Yejing Wang, Zijian Zhang, Howard Zhong, Chong Chen, Xiang Li, Wei Huang, and Feng Tian. Bridge the domains: Large language models enhanced cross-domain sequential recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025. doi: 10.1145/3726302.3729911.

[47] Shuchang Liu, Qingpeng Cai, Bowen Sun, Yuhao Wang, Ji Jiang, Dong Zheng, Kun Gai, Peng Jiang, Xiangyu Zhao, and Yongfeng Zhang. Exploration and regularization of the latent action space in recommendation. In *Proceedings of the ACM Web Conference 2023*, 2023. doi: 10.1145/3543507.3583244.

[48] Yun Luo, Zhicheng Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023.

[49] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022.

[50] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations (ICLR)*, 2023.

[51] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

[52] Shiyu Ni, Keping Bi, Jiafeng Guo, Minghao Tang, Jingtong Wu, Zengxin Han, and Xueqi Cheng. Annotation-efficient universal honesty alignment. *arXiv preprint arXiv:2510.17509*, 2025. doi: 10.48550/arXiv.2510.17509.

[53] Shiyu Ni, Keping Bi, Jiafeng Guo, Lulu Yu, Baolong Bi, and Xueqi Cheng. Towards fully exploiting LLM internal states to enhance knowledge boundary perception. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24315–24329. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1184. URL https://aclanthology.org/2025.acl-long.1184/.

[54] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms, 2018. URL https://arxiv.org/abs/1803.02999.

[55] Lynne M. Reder and Frank E. Ritter. What determines initial feeling of knowing? Familiarity with question terms, not with the answer. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18(3):435–451, 1992.

[56] Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026. URL https://arxiv.org/abs/2601.19897.

[57] Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. The curious case of hallucinatory (un)answerability: Finding truths in the hidden states of over-confident large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3607–3625. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.220. URL https://aclanthology.org/2023.emnlp-main.220/.

[58] Mark Steyvers and Megan A. K. Peters. Metacognition and uncertainty communication in humans and large language models. *Current Directions in Psychological Science*, 35(3): 131–139, 2026. doi: 10.1177/09637214251391158.

[59] Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. Unsupervised real-time hallucination detection based on the internal states of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14379–14391. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.854. URL https://aclanthology.org/2024.findings-acl.854/.

[60] Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. doi: 10.48550/arXiv.2412.15115.

[61] Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. doi: 10.48550/arXiv.2505.09388.

[62] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330/.

[63] 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 (ICLR)*, 2023.

[64] Yile Wang, Peng Li, Maosong Sun, and Yang Liu. Self-knowledge guided retrieval augmentation for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10303–10315. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.691. URL https://aclanthology.org/2023.findings-emnlp.691/.

[65] Jason Wei, Karina Nguyen, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models (SimpleQA). *arXiv preprint arXiv:2411.04368*, 2024. doi: 10.48550/arXiv.2411.04368.

[66] Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

[67] Derong Xu, Xinhang Li, Ziheng Zhang, Zhenxi Lin, Zhihong Zhu, Zhi Zheng, Xian Wu, Xiangyu Zhao, Tong Xu, and Enhong Chen. Harnessing large language models for knowledge graph question answering via adaptive multi-aspect retrieval-augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25570–25578, 2025. doi: 10.1609/aaai.v39i24.34747.

[68] 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*, pages 8653–8665, 2023. doi: 10.18653/v1/2023.findings-acl.551. URL https://aclanthology.org/2023.findings-acl.551/.

[69] 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: Human Language Technologies (Volume 1: Long Papers)*, pages 7113–7139. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.394. URL https://aclanthology.org/2024.naacl-long.394/.

[70] Wenlin Zhang, Xiaopeng Li, Yingyi Zhang, Pengyue Jia, Yichao Wang, Huifeng Guo, Yong Liu, and Xiangyu Zhao. Deep research: A survey of autonomous research agents. *arXiv preprint arXiv:2508.12752*, 2025. doi: 10.48550/arXiv.2508.12752.

[71] Zijian Zhang, Xiangyu Zhao, Qidong Liu, Chunxu Zhang, Qian Ma, Wanyu Wang, Hongwei Zhao, Yiqi Wang, and Zitao Liu. PromptST: Prompt-enhanced spatio-temporal multi-attribute prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3195–3205, 2023.

[72] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 95–103, 2018. doi: 10.1145/3240323.3240374.

[73] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

[74] Xiangyu Zhao, Long Xia, Lixin Zou, Hui Liu, Dawei Yin, and Jiliang Tang. UserSim: User simulation via supervised generative adversarial network. In *Proceedings of the Web Conference 2021*, pages 3582–3589, 2021. doi: 10.1145/3442381.3450125.

[75] 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*, 2023.

[76] Jiajun Zhou, Chenxuan Xie, Shengbo Gong, Zhenyu Wen, Xiangyu Zhao, Qi Xuan, and Xiaoniu Yang. Data augmentation on graphs: A technical survey. *ACM Computing Surveys*, 57(11):274, 2025. doi: 10.1145/3732282.

[77] Runchuan Zhu, Zinco Jiang, Jiang Wu, Zhipeng Ma, Jiahe Song, Fengshuo Bai, Dahua Lin, Lijun Wu, and Conghui He. GRAIT: Gradient-driven refusal-aware instruction tuning for effective hallucination mitigation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4006–4021. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.223. URL https://aclanthology.org/2025.findings-naacl.223/.

