# LEARNING TO SIGNAL UNCERTAINTY IN LLMS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large language models (LLMs) often produce confident yet incorrect answers, which can lead to risky failures in real-world applications. We study whether post-training can make a model's self-assessment explicit: when the model is uncertain, can it be trained to signal so within its own response? A central design question is where in the response this signal should be exposed: during reasoning, while the answer is still being formed, or at the end, once the answer has been produced. We study both. For end-of-reasoning self-assessment, we train the model to verbalize a confidence score for its response, with the aim of high confidence on correct answers and low confidence on incorrect ones. For during-reasoning self-assessment, we train the model to emit the marker `<uncertain>` as a warning, using final-answer correctness to supervise whether it appears. Across factual question-answering tasks, confidence training improves calibration, and generated warnings identify answers more likely to be incorrect. Both reports support selective retrieval to revise completed answers. We also examine internal representations through confidence-related geometry, confidence-digit probabilities across layers, and probes across model families. On shared responses, these analyses characterize associations between internal states, expressed confidence, and answer correctness.

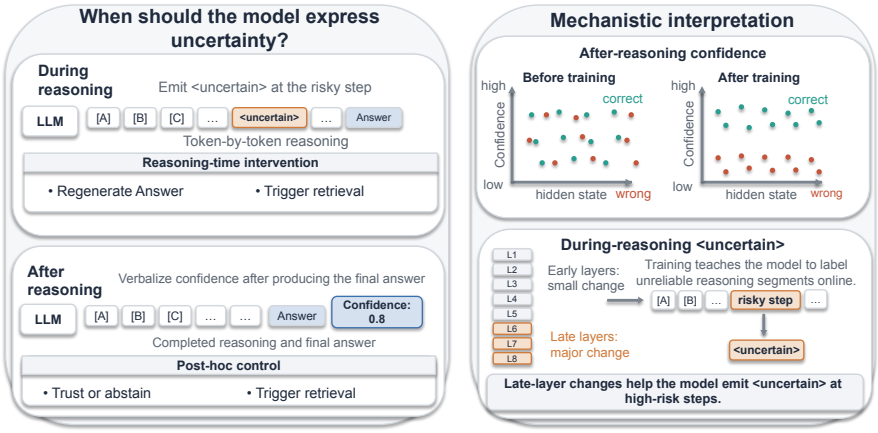



Figure 1: Two ways to report uncertainty. Our analyses examine associations between internal states, expressed confidence, and answer correctness. In our retrieval experiments, intervention follows the completed first response.

## 1 INTRODUCTION

Large language models (LLMs) often produce confidently wrong answers: they may invent facts that do not exist, or insist on answers to questions they cannot truly solve. Ideally, when the model is unable to answer a question correctly, it should signal so within its response, just as a person who is unsure about something would voice that hesitation rather than confidently guess. If the model can self-assess its reasoning quality at test time and communicate it clearly, a downstream controller can intervene by retrieving evidence, asking a clarifying question, invoking a tool, or abstaining.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Uncertainty quantification estimates the reliability of generated answers (He et al., 2026; Vashurin et al., 2025), while adaptive retrieval systems act on confidence scores, entropy, or response features (Jeong et al., 2024; Moskvoretskii et al., 2025; Su et al., 2024; Yao et al., 2025). Our focus is the report that the model itself produces: can post-training make uncertainty explicit in its response, so that a controller can use it without inferring it solely from token probabilities?

We study two complementary ways to report uncertainty. The confidence branch uses the correctness-plus-Brier reward of RLCR (Damani et al., 2026) to train a probability report after the answer. The marker branch learns an explicit `<uncertain>` warning during reasoning from final-answer correctness and warning presence. Generated uncertainty has precedents in linguistic calibration (Mielke et al., 2022) and learned `[IDK]` tokens (Cohen et al., 2024); reflection and backtracking tokens likewise expose signals for control (Asai et al., 2024; Zhang et al., 2025). Whereas TruthRL (Wei et al., 2026) rewards abstention, our marker permits continued answering and leaves assistance to a controller. We evaluate each reporting method against its own controls: whether training improves reporting, whether correctness-only rewards reproduce it, and whether the report helps select retrieval after a completed response. Representation analyses examine how these reports relate to information in the instruction-tuned base models and their trained counterparts.

Our main contributions are as follows.

- **Self-assessment as explicit exposure.** We frame LLM self-assessment as an exposure problem: training models to communicate their reasoning reliability within their own responses, making uncertainty explicit and actionable for downstream intervention.
- **Two complementary forms of uncertainty expression.** We train models to verbalize a confidence score after the final answer or emit an explicit `<uncertain>` warning during reasoning. Both use final-answer supervision to make uncertainty visible within the response.
- **Improved reliability and adaptive retrieval.** Across factual question-answering tasks, confidence training improves calibration, while generated warnings identify error-prone answers. Both reports support selective retrieval after the initial response.
- **Internal-representation analyses.** We examine confidence-related geometry, confidence-digit probabilities across layers, and reporting-position probes across model families. Comparisons on shared responses characterize associations between internal states, expressed confidence, and answer correctness in instruction-tuned base models and their trained counterparts.

## 2 Problem Setup and Notation

We study self-assessment expressed within the model's response. Given a question $x$ and reference answer $y^\star$, a model generates reasoning, an answer, and any uncertainty report as $z = (z_1, \ldots, z_T)$:

$$\pi_\theta(z \mid x) = \prod_{t=1}^{T} \pi_\theta(z_t \mid x, z_{<t}), \qquad \hat{y} = g(z). \tag{1}$$

Here $\theta$ denotes model parameters, $z_{<t}$ is the generated prefix, and $g$ extracts the answer $\hat{y}$. Its correctness indicator $Y$ supplies outcome supervision for the uncertainty report.

**Two reporting interfaces.** Confidence summarizes answer reliability with a generated scalar; a marker expresses a warning within the response:

$$c = g_c(z) \in [0,1], \qquad a_t = \mathbf{1}\{\text{a marker starts at position } t\}, \qquad E = \mathbf{1}\left\{\sum_{t=1}^{T} a_t > 0\right\}. \tag{2}$$

The parser $g_c$ reads the final field, e.g., `Confidence: 0.7`. This report differs from the probability of generating the answer string. A marker is the decoded `<uncertain>` string and may span multiple tokenizer tokens. When $E = 1$, its first position is $t_{\text{first}} = \min\{t : a_t = 1\}$. Both interfaces use final-answer supervision; no step-level error annotations are required.

**Outcome-supervised post-training.** Each interface has its own reward and training distribution. For $j \in \{\text{conf}, \text{mark}\}$, the objective is

$$J_j(\theta_j) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D}_j} \mathbb{E}_{z\sim\pi_{\theta_j}(\cdot|x)} \big[ r_j(z; x, y^\star) \big], \tag{3}$$

Here $\mathcal{D}_j$ is the training distribution and $r_j$ the reward defined in Sections 3–4. Separate Llama-3-8B-Instruct copies receive clipped policy-gradient updates with one response per prompt, raw-reward advantages, and no baseline or KL regularization. We use verl's GRPO trainer (Shao et al., 2024; Sheng et al., 2025); Appendix D gives implementation settings.

**Evaluation conventions.** Correctness depends on the matching rule. Let $Y_{\mathrm{EM}}$ indicate normalized exact match and $A, B$ denote distinct normalized token sets in $\hat{y}, y^\star$. Define

$$F_{\mathrm{set}} = \frac{2|A \cap B|}{|A| + |B|}, \qquad Y_\delta = \mathbf{1}\{Y_{\mathrm{EM}} = 1 \text{ or } F_{\mathrm{set}} \geq \delta\}, \tag{4}$$

with $F_{\mathrm{set}} = 0$ if either set is empty; reference aliases are matched separately. The main probability-quality check uses $Y_{0.5}$: EM equals one or F1 is at least $0.5$. Independent detection and six-task marker evaluation use type-aware matching or $F_{\mathrm{set}} > 0.5$. Sections 3–4 specify training labels; Appendix B.6 maps evaluation rules to populations.

## 3 Verbalized Confidence

The model ends its answer with a confidence value $c \in [0, 1]$, parsed from generated text. During training, $Y \in \{0, 1\}$ is determined by answer-type-aware matching: yes/no words, date components, or normalized exact match to a reference alias. Token-F1 relaxations are used only for evaluation. The reward is

$$r_{\mathrm{conf}}(z; x) = 2Yc - c^2 = Y - (c - Y)^2. \tag{5}$$

This is the correctness-plus-Brier reward used by Reinforcement Learning with Calibration Rewards (RLCR) (Damani et al., 2026). For a fixed answer distribution with correctness probability $q$, expected reward is $2qc - c^2$, maximized at $c = q$. This proper-scoring property rewards reporting the true probability in expectation (Gneiting & Raftery, 2007) and motivates the confidence objective. Our evaluation measures both answer quality and the calibration of the resulting probability reports.

**Measuring confidence quality.** Expected calibration error (ECE) (Guo et al., 2017) averages the absolute gaps between mean confidence and correctness within bins, weighted by bin size. The relaxed-correctness comparison uses ten equal-width bins and $Y_{\mathrm{rel}}$, including $c = 1$ in the last bin. Brier loss (Brier, 1950) averages $(c - Y)^2$. Confidence negative log-likelihood (NLL) is the correctness label's Bernoulli log loss, distinct from token-generation likelihood. Appendix G specifies the scoring rules; Table 3 and Table 9 report results. High-confidence errors also become less frequent in a separate factual-QA evaluation; Appendix B.7 reports the full confidence-band breakdown.

## 4 Reasoning-Time <UNCERTAIN> Marker

An explicit warning lets the model communicate uncertainty without abandoning its answer. We train it to emit `<uncertain>` during reasoning and finish with an `Answer:` line. Final-answer correctness supervises whether to warn, while the position of the marker records where that uncertainty is expressed in the response.

**Objective.** The reward makes silent errors more costly than warned errors while retaining a preference for correct answers. Let $m = \sum_t a_t$ count marker occurrences and $E = \mathbf{1}\{m > 0\}$. We assign

$$r_{\mathrm{mark}}(Y, E) = \begin{cases} 5.0 & Y = 1,\ E = 0, \\ 3.5 & Y = 1,\ E = 1, \\ 0.0 & Y = 0,\ E = 1, \\ -2.0 & Y = 0,\ E = 0. \end{cases} \tag{6}$$

The completed response supplies all supervision: correctness is determined by normalized answer matching, including yes/no and date matching, with a token-F1 fallback (Appendix D). No token-level uncertainty labels are used. We subtract $\min\{0.5 \max(m - 2, 0), 2.0\}$ to discourage repeated markers. The outcome ordering encourages selective warnings. A correct answer receives a lower reward when accompanied by a warning, whereas an incorrect answer receives a higher reward.

**Timing of the warning.** Warning timing describes when uncertainty becomes visible within a response, separately from how often the model warns. Figure 2 locates the first marker among responses that contain one, using the normalized position $t_{\text{first}}/T$ to account for different response lengths. Smaller values indicate earlier reporting. These conditional distributions complement the emission rates in Table 1, which count warnings across all responses.

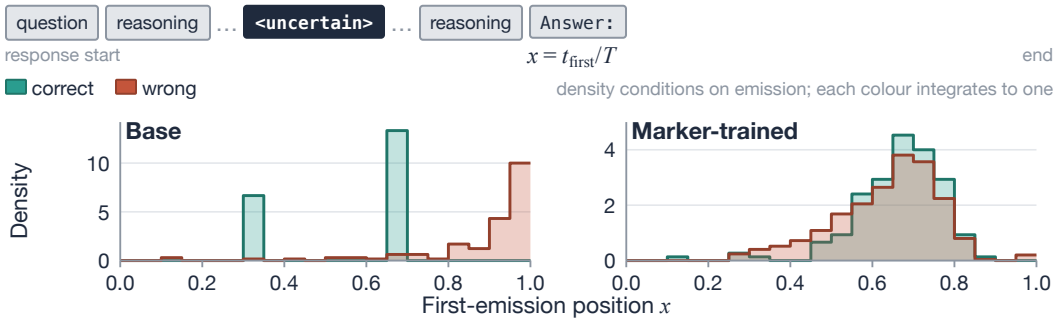



Figure 2: First-warning position $t_{\text{first}}/T$: first-marker token index divided by response length. Timing sample: 500 responses per model before grouping; densities condition on emission and correctness. Smaller ratios mean earlier warnings; density scales differ.

**Error coverage and precision.** Training exposes a larger share of errors while trading warning precision for coverage. On 1,100 development responses per model, warnings cover $455/581$ errors after training, compared with $355/799$ for the base model (Table 1). The higher coverage accompanies more frequent warnings, a smaller fraction of which accompany incorrect answers. Because the models generate their own answers, these results characterize their reporting behavior at the observed operating points. Section 5.2 separately tests warning informativeness on 2,433 independent answers shared by all detectors.

| Metric | Base | Trained |
|---|---|---|
| Emission rate | 33.6 | 59.1 |
| Error recall | 44.4 | 78.3 |
| Error precision | 95.9 | 70.0 |
| Accuracy, no marker | 39.2 | 72.0 |

Table 1: Raw-marker development behavior (%), 1,100 responses per model.

| Dataset | Base | Trained | Dataset | Base | Trained |
|---|---|---|---|---|---|
| 2Wiki | 12.4 | 25.6 | NQ | 21.6 | 40.8 |
| HotpotQA | 22.0 | 27.8 | TriviaQA | 36.2 | 56.0 |
| MuSiQue | 4.2 | 6.6 | IIRC | 9.6 | 14.4 |
| *Six-task mean* | | | | *17.7* | *28.5* |

Table 2: Marker-format accuracy (%), 500 questions per dataset.

**Answer quality across tasks.** The marker-trained model also improves the success of the complete answering pipeline across six factual-QA tasks. Mean accuracy rises from 17.7% to 28.5% in Table 2, which includes IIRC (Ferguson et al., 2020). Success requires both an extractable `Answer:` field and a correct answer; missing fields count as incorrect. The gains therefore combine answer-format adherence and factual correctness. Appendix Table 13 reports these components together.

## 5 Evaluation

We evaluate reporting quality and its usefulness for retrieval on downstream tasks. Calibration measures agreement between reported probabilities and correctness, while retrieval asks whether those reports help select additional evidence. Both interfaces start from Llama-3-8B-Instruct (Grattafiori et al., 2024) and are compared with their own baselines. Appendix B.6 specifies the response populations and correctness rules for each comparison.

### 5.1 Confidence Calibration

Calibration gains persist across correctness rules and probability-quality measures. Under $Y_{0.5}$, training reduces ECE from 52.5% to 4.9%, accompanied by lower Brier loss and confidence NLL

(Table 3). ECE also decreases under strict EM and $Y_{\mathrm{rel}}$. Agreement across these measures supports improved probability reporting beyond a single choice of correctness threshold or binning convention.

The largest behavioral change is in stated certainty, rather than answer correctness. For the evaluated checkpoint, mean confidence falls from $86.9\%$ to $40.3\%$, while correctness under $F_{\mathrm{set}} \geq 0.5$ changes from $34.5\%$ to $35.8\%$. The reports thus move closer to the observed frequency of correct answers. The accompanying Brier and NLL results assess the quality of individual reports, beyond this agreement of averages. The baseline comparison distinguishes adjusting confidence on fixed answers from changing the answering model itself. Global and adaptive temperature scaling (Guo et al., 2017; Xie et al., 2024) transform confidence reports on base-model answers; confidence-supervised fine-tuning (Kapoor et al., 2024; Luo et al., 2025) can change both answers and reports. Appendix Table 9 places answer quality alongside confidence reward to make this distinction explicit.

| Measure | Base | Trained |
|---|---|---|
| ECE (%), $Y_{0.5}$ (15 bins) | 52.5 | 4.9 |
| Brier loss | 0.504 | 0.166 |
| Confidence NLL | 4.987 | 0.498 |
| ECE (%), strict EM (10 bins) | 62.4 | 13.3 |
| ECE (%), $Y_{\mathrm{rel}}$ (10 bins) | 49.5 | 3.6 |

Table 3: Calibration on 498/500 parsed base/-trained answers. Main metrics use $Y_{0.5}$; ECE is a percentage.

| Detector | Trigger | Precision | Recall |
|---|---|---|---|
| Random reference | 65.6 | 67.2 | 65.6 |
| Raw marker | 65.6 | 81.8 | 79.8 |
| Marker + ridge | 56.5 | 86.5 | 72.8 |
| Marker + logistic | 47.4 | 89.6 | 63.2 |

Table 4: Detection on 2,433 shared test answers (%). Random matches the raw-marker budget; probe thresholds are fixed.

### 5.2 Warning-Based Error Detection

Raw warnings identify error-prone answers when every detector is evaluated on the same generated responses. The independent test contains 2,433 marker-model answers after excluding training/development overlaps, with correctness determined by type-aware matching or set-token F1 $> 0.5$. Warnings occur on $65.6\%$ of responses but cover $79.8\%$ of errors, exceeding the $65.6\%$ expected from random selection at that frequency. Their precision is $81.8\%$, compared with the random reference of $67.2\%$ (Table 4). Thus the warnings select errors beyond what their frequency alone would predict.

Filtering the emitted warnings offers a way to favor precision over coverage without changing the answer generator. The retrieval pipeline's layer-13 ridge probe, applied without refitting, raises precision to $86.5\%$ while retaining $72.8\%$ recall. A logistic probe fitted on training responses reaches $89.6\%$ precision at a lower trigger rate. These operating points give a controller different balances between requesting assistance and leaving errors unflagged. Appendix B.3 provides the data separation, layer comparisons, and thresholds.

### 5.3 Reward Ablations and Replication Across Base Models

Report-dependent rewards shape uncertainty reporting beyond answer-correctness supervision alone. Under correctness-only rewards, confidence ECE is $46.8 \pm 6.0\%$, compared with $7.2 \pm 1.2\%$ under the full reward, and marker emission nearly disappears (Table 5). The controls match each interface's prompts, data, initialization, and training-step budget, retaining the confidence parse penalty and marker answer-reward scale. These results identify direct report supervision as a key component of the learned behavior.

Repeated training supports the reporting improvements while showing variation in their magnitude. Marker recall is $68.5 \pm 5.2\%$ across repeated runs. For confidence, improved calibration accompanies lower mean answer accuracy than correctness-only training ($35.8 \pm 5.7\%$ versus $39.8 \pm 0.7\%$). The checkpoints used in the main evaluations achieve lower ECE and higher marker recall than any repeated run. These comparisons span changes in both seed and software stack. Appendix B.2 provides the full results and group-size-four controls.

**Reporting behavior across base models.** Low calibration error and selective warning emission also appear in the Mistral, Qwen, Gemma, and smaller Llama configurations shown in Table 5. The separately trained interfaces extend these reporting behaviors beyond Llama-3-8B. Each configuration is evaluated in a single run, and the metrics characterize calibration and warning operating points rather

| | Confidence | | | Marker | | |
|---|---|---|---|---|---|---|
| | Acc. | Mean $c$ | ECE | Emission | Prec. | Rec. |
| *Llama-3-8B, reward ablation* | | | | | | |
| Correctness only | $39.8 \pm 0.7$ | $86.6 \pm 5.7$ | $46.8 \pm 6.0$ | $0.0 \pm 0.1$ | – | $0.1 \pm 0.1$ |
| Full reward | $35.8 \pm 5.7$ | $34.5 \pm 9.7$ | $7.2 \pm 1.2$ | $51.8 \pm 8.6$ | $68.4 \pm 7.7$ | $68.5 \pm 5.2$ |
| *Other base models, full reward* | | | | | | |
| Mistral-7B | 20.6 | 21.6 | 8.2 | 49.2 | 99.8 | 72.6 |
| Qwen2.5-7B | 22.2 | 19.4 | 2.8 | 62.4 | 73.8 | 76.0 |
| Llama-3.2-3B | 32.3 | 31.6 | 1.6 | 68.2 | 72.7 | 80.9 |
| Qwen2.5-3B | 17.2 | 10.1 | 7.1 | 62.5 | 99.4 | 81.6 |
| Gemma-2-2B | 12.6 | 16.6 | 3.9 | 72.9 | 69.6 | 82.4 |

Table 5: Reward ablations and cross-model reporting results (%, $Y_{\text{rel}}$). Llama-3-8B rows report mean $\pm$ SD over three runs, while other rows report one run per interface. Confidence and marker columns describe separately trained models. For correctness-only markers, precision is undefined in two runs and based on one warning in the third.

than confidence-ranking quality. Appendix B.2 includes the protocols and two smaller configurations that produce near-constant reports.

### 5.4 Downstream Task: Selective Retrieval

Selective retrieval tests whether uncertainty reports help allocate additional evidence. We evaluate retrieval-augmented generation (RAG) (Lewis et al., 2020) on 500 questions each from HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), 2WikiMultihopQA (Ho et al., 2020), Natural Questions (Kwiatkowski et al., 2019), and TriviaQA (Joshi et al., 2017). Our trained reporting systems first answer without retrieval, then either keep that answer or request up to three BM25 passages (Robertson & Zaragoza, 2009) for a second answer from Llama-3.1-8B-Instruct. This decision follows the completed first response and permits one retrieval-assisted revision. All methods use the 2WikiMultihopQA index as a Wikipedia proxy for Natural Questions and TriviaQA.

Both interfaces support the same retrieve-or-keep decision through different report formats. The confidence gate retrieves when $c < \tau_c$; the marker gate retrieves when a marker occurs and its ridge-probe error score meets or exceeds $\tau_p$. We also evaluate raw marker emission without filtering. Appendix H gives the thresholds and retrieval settings. These experiments use the evaluated reporting checkpoints; Section 5.3 separately characterizes variation across training runs. Shared answer candidates isolate retrieval selection from changes in generation. For each question, this comparison fixes the first and retrieved answers, then varies only which answer the gate selects. Executed-system comparisons instead evaluate each complete answering pipeline. Keeping these comparisons distinct lets us assess both the value of the gate and the final quality of the system that uses it.

**The contribution of retrieval selection.** The learned gates outperform random retrieval at the same per-dataset budget. With shared answer candidates, their paired EM gains over random selection are $1.8$ points for confidence and $0.9$ for markers, with respective 95% intervals $[1.2, 2.3]$ and $[0.2, 1.6]$ (Table 6). Because both the answer candidates and retrieval budget are shared within each comparison, the differences reflect which questions receive evidence at these operating points. Marker-probe development overlaps the retrieval pool; the independently screened detector test in Section 5.2 evaluates warning informativeness separately. Retrieving indiscriminately can reduce answer quality, making selection consequential. The confidence gate reaches $31.9\%$ EM, compared with $29.1\%$ without retrieval and $27.7\%$ when retrieving for every question. The marker gate also exceeds Ret-All, but its gain over No-Ret is smaller: $0.8$ EM points with paired 95% CI $[0.0, 1.6]$. The evidence for marker selection is therefore clearer against random or indiscriminate retrieval than against keeping every first answer. Appendix H reports the full paired comparisons.