[78] Runchuan Zhu, Zhipeng Ma, Jiang Wu, Junyuan Gao, Jiaqi Wang, Dahua Lin, and Conghui He. Utilize the flow before stepping into the same river twice: Certainty represented knowledge flow for refusal-aware instruction tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26157–26165, 2025. doi: 10.1609/aaai.v39i24.34812.

A Limitations and future work

We evaluate the framework on short-answer factual QA with gradient-based updates, including three sequential update rounds. The MEMIT experiment in Appendix H tests desynchronization under one non-gradient editor, not whether our full framework repairs that editor. Longer update sequences, procedural tasks such as code generation and mathematical proof, and backbones whose baseline

FoK signal is near chance remain outside the tested scope. Strong unadapted estimators remain competitive without an update.

The cue analyses do not include a matched pretrained-model control, so they cannot establish whether training induces cue sensitivity or preserves sensitivity already in the backbone. Causal recovery locations depend on the prompt, token positions, and intervention; they do not establish a uniquely localized or human-equivalent metacognitive mechanism. More broadly, we use FoK in a functional sense, as answerability judgments modeled after human FoK, without implying subjective experience. Seed variation covers Stages 1–2 and FGPO on the main TriviaQA setting, not every preliminary or cross-dataset experiment.

B Broader impact

This work targets reliability after LLM knowledge updates by making answerability judgments track the updated model state. The intended positive impact is fewer unsupported answers after fine-tuning or continual updates. A possible negative impact is that better answerability judgments could increase trust in systems whose deployment still depends on data quality, monitoring, and domain-specific safety constraints. The method is not a content-safety system on its own and should be combined with deployment-specific safeguards before release.

C Additional experimental settings

Data construction. The main experiments use TriviaQA as the knowledge-update corpus and the in-domain QA benchmark. The update set contains 5,000 questions, and the held-out TriviaQA test split contains 1,000 questions. For OP-SFT, we sample multiple responses from the current answer model for each update question, repair the answer-relevant span when the sampled response is wrong, and train only the corrected answer span. The reported main setting uses 32 sampled responses per question. General capability is evaluated separately on MMLU and ARC-Easy, using the same generator whose TriviaQA result is reported in the main table.

Evaluation protocol. Answer accuracy on TriviaQA is the fraction of generated responses that match the gold answer, with surface-form equivalence checked by Qwen2.5-32B-Instruct [60] used as an LLM judge [75]. MMLU and ARC-Easy use their standard multiple-choice accuracy. Generation and judging are run with vLLM in bfloat16 precision; the 32B judge uses tensor parallelism across the available GPUs. FoK metrics use the model's pre-answer Yes/No judgment score: AUROC is computed from the normalized probability of the Yes token, and illusion, omission, and Bias are computed from the thresholded Yes/No prediction. Since illusion and omission are conditional error rates in $[0, 1]$, the signed Bias score lies in $[-1, 1]$. Positive values mean illusion errors dominate, and negative values mean omission errors dominate.

Baseline protocol. All FoK estimators are evaluated under the answer model used by that row. Base rows use the unadapted model; Vanilla K-SFT rows use the knowledge-updated model; Ours uses the combined knowledge and FoK adapters. Verbal-0 and Verbal-10 ask the model for a direct confidence judgment, Prob and N-Prob use sequence likelihood, Consis-Lex and Consis-Sem use multiple sampled responses, semantic entropy clusters sampled responses by meaning, and EliCal uses the released EliCal adapter whenever its public checkpoint is available for the backbone. Unlike estimators that only score fixed answers, loading the EliCal adapter can change answering behavior and answer accuracy. The fixed-answer EliCal transfer protocol in Appendix H is reported separately.

Training protocol. All trainable components are implemented as LoRA adapters on top of Qwen2.5-7B-Instruct. Vanilla K-SFT is trained with standard supervised fine-tuning on the same update questions. OP-SFT trains the knowledge adapter from repaired sampled responses. KΔ-FT freezes the OP-SFT knowledge checkpoints and trains a FoK adapter against labels collected from those knowledge states. FGPO then uses the knowledge and FoK adapters jointly for response optimization. Table 5 lists the full training hyperparameters.

Compute resources. The main experiments were run on a single multi-GPU server. Training used multi-GPU DDP with DeepSpeed ZeRO-2. Generation and judging used vLLM in bfloat16

precision, and the Qwen2.5-32B judge used tensor parallelism across the available GPUs. We report the parallelism and training configuration needed for reproduction, but did not log exact wall-clock time for every preliminary and failed run.

Table 5: Training hyperparameters for the main experiments.

Parameter	Value
Backbone	Qwen2.5-7B-Instruct
Training framework	LLaMA Factory
Adapter type	LoRA, rank 16, alpha 32, dropout 0
Parallelism	Multi-GPU DDP with DeepSpeed ZeRO-2
Precision	bfloat16
Vanilla K-SFT LR / epochs	10^{-4} / 5
OP-SFT sampled responses / epochs	32 / 1
OP-SFT LR	2×10^{-5}
KΔ-FT LR / epochs	10^{-4} / 3
FGPO LR / epochs	10^{-6} / 1
Per-device batch size	1
Gradient accumulation	1
Schedule	Cosine with 0.03 warmup ratio
FoK score	normalized `Yes`/`No` token probability

FGPO reward values. Table 6 lists the implemented FGPO rewards. The base reward gives $R_{\mathrm{FoK}} + R_{\mathrm{ans}}$ for each combination of answerability target, judgment, and response outcome. An output from which no yes/no judgment can be parsed receives -1.0 in place of the base reward. A further $+0.1$ is added when the output follows the full `<think>`/`<answer>`/`<response>` format, $+0.1$ when generation ends with the EOS token and -0.2 otherwise, and the length term uses $\lambda = 0.001$ with $|h|$ measured in characters of the reasoning trace.