**Answer quality and retrieval use.** The complete-system comparison shows a trade-off between answer quality and retrieval frequency. The confidence gate obtains $31.8\%$ EM and $41.0\%$ F1 while retrieving on $45.5\%$ of questions (Table 7). DRAGIN (Su et al., 2024) attains the highest aggregate answer scores, $32.7\%$ EM and $43.2\%$ F1, with retrieval on $76.8\%$ of questions. Confidence gating

| System | Match | F1 | Ret. |
|---|---|---|---|
| Conf. No-Ret | 29.1 | 37.1 | 0.0 |
| Conf. Ret-All | 27.7 | 36.1 | 100.0 |
| Conf. gate | 31.9 | 41.1 | 45.5 |
| Conf. Random | 30.1 | 39.0 | 45.5 |
| Marker No-Ret | 28.0 | 34.7 | 0.0 |
| Marker Ret-All | 24.6 | 32.2 | 100.0 |
| Marker gate | 28.8 | 36.4 | 53.7 |
| Marker Random | 27.9 | 35.2 | 53.7 |

Table 6: Match uses EM. Random matches the gate's retrieval budget.

| System | Match | F1 | Ret. |
|---|---|---|---|
| SR-7B | 6.5 | 16.7 | 46.2 |
| SR-13B | 8.8 | 22.4 | 29.5 |
| Single-Ret | 24.8 | 34.3 | 100.0 |
| FLARE | 21.6 | 30.7 | 38.4 |
| DRAGIN | 32.7 | 43.2 | 76.8 |
| ADARAGUE | 11.0 | 15.5 | 43.8 |
| Raw marker | 28.5 | 36.0 | 61.4 |
| Conf. gate | 31.8 | 41.0 | 45.5 |
| Marker gate | 29.0 | 36.6 | 53.7 |

Table 7: Executed systems (%); Ret. is retrieval frequency.

uses evidence less frequently while giving up some answer quality. Retrieval frequency measures this evidence use, rather than latency or total generation cost.

Both trained gates exceed FLARE (Jiang et al., 2023) in answer quality, although their first-pass generators differ (Llama-3 versus Llama-3.1). Comparisons with the ADARAGUE output-classifier gate (Moskvoretskii et al., 2025), the official Self-RAG implementation (Asai et al., 2024) with its final-sentence substring scorer, and Single-Ret, which retrieves once from the question before answering, likewise evaluate complete pipelines. The shared-candidate comparison isolates retrieval selection from these system differences.

The quality–retrieval balance varies with the dataset and the available corpus. The confidence gate scores higher than DRAGIN on Natural Questions and TriviaQA, while DRAGIN scores higher on the three multi-hop datasets (Table 8). Natural Questions and TriviaQA use the proxy index, so these comparisons do not separate reporting quality from the usefulness of the available retrieval corpus. Appendix H gives full intervals and separates executed-system results from selection with shared answer candidates.

| System | HotpotQA | MuSiQue | 2Wiki | NQ | TriviaQA |
|---|---|---|---|---|---|
| SR-7B | 4.2 / 16.6 / 57.8 | 0.6 / 5.2 / 56.0 | 4.4 / 14.8 / 45.6 | 17.8 / 22.0 / 18.0 | 5.6 / 24.9 / 53.6 |
| SR-13B | 2.6 / 16.0 / 42.4 | 0.8 / 6.1 / 39.8 | 3.4 / 16.6 / 30.0 | 30.8 / 36.9 / 5.4 | 6.4 / 36.5 / 29.8 |
| Single-Ret | 29.2 / 41.4 / 100.0 | 10.2 / 18.8 / 100.0 | 23.6 / 30.9 / 100.0 | 21.0 / 32.7 / 100.0 | 40.2 / 47.7 / 100.0 |
| FLARE | 22.8 / 33.5 / 49.4 | 6.8 / 14.8 / 42.4 | 17.0 / 23.2 / 38.4 | 18.0 / 29.8 / 32.6 | 43.4 / 52.3 / 29.0 |
| DRAGIN | 34.6 / 48.4 / 87.6 | 17.0 / 27.1 / 87.0 | 34.4 / 43.9 / 85.2 | 23.8 / 35.9 / 70.4 | 53.6 / 60.6 / 54.0 |
| ADARAGUE | 13.6 / 21.8 / 57.0 | 4.6 / 8.4 / 53.8 | 13.6 / 16.7 / 57.4 | 9.2 / 13.5 / 21.8 | 14.0 / 17.2 / 29.0 |
| Raw marker | 25.6 / 37.2 / 67.4 | 7.6 / 13.9 / 94.2 | 27.8 / 32.6 / 59.2 | 28.4 / 37.4 / 52.0 | 53.0 / 58.9 / 34.0 |
| Conf. gate | 31.6 / 43.0 / 87.2 | 10.2 / 18.0 / 76.8 | 28.2 / 32.6 / 48.2 | 33.0 / 46.0 / 6.4 | 56.2 / 65.5 / 9.0 |
| Marker gate | 26.2 / 37.5 / 60.2 | 7.4 / 13.6 / 90.2 | 27.4 / 31.9 / 55.0 | 29.4 / 38.9 / 40.8 | 54.8 / 61.0 / 22.4 |

Table 8: Answer match / F1 / retrieval frequency (%). Match uses EM; Self-RAG uses substring matching. 2Wiki: 2WikiMultihopQA; NQ: Natural Questions.

## 6 Internal-Representation Analyses

The reporting behavior has observable counterparts in internal representations. We examine confidence-related geometry and layer-wise readouts in Llama-3-8B, then use cross-family probes to separate information about expressed confidence from information about answer correctness. These analyses describe associations in sampled reporting-position states, characterizing the internal states that accompany the generated reports.

**Confidence-related geometry.** Expressed confidence is organized within the trained model's final-state projection. In Figure 3, low- and high-confidence responses occupy different regions, with intermediate values tracing a progression between them. Both models process the same trained responses, and both panels use the confidence parsed from those responses for coloring. Principal component analysis (Jolliffe & Cadima, 2016) is fitted separately to each model's final-layer states at

the confidence readout position. The within-panel association motivates the subsequent probes of how well these states predict the reported confidence.

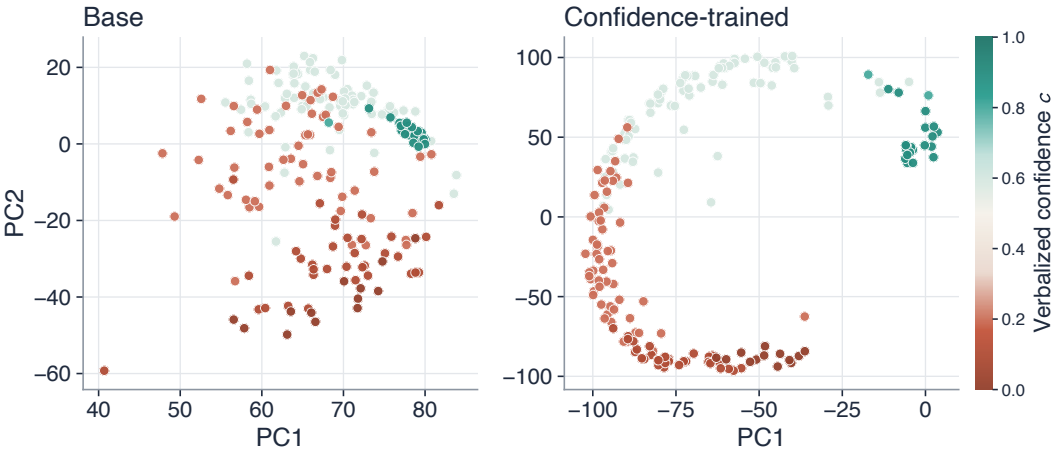



Figure 3: Final-state PCA on 200 shared trained responses. Panels are fitted independently and colored by the trained-model confidence.

**Confidence-related readouts across layers.** On shared incorrect responses, the trained model's later-layer readouts favor lower confidence digits. We inspect this pattern with the logit lens (nostalgebraist, 2020), which projects intermediate states through the model's output head. At the position predicting the fractional confidence digit, we apply final normalization, condition on the ten single-digit tokens, and group their mass into Low (0–3), Mid (4–6), and High (7–9). Figure 4 shows substantial High mass for the base model on wrong responses, whereas the trained model increasingly favors Low. This layer-wise pattern complements the final-state geometry by showing how the digit readout varies with depth. Appendix C.2 specifies the transformation, including its treatment of the final layer, and smoothing.

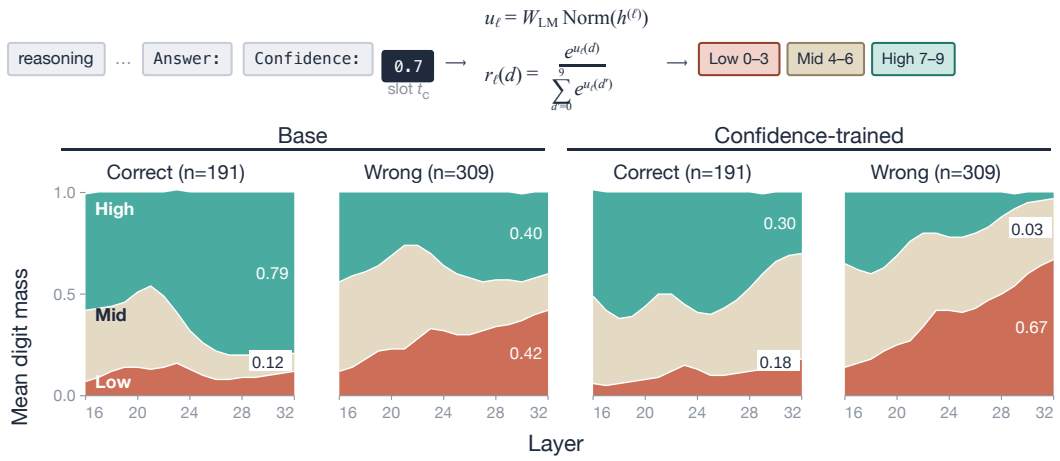



Figure 4: Smoothed digit-conditional mass by layer. The procedure of visualization is outlined in Appendix Eq. 7. Both models process the same trained responses: 191 correct and 309 wrong.

**Reporting-position representations across families.** Correctness and expressed confidence show different patterns of linear readability. In the Llama-3.2-3B, Qwen2.5-3B, and Gemma-2-2B samples, confidence-position correctness scores are similar before and after training, while a ridge probe predicts expressed confidence more accurately from trained states (Appendix Table 18). Marker-position correctness scores are also higher in the trained samples. These observations distinguish two questions that a single confidence-colored projection cannot resolve: whether a state predicts what the model reports, and whether it predicts that the answer is correct.

The probe comparisons characterize information available at the report itself. Each correctness probe uses 58–60 responses; within a configuration, base and trained models process the same responses, and cross-validation is grouped by question. States are read after the reporting token is consumed. Appendix C.4 provides the position rules, shuffled-label controls, and complete results, including the smaller configurations with near-constant reports. A complementary token-level analysis shows that prediction changes extend beyond reporting positions. Confidence digits and marker spans have elevated per-position KL, while the longer reasoning spans account for most of the total change. Appendix C.3 distinguishes per-position divergence from accumulated change.

## 7 Related Work

**From confidence scores to generated warnings.** Elicitation and post-training offer complementary ways to obtain confidence reports. Tian et al. (2023) find that verbalized confidence from models trained with human feedback can be better calibrated than their conditional token probabilities. Linguistic calibration trains expressions of uncertainty to reflect answer correctness (Mielke et al., 2022). Verbal confidence can also be learned through supervision (Lin et al., 2022; Xu et al., 2024), tokenized Brier loss (Li et al., 2025b), the correctness-plus-Brier reward of RLCR used here (Damani et al., 2026), or logarithmic rewards (Bani-Harouni et al., 2026). DCPO (Ma et al., 2026) separates reasoning and calibration objectives.

**Uncertainty within a response.** CURE (Liu & Wang, 2026) calibrates individual claims; RLCM (Wang et al., 2026) rewards confidence margins on intermediate completions. Cohen et al. (2024) train an explicit `[IDK]` token. R-Tuning (Zhang et al., 2024) constructs refusal-aware instruction data from the gap between model knowledge and training questions; TruthRL (Wei et al., 2026) learns abstention from outcome-dependent rewards. These works connect uncertainty to what a model chooses to say, while our outcome-trained warning accompanies a continued answer.

**Reporting and retrieval.** Retrieval can use confidence (Tao et al., 2024), reflection tokens (Asai et al., 2024), or generation-time signals (Jiang et al., 2023; Su et al., 2024). Moskvoretskii et al. (2025) compare uncertainty-based gates with adaptive pipelines. SKR (Wang et al., 2023) uses self-knowledge for retrieval, and Probing-RAG (Baek et al., 2025) uses internal probes. Semantic uncertainty (Kuhn et al., 2023) aggregates sampled answers by meaning, providing another source of reliability estimates. We train response-level reports separately from retrieval actions, so report learning and evidence selection can be examined individually.

**Internal representations of uncertainty.** Hidden states encode truth-related information (Burns et al., 2023; Marks & Tegmark, 2024), enabling error detection (Orgad et al., 2025) and semantic-entropy estimation (Kossen et al., 2024). The tuned lens (Belrose et al., 2023) learns an affine readout at each layer to decode intermediate predictions.

## 8 Conclusion

Outcome-supervised training makes uncertainty explicit through calibrated confidence after an answer or a warning during reasoning. Confidence quality improves across the tested correctness rules and probability losses; trained warnings concentrate errors beyond the expectation for matched-budget random selection. Both reports can drive a controller that selectively requests evidence after the initial response. Internal-representation analyses describe associations between final-layer states and reported confidence. Complementary prediction analyses show elevated changes at reporting positions, though these changes are not confined to them.

**Limitations.** The evaluations cover factual QA and one marker format, and the evaluated checkpoints lie outside the repeated-run ranges on the reporting metrics discussed in Section 5.3. Comparisons with agentic tool-use training (Jin et al., 2025) require matched compute and interaction budgets. Coding and long-form generation remain outside this evaluation.

### AI USE STATEMENT

Generative AI tools were used to assist with paper polishing and the implementation of analysis code. The authors reviewed all AI-assisted experimental artifacts, tests, and compiled outputs, and take responsibility for the final content of this work.

### REPRODUCIBILITY STATEMENT

Appendix D records training and evaluation settings, Appendix D.1 describes baseline implementations, and Appendix G defines the confidence metrics. The reporting evaluations specify correctness rules, parsed denominators, and confusion counts. Appendix C describes the internal-representation analyses and token-level prediction comparisons. Appendix B.2 reports repeated-training controls, and Appendix H provides paired retrieval intervals.

## REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.11511.

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

Ingeol Baek, Hwan Chang, Byeongjeong Kim, Jimin Lee, and Hwanhee Lee. Probing-RAG: Self-probing to guide language models in selective document retrieval. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3287–3304, 2025.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. Linguistic calibration of long-form generations. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. arXiv:2404.00474.

David Bani-Harouni, Chantal Pellegrini, Paul Stangel, Ege Özsoy, Kamilia Zaripova, Nassir Navab, and Matthias Keicher. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2503.02623.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

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. arXiv:2212.03827.

Yu-Neng Chuang, Prathusha Kameswara Sarma, Parikshit Gopalan, John Boccio, Sara Bolouki, Xia Hu, and Helen Zhou. Learning to route LLMs with confidence tokens. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10859–10878, 2025. arXiv:2410.13284.

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.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training LMs to reason about their uncertainty. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2507.16806.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. LM-Polygraph: Uncertainty estimation for language models.

In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 446–461. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-demo.41.

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.

James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. IIRC: A dataset of incomplete information reading comprehension questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 1137–1147, 2020. doi: 10.18653/v1/2020.emnlp-main.86.

Yichao Fu, Xuewei Wang, Hao Zhang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2508.15260.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. arXiv:2501.12948.

Jianfeng He, Linlin Yu, Changbin Li, Runing Yang, Fanglan Chen, Kangshuo Li, Min Zhang, Shuo Lei, Xuchao Zhang, Mohammad Beigi, et al. Survey of uncertainty estimation in LLMs: Sources, methods, applications, and challenges. *Information Fusion*, 130:104057, 2026. doi: 10.1016/j.inffus.2025.104057.

John Hewitt, Robert Geirhos, and Been Kim. Position: We can't understand AI using our existing vocabulary. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 81556–81572, 2025. arXiv:2502.07586.

John Hewitt, Oyvind Tafjord, Robert Geirhos, and Been Kim. Neologism learning for controllability and self-verbalization. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.08506.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Xinmeng Huang, Shuo Li, Mengxin Yu, Matteo Sesia, Hamed Hassani, Insup Lee, Osbert Bastani, and Edgar Dobriban. Uncertainty in language models: Assessment through rank-calibration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 284–312, 2024.

Chaeyun Jang, Moonseok Choi, Yegon Kim, Hyungi Lee, and Juho Lee. Verbalized confidence triggers self-verification: Emergent behavior without explicit reasoning supervision. *arXiv preprint arXiv:2506.03723*, 2025.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7036–7050, 2024. doi: 10.18653/v1/2024.naacl-long.389.

Ziwei Ji, Delong Chen, Etsuko Ishii, Samuel Cahyawijaya, Yejin Bang, Bryan Wilie, and Pascale Fung. LLM internal states reveal hallucination risk faced with a query. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 88–104, 2024.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, 2023. doi: 10.18653/v1/2023.emnlp-main.495.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A*, 374(2065):20150202, 2016. doi: 10.1098/rsta.2015.0202.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017.

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

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

Amita Kamath, Robin Jia, and Percy Liang. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5684–5696, 2020. doi: 10.18653/v1/2020.acl-main.503.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew G Wilson. Large language models must be taught to know what they don't know. *Advances in Neural Information Processing Systems*, 37: 85932–85972, 2024.

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J. Bell. AbstentionBench: Reasoning LLMs fail on unanswerable questions. *arXiv preprint arXiv:2506.09038*, 2025.

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

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2302.09664.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. Taming overconfidence in LLMs: Reward calibration in RLHF. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.09724.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Pengyi Li, Matvey Skripkin, Alexander Zubrey, Andrey Kuznetsov, and Ivan Oseledets. Confidence is all you need: Few-shot RL fine-tuning of language models. *arXiv preprint arXiv:2506.06395*, 2025a.

Yibo Li, Miao Xiong, Jiaying Wu, and Bryan Hooi. ConfTuner: Training large language models to express their confidence verbally. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. arXiv:2508.18847.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. arXiv:2205.14334.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*, 2024. arXiv:2305.19187.

Jingyu Liu, Jiaen Lin, and Yong Liu. How much can RAG help the reasoning of LLM? *arXiv preprint arXiv:2410.02338*, 2024.

Xin Liu and Lu Wang. Think through uncertainty: Improving long-form generation factuality via reasoning calibration. *arXiv preprint arXiv:2604.12046*, 2026.

Beier Luo, Shuoyuan Wang, Sharon Li, and Hongxin Wei. Your pre-trained LLM is secretly an unsupervised confidence calibrator. In *Advances in Neural Information Processing Systems*, volume 38, 2025. arXiv:2505.16690.

Xinxi Lyu, Michael Duan, Rulin Shao, Pang Wei Koh, and Sewon Min. Frustratingly simple retrieval improves challenging, reasoning-intensive benchmarks. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2507.01297.

Zhengzhao Ma, Xueru Wen, Boxi Cao, Yaojie Lu, Hongyu Lin, Jinglin Yang, Min He, Xianpei Han, and Le Sun. Decoupling reasoning and confidence: Resurrecting calibration in reinforcement learning from verifiable rewards. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. arXiv:2603.09117.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, 2023. doi: 10.18653/v1/2023.acl-long.546.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9004–9017, 2023. doi: 10.18653/v1/2023.emnlp-main.557.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *Proceedings of the First Conference on Language Modeling (COLM)*, 2024. arXiv:2310.06824.

Zhiting Mei, Christina Zhang, Tenny Yin, Justin Lidard, Ola Sho, and Anirudha Majumdar. Reasoning about uncertainty: Do reasoning models know when they don't know? In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 3408–3458, 2026. doi: 10.18653/v1/2026.findings-eacl.178. arXiv:2506.18183.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872, 2022. doi: 10.1162/tacl_a_00494.

Viktor Moskvoretskii, Maria Marina, Mikhail Salnikov, Nikolay Ivanov, Sergey Pletenev, Daria Galimzianova, Nikita Krayko, Vasily Konovalov, Irina Nikishina, and Alexander Panchenko. Adaptive retrieval without self-knowledge? bringing uncertainty back home. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6355–6384, 2025. arXiv:2501.12835.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352, 2023.

Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. When do LLMs need retrieval augmentation? mitigating LLMs' overconfidence helps retrieval augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11375–11388. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.675.

nostalgebraist. Interpreting GPT: The logit lens. LessWrong, August 31, 2020.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.02707.

Sophie Ostmeier, Brian Axelrod, Maya Varma, Asad Aali, Yabin Zhang, Magdalini Paschali, Sanmi Koyejo, Curtis Langlotz, and Akshay Chaudhari. Attention head entropy of LLMs predicts answer correctness. *arXiv preprint arXiv:2602.13699*, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Qwen, An Yang, Baosong Yang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019.

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

Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, and Luke Zettlemoyer. ReasonIR: Training retrievers for reasoning tasks. In *Proceedings of the Second Conference on Language Modeling (COLM)*, 2025. arXiv:2504.20595.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys '25)*, 2025. doi: 10.1145/3689031.3696075. arXiv:2409.19256.

Linxin Song, Taiwei Shi, and Jieyu Zhao. The hallucination tax of reinforcement finetuning. *arXiv preprint arXiv:2505.13988*, 2025.

Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. LACIE: Listener-aware finetuning for confidence calibration in large language models. *arXiv preprint arXiv:2405.21028*, 2024.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12991–13013, 2024. doi: 10.18653/v1/2024.acl-long.702.

Shuchang Tao, Liuyi Yao, Hanxing Ding, Yuexiang Xie, Qi Cao, Fei Sun, Jinyang Gao, Huawei Shen, and Bolin Ding. When to trust LLMs: Aligning confidence with response quality. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5984–5996. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.357. arXiv:2404.17287.

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*, 2023. arXiv:2305.14975.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, et al. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248, 2025.

Liaoyaqi Wang, Chunsheng Zuo, William Jurayj, Benjamin Van Durme, and Anqi Liu. Process supervision of confidence margin for calibrated LLM reasoning. *arXiv preprint arXiv:2604.23333*, 2026.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. arXiv:2506.01939.

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*, pp. 10303–10315, 2023. doi: 10.18653/v1/2023.findings-emnlp.691.

Ziyu Wang and Christopher C. Holmes. On subjective uncertainty quantification and calibration in natural language generation. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 258 of *Proceedings of Machine Learning Research*, pp. 3799–3807, 2025. arXiv:2406.05213.