Table 6: Base FGPO reward $R_{\mathrm{FoK}} + R_{\mathrm{ans}}$ by target Y, judgment j, and response outcome.

Y	j	Response outcome	Base reward
1	yes	correct answer	$+2.0$
1	yes	wrong answer	-1.5
1	no	—	-1.0
0	no	valid explanation	$+0.5$
0	no	invalid explanation	-0.5
0	yes	correct answer	-0.5
0	yes	wrong answer	-2.0

Existing assets and terms of use. Table 7 lists the external datasets, models, and code packages used in the experiments. We use each asset only for research evaluation or fine-tuning under the terms specified by the original publisher.

Table 7: External assets used in the experiments and their publisher-specified licenses or terms of use.

Asset	Use in this paper	License or terms
TriviaQA	Knowledge-update corpus and in-domain QA benchmark	Apache-2.0 (source)
MMLU	General capability benchmark	MIT License (source)
ARC-Easy	General capability benchmark	CC BY-SA 4.0 (source)
Qwen2.5-7B-Instruct and Qwen2.5-32B-Instruct	Main answer model and LLM judge	Apache-2.0 (Qwen 7B and 32B model cards)
Qwen3-8B	Thinking-mode backbone for the selective-prediction evaluation	Apache-2.0 (Qwen model card)
LLaMA-3.1 and LLaMA-2 checkpoints	Supplementary FoK synchronization results in Appendix G	Meta Llama 3.1 Community License and Llama 2 Community License (3.1, 2)
LLaMA Factory	LoRA fine-tuning framework	Apache-2.0 (source)
vLLM	Batched generation and LLM judging	Apache-2.0 (source)
EleutherAI lm-evaluation-harness	MMLU and ARC-Easy evaluation	MIT License (source)
EliCal official adapter	FoK baseline	Publisher model-card terms (source)

D Causal tracing and weight-delta supplementary figures

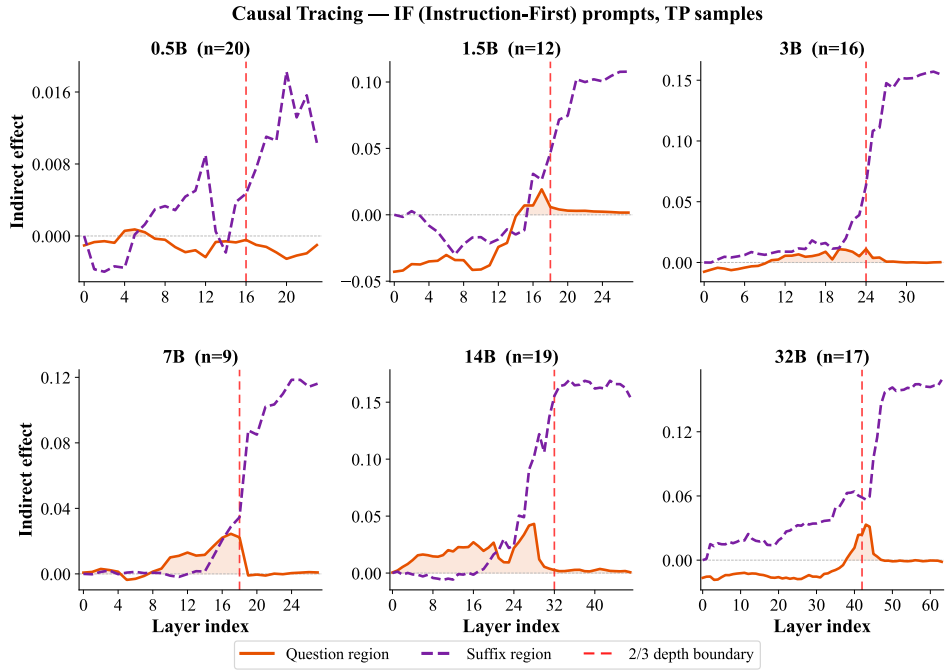


Figure 5: Per-layer indirect effect on true-positive (TP) samples for the instruction-first (IF) prompt across six Qwen2.5 sizes. The solid orange curve shows the question-region effect, the dashed purple curve shows the suffix-region effect, and the red dashed line marks two-thirds of model depth. The question-region effect concentrates in early and middle layers, whereas the suffix-region effect rises mainly after this boundary at every tested scale.