Zhepei Wei, Xiao Yang, Kai Sun, Jiaqi Wang, Rulin Shao, Jingxiang Chen, Mohammad Kachuee, Teja Gollapudi, Yiwei Liao, Nicolas Scheffer, Rakesh Wanga, Anuj Kumar, Yu Meng, Wen-tau Yih, and Xin Luna Dong. TruthRL: Incentivizing truthful LLMs via reinforcement learning. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. arXiv:2509.25760.

Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang. Know your limits: A survey of abstention in large language models. *Transactions of the Association for Computational Linguistics*, 2024. arXiv:2407.18418.

Jiayun Wu, Jiashuo Liu, Zhiyuan Zeng, Tianyang Zhan, Tianle Cai, and Wenhao Huang. Mitigating LLM hallucination via behaviorally calibrated reinforcement learning. *arXiv preprint arXiv:2512.19920*, 2025.

Johnathan Xie, Annie S Chen, Yoonho Lee, Eric Mitchell, and Chelsea Finn. Calibrating language models with adaptive temperature scaling. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18128–18138, 2024.

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 (ICLR)*, 2024. arXiv:2306.13063.

Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaoze Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. SaySelf: Teaching LLMs to express confidence with self-reflective rationales. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024. arXiv:2405.20974.

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. On verbalized confidence scores for LLMs. *arXiv preprint arXiv:2412.14737*, 2024a.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. Alignment for honesty. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. arXiv:2312.07000.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Liu Weichuan, Lei Hou, and Juanzi Li. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27022–27043, 2025.

Gal Yona, Roee Aharoni, and Mor Geva. Can large language models faithfully express their intrinsic uncertainty in words? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7752–7764, 2024.

Dongkeun Yoon, Seungone Kim, Sohee Yang, Sunkyoung Kim, Soyeon Kim, Yongil Kim, Eunbi Choi, Yireun Kim, and Minjoon Seo. Reasoning models better express their confidence. In *Advances in Neural Information Processing Systems*, volume 38, 2025. arXiv:2505.14489.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, volume 38, 2025. arXiv:2504.13837.

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)*, 2024. arXiv:2311.09677.

Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R Tomz, Christopher D Manning, and Weiyan Shi. Verbalized sampling: How to mitigate mode collapse and unlock LLM diversity. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. arXiv:2510.01171.

Yiming Zhang, Jianfeng Chi, Hailey Nguyen, Kartikeya Upasani, Daniel M Bikel, Jason Weston, and Eric Michael Smith. Backtracking improves generation safety. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2409.14586.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2505.19590.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, et al. The path not taken: RLVR provably learns off the principals. *arXiv preprint arXiv:2511.08567*, 2025.

# Appendix

**A Extended Related Work** **18**

**B Supplementary Quantitative Results** **20**

B.1 Confidence Calibration . . . . . 20

B.2 Report-Reward Controls and Training Variability . . . . . 20

B.3 Error Detection on Shared Test Answers . . . . . 21

B.4 Additional Results for Reasoning-Time Signaling . . . . . 21

B.5 Prompt-Side Attention-Entropy Pilot . . . . . 22

B.6 Evaluation Populations and Reporting Conventions . . . . . 23

B.7 Confidence Reports Across Factual-QA Datasets . . . . . 24

**C Internal-Representation Analysis Methods** **24**

C.1 Final-State Geometry . . . . . 24

C.2 Layer-Wise Confidence Readout . . . . . 24

C.3 Token-Level Distribution Changes . . . . . 25

C.4 Reporting-Position Probes Across Model Families . . . . . 26

**D Experimental Setup** **27**

D.1 Baseline Implementation Details . . . . . 28

**E Illustrative Marker Responses** **29**

E.1 Four-Way Examples for the <uncertain> Marker . . . . . 29

**F Broader Impact Statement** **30**

**G Evaluation Metrics and Signal Timing** **30**

**H Retrieval Protocol and Detailed Results** **31**

# A Extended Related Work

**Uncertainty estimation and calibration in language models.** A large body of work studies how to estimate whether a model's generated answer is reliable. Classical calibration methods evaluate whether predicted probabilities match empirical correctness, with temperature scaling serving as a simple and widely used post-hoc recalibration method (Guo et al., 2017). For language models, uncertainty is harder to define because outputs are free-form sequences rather than fixed-class predictions. Recent surveys and benchmarks organize LLM uncertainty estimation into likelihood-based, sampling-based, semantic, verbalized, and hybrid methods (He et al., 2026; Vashurin et al., 2025). LM-Polygraph provides a unified framework for implementing and comparing such estimators (Fadeeva et al., 2023). Semantic uncertainty methods aggregate generations by meaning rather than surface form, showing that uncertainty in natural language should often be measured over semantic equivalence classes rather than exact strings (Kuhn et al., 2023). Other work develops uncertainty quantification for black-box LLMs (Lin et al., 2024), evaluates rank-calibration in language models (Huang et al., 2024), and studies subjective uncertainty and calibration in natural language generation (Wang & Holmes, 2025). SelfCheckGPT (Manakul et al., 2023) detects hallucinations through consistency across sampled responses. Our work is complementary to this line: rather than only post-hoc estimation, we train the model to expose its self-assessment explicitly within the response itself.

**Verbalized confidence and model self-knowledge.** Several papers ask whether language models know when they are likely to be correct. Linguistic calibration trains conversational expressions of uncertainty to better reflect correctness (Mielke et al., 2022). Early work showed that models can be trained to express uncertainty in words and that such expressions can become more calibrated than raw model behavior (Lin et al., 2022), and that verbalized confidence elicited from RLHF-tuned models can be better calibrated than token probabilities (Tian et al., 2023). Related work studies whether models can judge the truth of their own answers through P(True)-style prompting and self-evaluation (Kadavath et al., 2022). Fine-tuning approaches teach calibrated verbal confidence through self-reflective rationales (Xu et al., 2024), listener-aware preference optimization (Stengel-Eskin et al., 2024), decision-theoretic linguistic calibration of long-form text (Band et al., 2024), or tokenized proper-scoring losses (Li et al., 2025b). Empirical studies of confidence elicitation show that LLMs can produce verbalized confidence scores, but these scores are often sensitive to prompting and may remain overconfident without additional training (Xiong et al., 2024; Yang et al., 2024a; Yona et al., 2024). Other recent work argues that pretrained LLMs already contain useful confidence signals that can be extracted or recalibrated without full retraining (Luo et al., 2025), while supervised approaches teach models to better distinguish what they know from what they do not know (Kapoor et al., 2024). Recent studies also investigate how verbalized confidence affects generation diversity and self-verification behavior (Zhang et al., 2026; Jang et al., 2025; Fu et al., 2026). Our confidence interface builds on this work through a reward combining answer correctness and a proper scoring rule (Brier, 1950; Gneiting & Raftery, 2007). We evaluate it alongside an outcome-trained marker that reports uncertainty during reasoning.

**Reward-based post-training for calibrated behavior.** Post-training with reinforcement learning has become a standard way to shape LLM behavior (Schulman et al., 2017; Ouyang et al., 2022; Shao et al., 2024; Guo et al., 2025). Recent work shows that reward design can strongly affect reasoning style, calibration, and hallucination behavior (Li et al., 2025a; Wu et al., 2025; Leng et al., 2025). RLCR (Damani et al., 2026) combines answer correctness with a Brier penalty, the reward form used in our confidence branch; Rewarding Doubt (Bani-Harouni et al., 2026) instead uses a logarithmic scoring reward. CONQORD (Tao et al., 2024) combines response-quality and confidence-ordering rewards and applies its confidence estimates to retrieval decisions. DCPO (Ma et al., 2026) decouples reasoning and calibration objectives to address conflicting optimization signals. Related work reports that reinforcement fine-tuning can increase hallucinations (Song et al., 2025) and analyzes how correctness-only scoring can encourage guessing (Kalai et al., 2025). We build on calibration-aware training by studying an after-answer confidence report alongside a reasoning-time marker, evaluating the reliability of each report and its use as a retrieval trigger. Our confidence reward penalizes higher-confidence errors more strongly and rewards higher confidence on correct answers. This connects to analyses of post-training changes in token-level behavior and trajectory selection (Wang et al., 2025; Zhao et al., 2026; Zhu et al., 2025), including work examining whether

verifiable-reward training reweights reasoning paths already supported by the base model (Yue et al., 2025).

**Granularity of uncertainty supervision.** Calibration can also target units within a response. CURE pairs atomic claims with explicit confidence estimates and trains claim-level calibration for long-form generation (Liu & Wang, 2026). RLCM uses a hidden-state confidence probe and a margin reward over intermediate reasoning prefixes (Wang et al., 2026). Our marker learns a generated warning from final-answer outcomes and emission behavior, providing a binary report that a separate controller can use to request assistance.

**Selective prediction, abstention, and retrieval decisions.** Uncertainty estimates are useful only when connected to a downstream decision, such as abstaining, asking for help, or retrieving evidence. Selective QA under domain shift learns when to abstain using a correctness calibrator (Kamath et al., 2020). Abstention has been trained through refusal-aware instruction tuning and honesty alignment (Zhang et al., 2024; Yang et al., 2024b) and surveyed by Wen et al. (2024); reasoning-oriented fine-tuning can degrade it (Kirichenko et al., 2025). TruthRL (Wei et al., 2026) trains abstention with rewards distinguishing correct answers, hallucinations, and abstentions. Adaptive retrieval systems use uncertainty or complexity estimates to decide when retrieval is worthwhile; Mallen et al. (2023) study popularity-gated retrieval for long-tail facts. Ni et al. (2024) explicitly address overconfidence in deciding when retrieval is needed. Adaptive-RAG learns to adapt retrieval behavior based on question complexity (Jeong et al., 2024); FLARE triggers retrieval during generation based on low-confidence token predictions (Jiang et al., 2023); DRAGIN retrieves based on real-time information needs (Su et al., 2024); and ADARAGUE reintroduces uncertainty features into adaptive retrieval control (Moskvoretskii et al., 2025). SKR uses self-knowledge to select retrieval (Wang et al., 2023). SEAKR similarly studies self-aware knowledge retrieval for adaptive RAG (Yao et al., 2025). Other work studies how much retrieval can improve reasoning (Liu et al., 2024), improves retrievers for reasoning-intensive tasks (Shao et al., 2025), and shows that simple retrieval can help challenging reasoning benchmarks (Lyu et al., 2026). Probing-RAG uses model-internal probing to guide selective retrieval (Baek et al., 2025), which is especially close to our hidden-state probe analysis, and RL-trained search agents interleave retrieval calls with reasoning end to end (Jin et al., 2025). Our marker is trained from answer outcomes and emission behavior, without retrieval-action or passage-critique labels. It supplies a report that a separate controller can read; the current application is retrieval after a completed first response. This makes report quality and downstream action distinct evaluation targets.

**Learned tokens, control markers, and reasoning-time intervention.** A related line of work studies whether special tokens or learned markers can package complex behaviors into compact controllable symbols. Gist tokens compress prompts into learned latent handles (Mu et al., 2023), and recent work on neologism learning studies how new tokens can acquire controllable meanings through training (Hewitt et al., 2026; 2025). In reasoning and tool-use settings, models can also learn to emit action-like markers that control retrieval, critique, or generation behavior. Self-RAG trains models to retrieve, generate, and critique using reflection tokens (Asai et al., 2024), while backtracking tokens allow models to mark unsafe or undesirable generation paths and revise them (Zhang et al., 2025). Confidence-token routing further shows that explicit learned tokens can support model selection or rejection decisions (Chuang et al., 2025). Reasoning models also revise their expressed confidence during chain-of-thought, although they can remain overconfident when they do not know (Yoon et al., 2025; Mei et al., 2026). Our `<uncertain>` marker exposes possible answer failure before the response ends. Its presence supplies a binary warning at a generated reasoning position, while its supervision comes from the final-answer outcome.

**Internal states and mechanistic evidence for self-assessment.** Several works suggest that a model's hidden states can contain information about truthfulness, correctness, or hallucination risk that is not always faithfully expressed in the output. Latent-knowledge work shows that internal representations can encode truth-related information without direct supervision (Burns et al., 2023), with approximately linear true/false structure (Marks & Tegmark, 2024). Other studies find that hidden states can reveal when a model is lying or likely to hallucinate (Azaria & Mitchell, 2023; Ji et al., 2024), that truthfulness information concentrates in specific answer tokens while probes transfer poorly across datasets (Orgad et al., 2025), and that semantic entropy and its probes can

detect hallucination risk from generations or internal activations (Farquhar et al., 2024; Kossen et al., 2024). Mechanistic and layerwise analyses of post-training further suggest that fine-tuning can affect different layers and token positions unevenly (Wang et al., 2025). The logit lens (nostalgebraist, 2020) reads intermediate states through the output head; the tuned lens (Belrose et al., 2023) learns an affine probe for each layer. These approaches motivate a distinction between information available in internal states and information exposed in the generated response. Our behavioral evaluation concerns the latter, while the supplementary probe study examines whether activations can filter emitted warnings.

**Positioning of this work.** Prior work establishes methods for training confidence, abstaining, retrieving evidence, and reading internal states. We connect these directions through two complementary output interfaces: numerical confidence after answering and a binary warning during reasoning. Our study combines calibration and error-detection evaluation with selective retrieval driven by either interface after a completed first response. The warning objective separates answer correctness from whether risk is reported during generation. A correct answer may carry a warning, and an incorrect answer may remain silent; distinguishing these outcomes enables learned reporting while generation continues. Numerical confidence supplies a graded alternative. We evaluate each report's reliability, internal readouts, and use by a retrieval controller.

# B Supplementary Quantitative Results

## B.1 Confidence Calibration

Table 3 evaluates the same confidence predictions under strict exact match and two token-overlap thresholds. With set-token F1 $\geq 0.5$, mean confidence minus empirical correctness decreases from $+0.523$ to $+0.045$. Parse rates are $498/500$ for the base model and $500/500$ for the confidence-trained model. Brier loss and negative log-likelihood assess individual confidence reports, while ECE summarizes bin-level agreement. Appendix G specifies their common inclusion rules.

Table 9 compares answer quality and confidence reward on 500 held-out prompts. EM and $\mathrm{Acc}_{\mathrm{rel}}$ use parsed responses under $Y_{\mathrm{EM}}$ and $Y_{\mathrm{rel}}$. Reward averages $2Y_{\mathrm{rel}}c - c^2$ over all prompts, assigning $-0.5$ to parse failures.

| Method | Parsed $N$ | EM (%) | $\mathrm{Acc}_{\mathrm{rel}}$ (%) | Reward |
|---|---|---|---|---|
| Base | 498 | 24.5 | 37.3 | $-0.108$ |
| Global TS | 498 | 24.5 | 37.3 | $+0.116$ |
| ATS | 498 | 24.5 | 37.3 | $+0.123$ |
| SFT-Conf | 498 | 21.1 | 33.5 | $+0.083$ |
| SFT-KWDK | 499 | 22.4 | 34.5 | $+0.105$ |
| Confidence-trained | 500 | 27.4 | 38.2 | $+0.210$ |

Table 9: Answer quality and confidence reward. Accuracies are percentages; $N$ counts parsed responses.

## B.2 Report-Reward Controls and Training Variability

The confidence control rewards $Y$ for a parsed response and assigns $-0.5$ for a parse failure. The marker control assigns $+5$ to a correct answer and $-2$ to an incorrect answer, independently of marker emission; emission and repetition terms are removed. Tables 10 and 11 summarize repeated training and the reward controls. The $n = 4$ controls sample four responses per prompt at the same optimizer-step budget; the remaining runs sample one.

**Variation across training runs.** Tables 10 and 11 summarize three runs per reward and interface as mean $\pm$ SD. Across report-reward runs, confidence ECE ranges from $5.8$ to $8.1\%$, marker recall from $63.7$ to $74.0\%$, and emission from $41.9$ to $57.9\%$. The study varies both seed and software stack: the evaluated checkpoints were trained on H100 GPUs, and the repeated runs on Blackwell GPUs with a different software stack. All use two GPUs with tensor parallelism of two.

| Reward | $n$ | Accuracy | ECE |
|---|---|---|---|
| Report | 1 | $35.8 \pm 5.7$ | $7.2 \pm 1.2$ |
| Correctness only | 1 | $39.8 \pm 0.7$ | $46.8 \pm 6.0$ |
| Report | 4 | 36.2 | 18.8 |

Table 10: Confidence reward controls (%) on 500 prompts under $Y_{\text{rel}}$. Both $n = 1$ rows give mean $\pm$ SD over three runs; $n = 4$ gives one run. Accuracy conditions on parsed responses; ECE uses ten endpoint-inclusive bins.

| Reward | $n$ | Accuracy | Emission | Precision | Recall |
|---|---|---|---|---|---|
| Report | 1 | $48.9 \pm 1.6$ | $51.8 \pm 8.6$ | $68.4 \pm 7.7$ | $68.5 \pm 5.2$ |
| Correctness only | 1 | $48.1 \pm 1.8$ | $0.0 \pm 0.1$ | – | $0.1 \pm 0.1$ |
| Report | 4 | 54.4 | 46.9 | 69.8 | 71.7 |

Table 11: Marker reward controls (%) on 1,100 development responses under $Y_{\text{rel}}$. Both $n = 1$ rows give mean $\pm$ SD over three runs; $n = 4$ gives one run. Correctness-only precision is undefined in two runs and based on one warning in the third. $n$ is responses per prompt.

Sampling four responses per prompt changes both the sampled group and the advantage calculation. Its single confidence run has ECE $18.8\%$, compared with $7.2 \pm 1.2\%$ for single-response runs, and mean confidence $27.0\%$ against accuracy $36.2\%$. This operating point is underconfident on average; increasing group size alone does not ensure better calibration under this training budget.

**Replication across base models.** Table 5 applies the reporting recipe of each interface to five further base models. Marker precision is omitted for correctness-only controls because emission is absent or extremely rare. Confidence runs are scored on the same evaluation population as Table 3; marker runs report raw emission at the same checkpoint step as the main marker model. Two smaller base models, Llama-3.2-1B and Qwen2.5-1.5B, are absent from the table: both emit markers on every evaluated response and report near-zero confidence. These are observed outcomes of this recipe; aggregate accuracy alone does not explain which configurations produce constant reports. Appendix C.4 reports their probe scores. Under $Y_{\text{rel}}$, confidence accuracy is $11.9\%$ for Llama-3.2-1B and $7.0\%$ for Qwen2.5-1.5B; their separately trained marker models have accuracy $28.4\%$ and $29.5\%$, respectively.

### B.3 Error Detection on Shared Test Answers

Every detector in Table 12 sees the same marker-model answers. The evaluation retains 2,433 questions after matching question identifiers against the marker training and development sets. Correctness uses type-aware answer matching or set-token F1 $> 0.5$. Non-emitting responses remain in the evaluation population and, when incorrect, in the recall denominator; they cannot trigger the span-based gates. Precision conditions on triggered responses.

Logistic probes are fitted on training responses, with regularization selected by development AUROC. The reported logistic rows trigger at decision score $\geq 0$; additional budget-based operating points use training-score quantiles. The retrieval pipeline's layer-13 ridge probe is applied without refitting, at threshold $-0.329$. AUROC among emitters uses the 1,573 responses with a locatable marker span. Each raw-marker and probe row reports its fixed operating point and resulting trigger frequency. The random reference in Table 4 matches raw emission frequency.

### B.4 Additional Results for Reasoning-Time Signaling

Table 13 reports both answer accuracy and response-format behavior on six factual-QA datasets. Producing an explicit final-answer line becomes more consistent after training. Accuracy measures the complete answering pipeline, including production of an extractable final answer; the Answer Line columns separately quantify that formatting behavior.

| Detector | Trigger | Precision | Recall | $\text{Acc}_{\neg R}$ | $\text{AUROC}_{\text{emit}}$ |
|---|---|---|---|---|---|
| Raw marker | 65.6 | 81.8 | 79.8 | 60.6 | – |
| Marker + ridge | 56.5 | 86.5 | 72.8 | 58.0 | 80.9 |
| Marker + logistic (L8) | 46.9 | 89.0 | 62.1 | 52.1 | 75.4 |
| Marker + logistic | 47.4 | 89.6 | 63.2 | 53.1 | 78.8 |
| Marker + logistic (L16) | 47.1 | 89.8 | 63.0 | 53.0 | 77.5 |

Table 12: Detectors on identical test answers ($N = 2,433$). Values are percentages. Trigger is the fraction flagged; precision and recall concern errors; $\text{Acc}_{\neg R}$ is accuracy among untriggered answers. $\text{AUROC}_{\text{emit}}$ ranks error scores among emitters with a locatable span. The ridge probe is transferred without refitting; both unqualified probe rows use layer 13. Other layers are labeled explicitly.

| | Accuracy | | | Answer Line | | Emit Rate | | Wrong / Correct + Emit | |
|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | **Base** | **Trained** | **Δ** | **Base** | **Trained** | **Base** | **Trained** | **W+E** | **C+E** |
| 2WikiMultihopQA | 12.4 | 25.6 | +13.2 | 51.4 | 100.0 | 43.2 | 59.8 | 43.2 / 55.4 | 0.0 / 4.4 |
| HotpotQA | 22.0 | 27.8 | +5.8 | 62.8 | 99.6 | 36.6 | 70.8 | 35.8 / 59.0 | 0.8 / 11.8 |
| MuSiQue | 4.2 | 6.6 | +2.4 | 50.8 | 100.0 | 49.8 | 94.6 | 49.8 / 89.0 | 0.0 / 5.6 |
| NQ | 21.6 | 40.8 | +19.2 | 77.0 | 100.0 | 23.6 | 56.6 | 23.0 / 39.6 | 0.6 / 17.0 |
| TriviaQA | 36.2 | 56.0 | +19.8 | 72.0 | 100.0 | 31.8 | 44.2 | 30.2 / 31.6 | 1.6 / 12.6 |
| IIRC | 9.6 | 14.4 | +4.8 | 39.4 | 100.0 | 46.2 | 87.2 | 45.8 / 77.6 | 0.4 / 9.6 |
| Six-task mean | 17.7 | 28.5 | +10.9 | – | – | – | – | – | – |

Table 13: Base and marker-trained models on six factual-QA datasets, with 500 questions per dataset. "Answer Line" is the fraction of responses containing an explicit final answer line. "Emit Rate" is the overall `<uncertain>` emission rate. Each entry in the last two columns is ***Base / Trained***: the fraction of all examples that are wrong and emit (W+E) or correct and emit (C+E); these two joint frequencies sum to Emit Rate. All values are percentages; Δ is trained minus base accuracy in percentage points, computed before rounding.

### B.5 Prompt-Side Attention-Entropy Pilot

A supplementary pilot evaluates a passive error predictor using the attention-feature approach of Head Entropy (Ostmeier et al., 2026). It uses later Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct model pairs (Grattafiori et al., 2024; Qwen et al., 2025), separately from the original Llama-3-8B marker experiment.