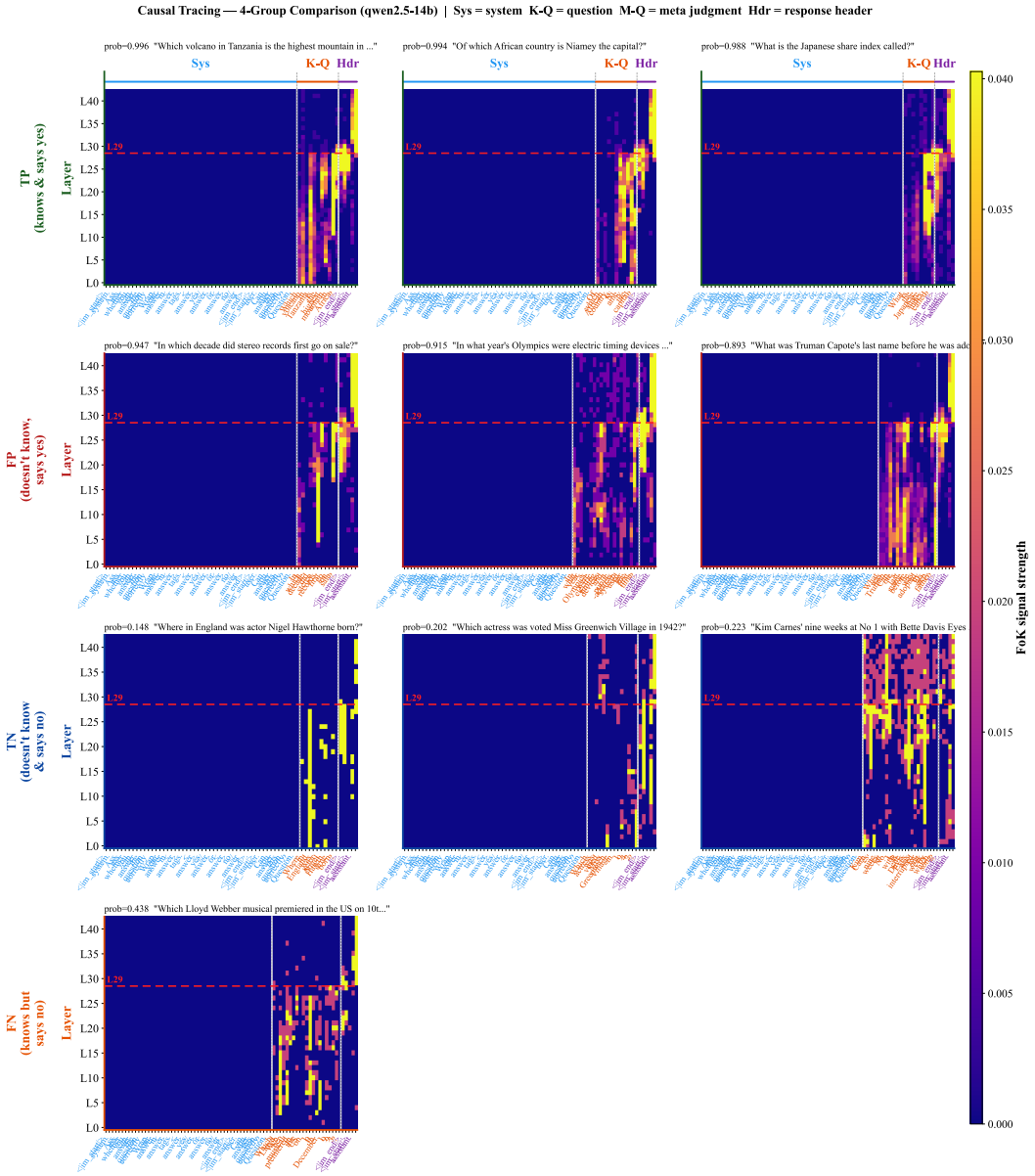


Figure 6: ROME-style causal tracing on Qwen2.5-14B, broken down by outcome group. Rows: TP (answers correctly and judges yes), FP (answers incorrectly but judges yes), TN (answers incorrectly and judges no), and FN (answers correctly but judges no). Each row shows three samples, except FN, which has only one sample with $p > 0.4$. Each cell is the layer $\times$ token restoration heatmap; tokens are grouped into Sys (system), K-Q (knowledge question), and Hdr (response header). The red dashed line at layer 29 of 48 (about 60% of depth) marks the depth boundary used in our analysis. The suffix-region (Hdr) signal appears mainly in deep layers across all four outcome groups, suggesting that the late FoK signal does not simply track answer correctness.

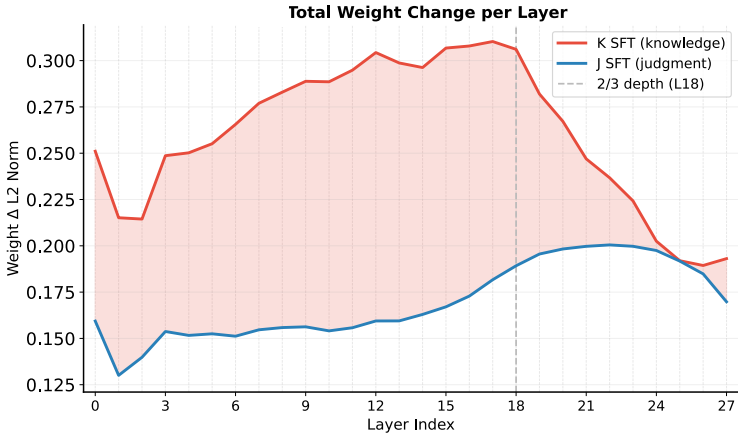
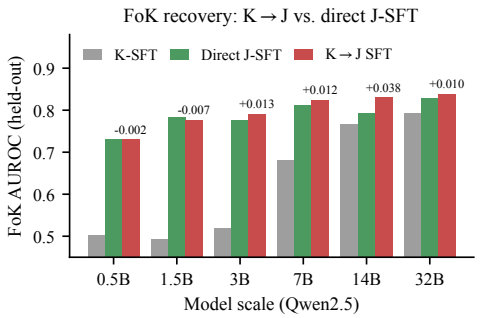
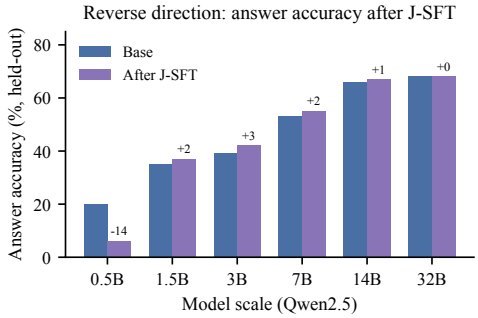


Figure 7: Layer-wise weight delta $\|\Delta W_\ell\|_F$ after one epoch of K-SFT (red) vs. J-SFT (blue) on Qwen2.5-7B (full SFT from base, same data budget). K-SFT changes concentrate in shallow/mid layers, while J-SFT changes remain small until the deep layers. This weight-side pattern is consistent with the depth separation observed in causal tracing.