Features come from the prompt alone. For attention probabilities $a_k$ over key positions $k$, order-two Rényi entropy is $H_2 = -\log \sum_k a_k^2$. Entropies are averaged over query positions and concatenated across layers and heads. Logistic regression predicts final-answer error, defined by failure of both answer matching and the pilot evaluator's token-F1 $\geq 0.5$ rule. Each model contributes 160 responses. Scores are stratified, shuffled five-fold out-of-fold predictions: each response is scored by a classifier fitted on the other folds. Feature standardization is fitted within each fold, followed by L1 logistic regression with balanced class weights, the liblinear solver, $C = 1$.

| Model pair | Base AUROC | Marker-trained AUROC |
|---|---|---|
| Llama-3.1 | 0.708 | 0.654 |
| Qwen2.5 | 0.749 | 0.617 |

Table 14: Prompt-side error prediction, $N = 160$ responses per model. AUROC is the area under the receiver operating characteristic curve, with incorrect answers as positives. Values describe out-of-fold discrimination within each model's sample, using only features available before response generation. The full-set marker results evaluate generated warnings separately.

The scores indicate predictive information in prompt-side attention features in these samples. Folds are stratified by response label, rather than grouped by question ID, so this is a pilot within the sampled responses. It complements the generated-warning evaluation by examining information available before an answer is produced.

### B.6 Evaluation Populations and Reporting Conventions

Each interface is evaluated for its reporting task, with a specified prompt, response population, and correctness rule. Table 15 identifies the population and denominator used for each result, so that answer quality, calibration, and warning coverage can be interpreted within their respective experiments.

| Study | Evaluation population | Scoring and purpose |
|---|---|---|
| Confidence calibration | 500 prompts; 498 base and 500 trained parsed responses | Strict EM and relaxed accuracy; confidence metrics use the stated correctness rule. Reward includes all 500 prompts. |
| Warning timing (Figure 2) | 500 responses per model before grouping | First-marker position divided by response length, among emitting responses and grouped by answer correctness. |
| Raw-marker development | 1,100 development responses per model; 799 base and 581 trained errors | $Y_{\mathrm{rel}}$ labels; precision conditions on warnings, recall on errors. Compares answer-and-warning behavior. |
| Independent detector test | 2,433 shared marker-model answers, excluding training/development overlaps | Type-aware matching or set-token F1 $> 0.5$; raw markers and probe filters are evaluated on identical answers. |
| Confidence factual QA | Five datasets with 500 prompts each, plus 100 IIRC prompts | 2,588 base and 2,600 trained parsed responses; accuracy and confidence bands use EM or set-token F1 $\geq 0.5$ against any reference alias. |
| Marker factual QA | Six datasets with 500 prompts each | Type-aware matching or set-token F1 $> 0.5$; answer-line frequency is reported alongside accuracy. The mean weights six datasets equally. |
| Selective retrieval | Five datasets with 500 questions each | Final-answer match/F1 and actual retrieval frequency; Self-RAG uses substring matching, other methods normalized EM. Equal dataset weights. |

Table 15: Populations and metric denominators. A dataset name identifies a benchmark, while the study also specifies the reporting prompt, generated responses, scorer, and aggregation. "Base" refers to the baseline within the corresponding study.

**Reading accuracy across studies.** Table 8 reports final-answer match/F1 after the retrieval decision and supplies the per-dataset results summarized in Table 7. The marker-format evaluation in Table 2 measures answer extraction and correctness without retrieval. Table 16 records the response counts for the confidence-band analysis, using EM or set-token F1 $\geq 0.5$. These tables report different response populations; comparisons between strict and relaxed scoring of the same responses appear in Table 3.

**Confidence targets and aggregation.** On the same 500-prompt confidence test, trained-model ECE is $3.6\%$ under $Y_{\mathrm{rel}}$ and $13.3\%$ under strict EM, both with ten bins. The $4.9\%$ result uses set-token F1 $\geq 0.5$ and fifteen bins. These calibration checks apply different labels to the same predictions. Confidence reward is reported under $Y_{\mathrm{rel}}$ throughout Table 9 and its discussion. The separate factual-QA analysis includes IIRC with 100 prompts; its confidence bands pool wrong answers (Table 17), with dataset-level response counts in Table 16.

**Probe operating points.** The base-answer detector and trained-answer detector use different response populations. The independent trained-answer evaluation applies the fixed layer-13 retrieval probe and separately fitted logistic probes to the same 2,433 answers (Appendix B.3). ADARAGUE in the retrieval table denotes an output-feature classifier gate, not this hidden-state detector.

### B.7 Confidence Reports Across Factual-QA Datasets

This confidence-reporting evaluation contains 2,600 prompts: 500 each from 2WikiMultihopQA, HotpotQA, MuSiQue, Natural Questions, and TriviaQA, and 100 from IIRC. Table 16 records the parsed-response counts underlying the confidence-band analysis in Table 17. An answer is correct if normalized exact match succeeds or set-token F1 is at least $0.5$ against any reference alias.

| | | Parsed responses | |
|---|---|---|---|
| Dataset | Prompts | Base | Trained |
| 2WikiMultihopQA | 500 | 499 | 500 |
| HotpotQA | 500 | 498 | 500 |
| MuSiQue | 500 | 496 | 500 |
| Natural Questions | 500 | 499 | 500 |
| TriviaQA | 500 | 498 | 500 |
| IIRC | 100 | 98 | 100 |
| All six datasets | 2,600 | 2,588 | 2,600 |

Table 16: Response counts for the factual-QA confidence-band analysis. The base/trained populations contain 916/989 correct answers and 1,672/1,611 errors under the stated labels. IIRC has 100 prompts here, versus 500 in the marker-format evaluation.

**Confidence when an answer is wrong.** The same parsed populations contain 1,672 base errors and 1,611 trained errors. Table 17 pools these errors across all six datasets. High-confidence errors become less frequent: reports above $0.7$ account for $1{,}482/1{,}672 = 88.6\%$ of base errors and $63/1{,}611 = 3.9\%$ of trained errors. Mean confidence among errors decreases from $0.837$ to $0.306$. These statistics measure confidence conditional on an incorrect answer, complementing the calibration checks in Table 3.

| Confidence band | Base errors | Trained errors |
|---|---|---|
| $c > 0.7$ | 1,482 (88.6%) | 63 (3.9%) |
| $0.5 < c \leq 0.7$ | 63 (3.8%) | 500 (31.0%) |
| $0.3 < c \leq 0.5$ | 3 (0.2%) | 1 (0.1%) |
| $0.1 < c \leq 0.3$ | 45 (2.7%) | 512 (31.8%) |
| $c \leq 0.1$ | 79 (4.7%) | 535 (33.2%) |
| All errors | 1,672 (100%) | 1,611 (100%) |
| Mean confidence on errors | 0.837 | 0.306 |

Table 17: Pooled confidence bands among incorrect parsed responses from the six-dataset confidence evaluation. Counts are summed across datasets before percentages are calculated; rounding may prevent displayed percentages from summing to exactly 100.

## C Internal-Representation Analysis Methods

### C.1 Final-State Geometry

We project the final-layer hidden state at the confidence readout position onto two principal components. PCA is fitted independently for the base and confidence-trained panels, with 200 response representations per model. Both models process the same trained responses, and both panels are colored by the confidence parsed from those trained responses. The base panel therefore does not depict the base model's generated confidence distribution. A color gradient indicates that confidence-related variation is organized in the displayed subspace. Each panel has its own centered coordinate system, so the analysis examines confidence-related geometry within each panel.

### C.2 Layer-Wise Confidence Readout

The logit-lens readout (nostalgebraist, 2020) uses the state at the final period in the last literal `Confidence: 0.` prefix to predict the following digit. For layer $\ell$, the diagnostic applies the

model's final normalization and language-model head to the returned state. If $u_\ell(d)$ is the resulting logit for single-digit token $d$, the digit-conditional readout is

$$r_\ell(d) = \frac{\exp u_\ell(d)}{\sum_{d'=0}^{9} \exp u_\ell(d')}, \qquad d \in \{0, \ldots, 9\}. \tag{7}$$

Each response is normalized over these ten tokens before averaging within outcome groups. We smooth along the layer axis with a Gaussian filter ($\sigma = 1$) and sum into Low=$\{0, 1, 2, 3\}$, Mid=$\{4, 5, 6\}$, and High=$\{7, 8, 9\}$. Figure 5 gives all two-decimal cells underlying the stacked view. In particular, the base wrong-answer endpoint is Low/Mid/High=$0.42/0.18/0.40$; the trained endpoint is $0.67/0.30/0.03$.

Both models process the same question and trained response, with correctness groups defined by that response (191 correct and 309 wrong). The input concatenates the user text and response. The prefix selector restricts this diagnostic to matching decimal reports; it does not include multi-character digit tokens or treat a `Confidence: 1.0` prefix as a High-bin report. Consequently, the displayed quantity is conditional digit mass, while Tables 3 and 9 evaluate the complete parsed scalar reports.

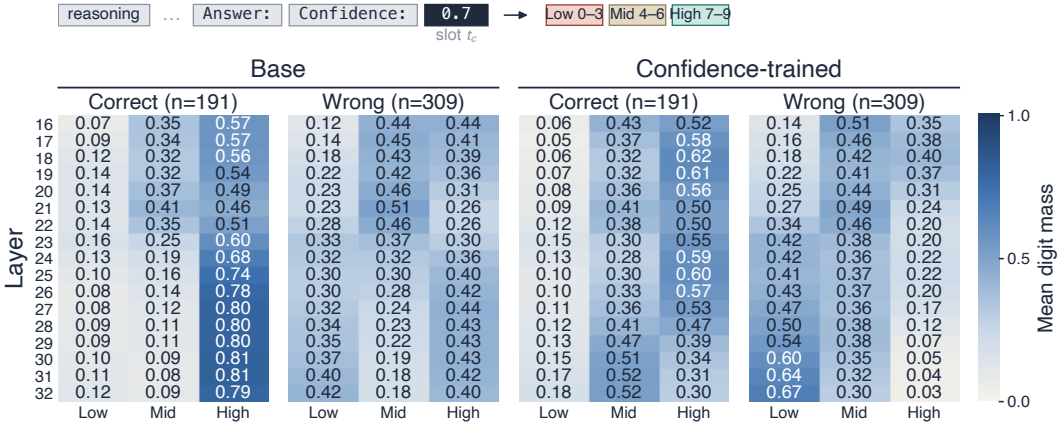



Figure 5: Confidence-digit readout cells for layers 16–32, using a shared 0–1 scale. Columns show Low/Mid/High mass and rows show layers; outcome groups follow the trained responses shared across models. Values are rounded to two decimals.

**Readout transformation.** The readout transformation is $W_{\mathrm{LM}}\,\mathrm{Norm}(h^{(\ell)})$, where $W_{\mathrm{LM}}$ is the language-model output-head matrix, $h^{(\ell)}$ is the returned layer state, and Norm is the decoder's final normalization. The same transformation is applied to all displayed layers. The returned final state has already passed through the model's final normalization, so this last readout composes that normalization with a second application. The final-layer absolute values therefore describe this transformation rather than the model's native output probabilities. Digit conditioning and layer smoothing define the displayed readout; behavioral calibration uses the scalar confidence parsed from the generated response.

### C.3 Token-Level Distribution Changes

The KL analysis uses 200 traces per interface. Base and trained distributions are evaluated on the same token prefix within each trace, using $\mathrm{KL}(p_{\mathrm{base}} \| p_{\mathrm{trained}})$. A row labeled by token $t$ describes the prediction of token $t + 1$, after $t$ has been consumed. This aligns the diagnostic with changes within and following reporting spans.

The labeler identifies the first marker span and the confidence label/digits by token subsequence. Nearby windows contain up to five positions before or after the report. Pre-context replaces reasoning/other labels; post-context replaces other labels and, for the marker branch, reasoning labels. Answer positions are separately labeled when matched. The residual Other category can include reasoning or answer text following a marker, so it groups unmatched positions rather than a single semantic role. Padding and prompt positions are excluded from assistant-token aggregates.