E FoK dependency on knowledge



(a) FoK recovery. Numbers above the K→J bars give the gap to direct J-SFT. K→J recovers FoK at every scale, stays within 0.007 of direct J-SFT at 0.5B and 1.5B, and exceeds it from 3B upward.



(b) Reverse direction. Answer accuracy before and after J-SFT. From 1.5B upward, J-SFT changes accuracy by at most 3 points; at 0.5B it lowers accuracy by 14 points.

Figure 8: FoK dependence on updated knowledge, from the same held-out TriviaQA runs as Table 1. Left: judgment training after a knowledge update restores FoK at every scale. Right: judgment training alone leaves answer accuracy largely unchanged except at 0.5B.

F FoK judgment along OP-SFT training

Figure 9 tests whether the trained FoK channel only works at the final OP-SFT checkpoint. Answer accuracy changes substantially along the OP-SFT trajectory, while FoK AUROC remains above 0.80 on both splits for every checkpoint. The thresholded errors show a calibration shift: omission is larger near the base model, whereas illusion grows on later held-out checkpoints. Thus KΔ-FT provides a stable ranking signal across the knowledge trajectory, but threshold calibration still depends on the underlying knowledge state of the model.

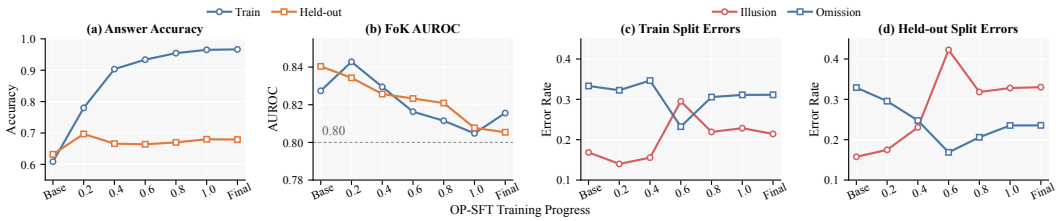


Figure 9: In a separate small-scale study, we evaluate a fixed, trained KΔ-FT FoK channel across successive OP-SFT knowledge checkpoints. We report answer accuracy, FoK AUROC, and thresholded illusion and omission errors.

G FoK synchronization results across model scale and family

Table 8: Qwen2.5 size scaling in a dedicated setup (5,000 update and 1,000 held-out questions).

	Base		K-SFT				J-SFT	K→J SFT		
Model	EM	AUROC	EM	AUROC	Illu.	Om.	AUROC	EM	AUROC	Illu.
0.5B	4.2%	0.523	16.9%	0.502	0%	99%	0.801	13.1%	0.763	3%
1.5B	30.9%	0.527	35.5%	0.500	66%	48%	0.824	34.8%	0.815	28%
3B	35.4%	0.724	45.7%	0.519	24%	80%	0.830	45.4%	0.822	40%
7B	47.2%	0.735	55.8%	0.592	28%	51%	0.855	55.1%	0.833	82%
14B	59.8%	0.756	65.2%	0.767	73%	9%	0.850	66.1%	0.814	85%
32B	61.2%	0.812	66.8%	0.792	80%	5%	0.844	66.8%	0.831	91%

Table 9: Cross-family FoK synchronization results with 5,000 update and 1,000 held-out questions.

Model	Base AUROC	K-SFT AUROC	J-SFT AUROC	K→J SFT AUROC	K→J SFT EM	Note
Qwen2.5-7B	0.735	0.592	0.855	0.833	55.1%	Reference
LLaMA-3.1-8B (LR $= 2 \times 10^{-6}$)	0.646	0.503	0.786	0.836	8.6%	K→J forgetting
LLaMA-3.1-8B (LR $= 10^{-6}$)	0.646	0.526	0.782	0.749	73.5%	LR corrected
LLaMA-2-7B	0.520	0.500	0.500	0.500	63.5%	No FoK improvement
LLaMA-2-13B	0.525	0.500	0.500	0.500	70.1%	No FoK improvement

Knowledge training and FoK degradation. Consistent with the main-text findings, the K-SFT results in Table 8 show that knowledge training can improve answer accuracy while impairing FoK. An update optimized for answering alone need not preserve the model's ability to distinguish its correct answers from errors, motivating explicit FoK synchronization rather than assuming that knowledge gains automatically improve self-judgment.

LLaMA-2 negative result. The LLaMA-2 results reveal a distinction between acquiring knowledge and judging answerability. Although answer accuracy improves, FoK remains at chance after both knowledge and judgment training (Table 9). This suggests that correctness supervision alone is insufficient to elicit a discriminative FoK signal in this setting: the model can learn answers without learning to distinguish when its own answers are correct.

H Additional validation

We conduct additional small-scale experiments, each with its own training and evaluation setup, to examine different aspects of FoK behavior.

H.1 Accuracy-preserving updates and sequential knowledge changes

OP-SFT comparison. Table 10 compares FoK estimators starting from the same OP-SFT knowledge checkpoint. Adding a FoK adapter changes the combined model state, and the KΔ-FT row therefore has a different answer accuracy. For Base, EliCal uses its released adapter, which is trained extensively and therefore changes answering behavior. When combined with OP-SFT, EliCal uses an adapter with only a small amount of training, so answer accuracy is unchanged.

Table 10: FoK evaluation before and after OP-SFT on the 1,000-question held-out TriviaQA split. All OP-SFT rows start from the same knowledge checkpoint.

Answer model	FoK estimator	Test Acc.	Test AUROC
Base	Direct judgment	0.672	0.688
Base	Prob	0.672	0.730
Base	EliCal	0.622	0.810
OP-SFT	Direct judgment	0.679	0.603
OP-SFT	Prob	0.679	0.571
OP-SFT	EliCal	0.679	0.618
OP-SFT + FoK adapter	KΔ-FT	0.699	0.770

Three sequential updates. We divide 5,000 TriviaQA update questions into three disjoint shards. Round i applies OP-SFT with on-policy rollouts to the previous round's model using shard i. The frozen FoK channel φ_1 is trained after round 1 and kept fixed thereafter. The adaptive channel φ_i is trained with KΔ-FT on each round's knowledge trajectory. All held-out columns use the same 1,000 questions. Table 11 reports before/after answer accuracy on the *new shard for that round*. The adaptive channel maintains discrimination and reduces illusion in later rounds, but held-out answer accuracy remains below Base.

Table 11: Three sequential TriviaQA updates. Illusion is the conditional rate of predicting Yes on questions the current answering model gets wrong.

	New-shard	Test	φ_1 (frozen)		φ_i (adaptive)	
State	Acc. before $\rightarrow$ after	Acc.	AUROC	Illusion	AUROC	Illusion
Base	—	0.672	—	—	—	—
Round 1	$0.575 \rightarrow 0.972$	0.659	0.798	0.455	0.798	0.455
Round 2	$0.676 \rightarrow 0.938$	0.644	0.776	0.688	0.800	0.202
Round 3	$0.631 \rightarrow 0.938$	0.661	0.754	0.752	0.797	0.195

H.2 Cross-dataset evaluation and a non-gradient update

Zero-shot transfer. We evaluate the TriviaQA-trained systems on SelfAware (3,369 questions), Natural Questions (3,610), and WebQuestions (2,032), without additional training (Table 12). FoK targets each evaluated model's own answer correctness. In SelfAware, unanswerable questions count as incorrect answers; this adapts the original testbed to model-specific answerability rather than using its answerability annotation as a fixed target. In this transfer experiment, EliCal scores the fixed answers of the corresponding Base or OP-SFT generator, so each paired row has the same answer accuracy. This differs from the adapter-based answering evaluation of Table 10.

Table 12: Zero-shot transfer from TriviaQA. SA: SelfAware; NQ: Natural Questions; WebQ: WebQuestions. Acc. measures answer correctness; AUROC measures FoK discrimination.

	SA		NQ		WebQ	
System	Acc.	AUROC	Acc.	AUROC	Acc.	AUROC
Base	0.282	0.768	0.376	0.665	0.527	0.597
Base + EliCal	0.282	0.702	0.376	0.735	0.527	0.636
Vanilla K-SFT	0.187	0.569	0.191	0.571	0.224	0.519
OP-SFT + EliCal	0.305	0.590	0.381	0.579	0.539	0.515
Ours	0.305	0.810	0.381	0.733	0.539	0.644

Table 13: SelfAware detail for the same zero-shot systems. TPR is the rate of predicting Yes on correctly answered questions; TNR is the rate of predicting No on incorrectly answered questions.

System	AUROC	TPR	TNR	Answer Acc.
Base	0.768	0.388	0.898	0.282
Vanilla K-SFT	0.569	0.302	0.760	0.187
Ours	0.810	0.765	0.701	0.305

Natural Questions replication. Separately from zero-shot transfer, we run the full pipeline on a 5,000-update / 1,000-held-out Natural Questions split (Table 14). The resulting update-shard accuracy is 0.925. This replicates the knowledge-update comparison on a second factual-QA dataset.

Table 14: Full-pipeline replication on Natural Questions. This held-out split differs from the zero-shot population in Table 12.

System	Held-out Acc.	FoK AUROC
Base	0.319	0.629
Vanilla K-SFT	0.199	0.585
Ours	0.366	0.647

MEMIT. We apply one MEMIT batch edit [50] to 200 facts in the zsRE setting [37]. Table 15 measures the edited facts and the same 1,000 held-out TriviaQA questions before and after editing. Edited-fact accuracy rises while FoK AUROC decreases. Knowledge and FoK thus also desynchronize under a non-gradient operator.

Table 15: Knowledge and functional FoK before and after a MEMIT edit. Train denotes 200 edited zsRE facts; Test denotes 1,000 held-out TriviaQA questions.

	Train Acc.	Train AUROC	Test Acc.	Test AUROC
Before edit	0.035	0.594	0.633	0.726
After edit	0.640	0.553	0.627	0.709

H.3 Training-seed variation

Table 16 reports independent training runs for Vanilla K-SFT (three seeds) and OP-SFT + KΔ-FT (four seeds). The Vanilla K-SFT seed runs train a LoRA adapter (rank 16, alpha 32) on standard SFT data built from the update questions, with learning rate 10^{-4}, 5 epochs, per-device batch size 8, and gradient accumulation 1. Table 17 varies the FGPO seed from one shared Stage-1+2 initialization. We report the arithmetic mean and sample standard deviation, $s = \sqrt{\sum_i (x_i - \bar{x})^2/(n-1)}$, over the available runs; Vanilla K-SFT statistics use its three seeds (43, 44, and 0). These statistics quantify training-seed variation under the fixed evaluation split.

Table 16: Training-seed variation of Vanilla K-SFT and OP-SFT + KΔ-FT on held-out TriviaQA.

	Vanilla K-SFT		OP-SFT + KΔ-FT	
Seed	Acc.	AUROC	Acc.	AUROC
42	—	—	0.699	0.770
43	0.522	0.635	0.665	0.770
44	0.522	0.627	0.670	0.790
0	0.520	0.619	0.665	0.778
Mean $\pm$ SD	0.521 ± 0.001	0.627 ± 0.008	0.675 ± 0.016	0.777 ± 0.009

Table 17: FGPO seed variation from one shared Stage-1+2 initialization, evaluated end to end on held-out TriviaQA. Seed 1 is the original full-method run.

State	Test Acc.	Test AUROC
Shared Stage-1+2 initialization	0.699	0.770
FGPO, seed 1	0.692	0.809
FGPO, seed 43	0.677	0.810
FGPO, seed 44	0.672	0.815
FGPO mean $\pm$ SD	0.680 ± 0.010	0.811 ± 0.003

H.4 Functional FoK and accessible cues

These analyses measure behavior of the trained Qwen2.5-7B FoK channel.

Consistency at fixed incorrectness. From a 5,000-question TriviaQA pool, we sample 16 answers per question and retain the 1,149 questions where all 16 answers are wrong. Consistency is the fraction giving the most frequent incorrect answer. FoK rises with this consistency even though sampled correctness is zero (Table 18). The reported Spearman correlation is $\rho = 0.24$ ($p = 1.8 \times 10^{-16}$).

Table 18: Functional FoK among 1,149 questions with 16/16 incorrect sampled answers. Bin labels follow the evaluation grouping.

Answer consistency	Questions	Mean J
0–0.2	840	0.413
0.2–0.4	178	0.486
0.4–0.6	71	0.571
0.6–0.8	29	0.591
0.8–1.0	31	0.808

Accessible reasoning. We progressively reveal the generated reasoning trace on 440 MATH problems using the reasoning-prefix prompts of the preliminary study. Table 19 shows better discrimination as more of the trace becomes available. The mean-J gap is the difference between correct and incorrect solution groups.

Table 19: FoK discrimination with progressively revealed reasoning on 440 MATH problems.

Reasoning shown	AUROC	Mean-J gap
Question only	0.60	0.05
25%	0.66	0.09
50%	0.67	0.10
75%	0.74	0.15
Full reasoning	0.91	0.32

Context intervention. We prepend short contexts to a filtered 1,460-question subset of the same pool, removing contexts containing a gold answer. Table 20 reports FoK and answer accuracy; deltas are relative to the bare question. Related versus unrelated words raise mean J by 0.490 (paired Wilcoxon test on per-question J, reported $p = 5 \times 10^{-239}$). Incorrect related sentences increase J while slightly decreasing accuracy, showing that the judgment is not determined by correctness alone.

Table 20: Context interventions on 1,460 questions after filtering gold-answer-containing contexts.

Format	Condition	J	ΔJ	Acc.	ΔAcc.
—	None	0.599	—	0.408	—
Words	Unrelated	0.184	-0.415	0.424	$+0.016$
Words	Related	0.674	$+0.075$	0.478	$+0.070$
Sentences	Unrelated	0.298	-0.301	0.436	$+0.028$
Sentences	Correct related	0.677	$+0.078$	0.510	$+0.102$
Sentences	Incorrect related	0.772	$+0.173$	0.397	-0.011

H.5 Prompt sensitivity and alternative knowledge updates

Judgment templates. We repeat causal tracing on the same 40 Qwen2.5-7B samples under six judgment templates (Table 21). Recovery peaks stay at layers 20–21 of 28, with sensitive ranges within layers 16–23. This supports robustness across the tested templates. The templates cover both rephrasings and label changes; the sensitive range denotes the layers with strong recovery.

Table 21: Causal-tracing sensitivity to judgment templates on 40 shared samples. Layers out of 28.

Template	Peak layer	Sensitive range
Original	20	16–22
A: rephrased	21	17–22
B: third-person elicitation	21	16–21
C: numeric labels 1/2	21	18–23
D: letter labels A/B	21	18–21
E: labels can/cannot	20	18–20

Correction strategies. Table 22 compares knowledge-update strategies on Qwen2.5-7B-Instruct using 5,000 TriviaQA update questions and 1,000 held-out questions from a split different from the main tables. General capability is measured on the 1,319-question GSM8K test set [7]. Reptile uses first-order meta-updates [54]; SDFT uses the released self-distillation setup [56]. The lookahead variant penalizes deviation from Base after a one-step SFT update; the proximal variant adds an L2 constraint to Base. Average is the unweighted mean of the three reported accuracies and is given only when all three are available. Span replacement has the highest average but trails Base on GSM8K.

Table 22: Alternative knowledge-update strategies on a separate TriviaQA split. Unlikelihood was evaluated on TriviaQA only, so its GSM8K and Average cells show —.

Strategy	Update Acc.	Held-out Acc.	GSM8K	Average
Base	0.685	0.603	0.798	0.695
Vanilla SFT	0.992	0.452	0.368	0.604
Reptile meta-update	0.558	0.584	0.640	0.594
SDFT	0.529	0.565	0.367	0.487
Unlikelihood on wrong span	0.500	0.528	—	—
One-step lookahead KL	0.738	0.619	0.646	0.668
L2 proximal constraint	0.560	0.587	0.639	0.595
Replace wrong span	0.832	0.626	0.653	0.704

Supplementary assets. Table 23 records publisher-specified terms for additional datasets and the editing implementation. Dataset terms are distinguished from repository software licenses, particularly for Natural Questions.

Table 23: Additional assets and publisher-specified terms.

Asset	Role	License or terms
SelfAware	Zero-shot QA	CC BY-SA 4.0 (source)
Natural Questions	Transfer and replication	CC BY-SA 3.0 (data source)
WebQuestions	Zero-shot QA	CC BY 4.0 (source)
MATH	Reasoning evaluation	MIT (repository)
GSM8K	Capability evaluation	MIT (repository)
MEMIT	Editing implementation	MIT (repository)

I Detailed results for the main-text tables

I.1 Knowledge and judgment SFT

These tables retain all the error-rate columns of Table 1. Accuracy, illusion, and omission are percentages; their changes are percentage points. AUROC and its changes are unitless. Illusion and omission are conditional error rates, rather than fractions of all evaluated questions.

Table 24: Base values for the preliminary SFT comparison.

Model	Acc. (%)	AUROC	Illusion (%)	Omission (%)
Qwen2.5-0.5B	20	0.519	28	85
Qwen2.5-1.5B	35	0.529	56	38
Qwen2.5-3B	39	0.739	8	68
Qwen2.5-7B	53	0.724	51	20
Qwen2.5-14B	66	0.762	50	19
Qwen2.5-32B	68	0.819	62	8
LLaMA-3.1-8B	74	0.672	81	16

Table 25: Changes from Base under K-SFT, J-SFT, and sequential K→J SFT. Positive accuracy and AUROC changes and negative illusion and omission changes indicate improvement.

	K-SFT				J-SFT				K→J SFT			
Model	Acc.	AUC	Illu.	Om.	Acc.	AUC	Illu.	Om.	Acc.	AUC	Illu.	Om.
Qwen2.5-0.5B	+2	-0.016	-27	+13	-14	+0.213	-28	+15	+2	+0.211	-21	-10
Qwen2.5-1.5B	+6	-0.035	+6	-2	+2	+0.255	-28	-6	+5	+0.248	-21	-12
Qwen2.5-3B	+12	-0.220	+10	+18	+3	+0.038	+11	-29	+11	+0.051	+20	-41
Qwen2.5-7B	+7	-0.042	-15	+19	+2	+0.088	-9	-4	+7	+0.100	+6	-12
Qwen2.5-14B	+6	+0.004	+9	-4	+1	+0.030	+9	-9	+5	+0.068	+29	-17
Qwen2.5-32B	+4	-0.026	+11	0	0	+0.010	-10	+2	+4	+0.020	+3	-3
LLaMA-3.1-8B	+2	-0.038	-11	+9	+1	+0.045	+7	-11	+2	+0.063	+14	-14

I.2 Update-set stage-wise ablation

These are the update-set columns of the same runs used in Table 4.

Table 26: Update-set results for Base and the three cumulative training stages.

Stage	Acc.↑	AUROC↑	Illu.↓	Om.↓	Bias→ 0
Base	0.680	0.702	0.653	0.109	0.381
Stage 1: + OP-SFT	0.966	0.575	0.589	0.319	0.454
Stage 2: + KΔ-FT	0.967	0.803	0.350	0.190	0.270
Stage 3: + FGPO	0.961	0.817	0.250	0.240	0.245

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the FoK synchronization problem, the three empirical findings, the proposed method, and the experimental scope. The claims are tied to the reported factual-QA update setting and to the model families evaluated in the experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix A includes a separate Limitations and Future Work section discussing the factual-QA task scope, gradient-based update setting, out-of-scope knowledge editors and procedural tasks, backbone requirements, and the remaining FoK-performance gap.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper is empirical and methodological; it does not introduce formal theoretical results, theorems, or proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4.1 describes the core datasets, models, baselines, metrics, and update setting; Appendix C gives the detailed data construction, evaluation protocol, baseline protocol, and training hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is released at `Applied-Machine-Learning-Lab/NeurIPS2026_Know-thyself`, with scripts for the three training stages, the baselines and the evaluation, and a README with setup instructions. All datasets and model checkpoints are public and cited in the paper, and Appendix C describes the experimental protocol.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 specifies the core training and evaluation setting, including data splits, model families, baselines, metrics, and judge protocol; Appendix C gives the detailed implementation and optimizer settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Appendix H.3 reports individual runs, seed counts, means, and sample standard deviations for Vanilla K-SFT, OP-SFT + KΔ-FT, and FGPO. FGPO varies its stage-specific seed from a shared Stage-1+2 initialization, rather than independently rerunning the full pipeline. This quantifies training-seed variability; it is not a significance claim or a confidence interval, and not every supplementary experiment has seed replication.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix C reports the parallelism and main training and inference configuration, but exact wall-clock time and total compute were not logged for every preliminary and failed run.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses public benchmark data and open-weight LLMs for empirical evaluation and does not involve human subjects, private data, or deployment on users.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix B discusses the intended reliability benefit and the risk that improved answerability judgments may increase trust in systems that still require data-quality, monitoring, and domain-specific safety controls.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new pretrained model, scraped dataset, or deployment system with high misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: The paper credits the original assets and records publisher-specified terms for the core setup in Appendix C and additional assets in Table 23. However, the exact zsRE distribution, SDFT implementation revision, and all auxiliary checkpoint licenses are not identified in the available experiment record, so this is not a complete license and terms-of-use audit.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The released code (`Applied-Machine-Learning-Lab/NeurIPS2026_Know-thyself`) is documented in its README, which covers setup, the three-stage pipeline and the repository layout, and is distributed under the MIT License. The paper does not release a new dataset, benchmark, or pretrained model.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are the experimental subjects and the core model class studied by the method; Sections 3 and 4 describe how they are used.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.