For vocabulary $\mathcal{V}$, per-position divergence is $D_t = \sum_{v \in \mathcal{V}} p_t(v) \log[p_t(v)/q_t(v)]$, where $p_t, q_t$ are the base and trained next-token distributions. Figure 6 groups these values by consumed-token role. Its left panels show per-position distributions; the right panels compare each role's share of summed KL with its share of assistant-token positions. Report spans can carry high per-position KL, while reasoning contributes the largest total KL share; an excess over position share is not unique to reporting spans.

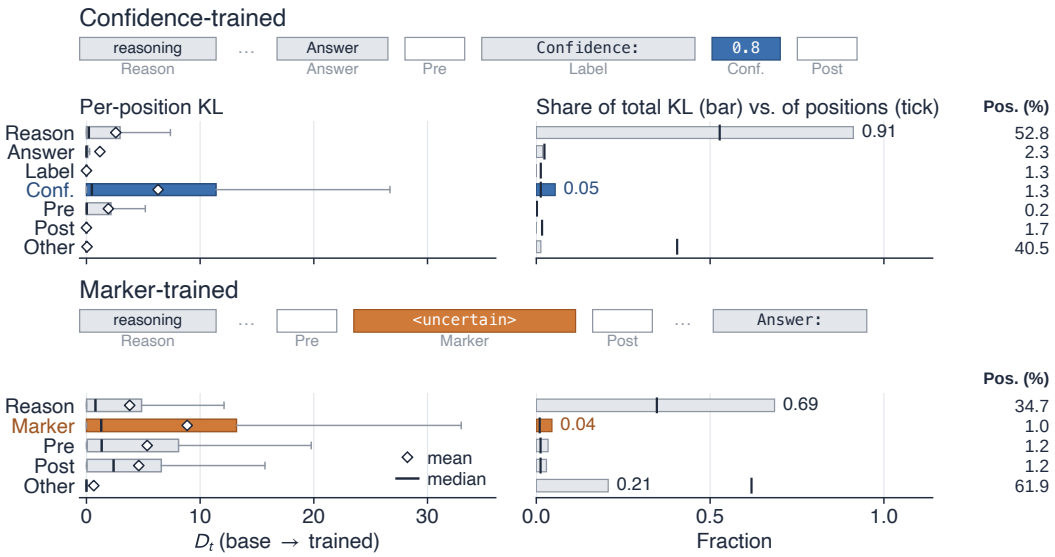



Figure 6: Where next-token predictions change, by consumed-token role, on 200 traces per interface. Each block pairs one interface's token strip, which fixes what the categories cover, with the two measures on a shared category axis. Left: per-position KL, with boxes giving quartiles and median, diamonds means, and whiskers within 1.5 interquartile ranges; outliers hidden. Right: each category's share of total KL (bars) against its share of assistant-token positions (ticks, with percentages at right); a bar extending beyond its tick indicates above-average KL per position. Pre/Post: up to five positions before/after the report.

### C.4 Reporting-Position Probes Across Model Families

The cross-family analysis covers five configurations from three families: Llama-3.2-1B/3B, Qwen2.5-1.5B/3B, and Gemma-2-2B. Table 18 reports all five. In the two smallest the generated reports are near-constant, so the confidence target is degenerate and marker-probe scores show no consistent improvement; we report them here to describe the coverage of the evaluation. Each configuration contributes one base/trained pair, read from the same capture so that both members process the same generated responses; responses differ across pairs, so rows compare within a pair rather than across models.

**Representation and position.** Each probe input is the final-layer state after final normalization, taken at the reporting position: the state after consuming the first confidence digit for the confidence interface, or the first marker token for the marker interface, with a fallback position when the span is absent. These are reporting-position representations. They do not measure whether an error is identifiable before the report is emitted.

**Targets and probes.** The confidence probe is ridge regression on the expressed confidence; the correctness probe is logistic regression on final-answer correctness. Correctness here uses normalized exact or bidirectional containment matching on the extracted `Answer:` line, which differs from the relaxed rule $Y_{\text{rel}}$ used for marker development, so these scores are not comparable with Table 1. Both probes use `GroupKFold` cross-validation grouped by question with five splits, sweeping the regularization strength within each fold. Captures record 60 responses each, taken in file order rather than by random or stratified sampling; invalid labels reduce some sample sizes, so each reported score carries its own $n$. Shuffled-label nulls use five independent label permutations.

| | Confidence position Correctness AUROC | | | Reported confidence $R^2$ | | Marker position Correctness AUROC | | |
|---|---|---|---|---|---|---|---|---|
| Configuration | Base | Trained | Shuf. | Base | Trained | Base | Trained | Shuf. |
| Llama-3.2-3B | 0.89 | 0.89 | 0.55 | 0.39 | 0.62 | 0.59 | 0.67 | 0.51 |
| Qwen2.5-3B | 0.96 | 0.96 | 0.55 | 0.81 | 0.88 | 0.79 | 0.82 | 0.44 |
| Gemma-2-2B | 0.98 | 0.98 | 0.35 | 0.90 | 0.96 | 0.72 | 0.78 | 0.50 |
| Llama-3.2-1B | 0.48 | 0.45 | 0.57 | – | – | 0.34 | 0.52 | 0.50 |
| Qwen2.5-1.5B | – | – | – | – | – | 0.73 | 0.72 | 0.51 |

Table 18: Mean cross-validated probe scores at the reporting position. Shuf. columns are nulls over five independent label permutations. Correctness probes use 58–60 responses per cell; the share of correct responses is $0.07$–$0.30$ at the confidence position and $0.31$–$0.53$ at the marker position, so AUROC should be read against the null in the same row. Entries marked – are degenerate in that sample rather than zero: Qwen2.5-1.5B has almost no correct responses at the confidence position, and both smallest configurations report near-constant confidence.

**Scope limits.** Three points bound how far these scores can be read. The Gemma-2-2B confidence labels in this sample take only two values, so its high $R^2$ indicates that this report target is readable, not that a continuous confidence manifold has been demonstrated. Correctness labels are imbalanced in some captures, so AUROC should be read against the shuffled null in the same row. Finally, predicting the expressed report and predicting correctness are separate measurements; neither establishes a causal mechanism or error localization at individual reasoning steps.

## D Experimental Setup

Both reporting models initialize from `meta-llama/Meta-Llama-3-8B-Instruct`. Training uses single-sample policy gradients with PPO clipping and no baseline, implemented through the GRPO trainer in `verl`/HybridFlow (Sheng et al., 2025) with vLLM rollouts. Table 19 gives the configurations.

| Setting | Confidence | Marker |
|---|---|---|
| Training / held-out prompts | 5,500 / 500 | 11,000 / 1,100 |
| Prompt / response limit | 1,024 / 512 | 1,024 / 512 |
| Train batch / update minibatch | 256 / 64 | 256 / 64 |
| Actor microbatch per GPU | 16 | 16 |
| Actor learning rate | $10^{-6}$ | $10^{-6}$ |
| KL regularization | Off | Off |
| Optimizer steps | 150 | 126 |
| Responses per prompt | 1 | 1 |
| Save / validation frequency (steps) | 30 / 5 | 50 / 5 |
| GPUs / tensor parallelism | 2 / 2 | 2 / 2 |

Table 19: Training settings for the evaluated reporting models. Prompt and response limits are token counts; save and validation frequencies are measured in optimizer steps.

Both runs enable gradient checkpointing and padding removal, allocate $0.5$ of GPU memory to rollouts, and keep actor parameters and optimizer states on device. Rollout temperature is $1.0$ for both models; top-$p$ is $1.0$ for confidence and $0.95$ for the evaluated marker model. Neither a KL loss nor a KL reward penalty is applied. With one response per prompt, the outcome advantage equals the raw reward; updates use the clipped policy objective without a group baseline.

**Training the reporting interfaces.** The confidence data pool combines HotpotQA, MuSiQue, 2WikiMultihopQA, and Natural Questions: 5,500 questions are used for training and 500 are held out from the pooled training splits. Marker training adds TriviaQA, using 11,000 training questions and 1,100 development questions drawn from the respective benchmark splits. Each interface uses its own prompts and training schedule (Table 19).

**Prompts and initialization.** The confidence prompt is a user message requesting final `Answer:` and `Confidence:` fields, with confidence in $[0, 1]$. The marker prompt is a system message requesting step-by-step reasoning, the literal string `<uncertain>` when a fact is uncertain, and a final `Answer:` line. For the marker interface, `<uncertain>` is a multi-token string in the existing vocabulary. Both runs initialize directly from the instruction-tuned base model.

**Outcome reward and evaluation.** The marker run uses Equation 6, including the repetition penalty, to score final-answer correctness and warning presence. Training uses type-aware yes/no and date matching and normalized answer matching with a token-F1 fallback. Development reporting uses $Y_{\text{rel}}$; the independent detector test uses type-aware matching or set-token F1 $> 0.5$, as specified in Section 5.2.

The six-task marker evaluation measures answer accuracy and response-format adherence without retrieved context. Its 500 questions per dataset yield the scores in Table 2; Table 13 adds answer-line and emission frequencies for those responses. Final-answer EM/F1 after a retrieval decision is evaluated separately in Table 8.

### D.1 Baseline Implementation Details

Table 9 evaluates verbal confidence on 500 held-out prompts, reporting strict exact match, relaxed accuracy under $Y_{\text{rel}}$, and the mean confidence reward. These summaries use different parsed and all-prompt denominators, so subtracting displayed reward from displayed accuracy does not give an exact Brier score. Table 12 evaluates binary error triggers on 2,433 shared test responses; Table 20 separately records the 1,100-response development evaluations. These are separate evaluations: confidence scores describe answer reliability, whereas a detector makes a binary warning decision.

**Self-RAG implementation.** The SR-7B and SR-13B rows use the authors' official Self-RAG implementation (Asai et al., 2024). Tables 7 and 8 report its results in the present retrieval evaluation; they are not transcribed scores from the Self-RAG paper.

**Verbalized-confidence baselines.** **Base.** The uncalibrated Llama-3-8B-Instruct model is prompted with the shared verbalized-confidence template and its native emitted decimal confidence is used directly. **Global TS.** A single scalar temperature is fit on the base model's training predictions by minimizing Bernoulli negative log-likelihood in logit space, then applied post hoc to the base model's test confidences. **ATS.** Adaptive temperature scaling predicts an example-specific temperature from lightweight response features, including the raw confidence logit, response length, answer length, and reasoning depth; the feature weights are fit on base-model outputs with L2-regularized Bernoulli NLL. **SFT-Conf.** This supervised baseline fine-tunes the model to reproduce the base model's reasoning and answer text while replacing the final confidence with a continuous confidence target derived from answer overlap with the reference. **SFT-KWDK.** This variant uses the same data and training pipeline as SFT-Conf, but replaces the continuous F1 target with a four-bucket confidence mapping. It tests whether coarse uncertainty supervision is sufficient, as opposed to continuous confidence regression.

**Error-detection baselines.** **Base marker.** The untrained Llama-3-8B-Instruct model is prompted with the same `<uncertain>` instruction used for policy-gradient training, and a trigger is fired whenever the literal token string appears anywhere in the greedy response. **Hidden probe.** We extract a hidden-state representation from the base model at a designated pre-answer readout position, fit a logistic regression probe to predict wrongness, choose the best layer by development AUPRC, and tune the decision threshold on held-out development data. **Output classifier.** This baseline fits logistic regression on surface response features only, including response length, reasoning-line count, hedging cues, and whether `<uncertain>` already appears. It tests whether errors in base-model responses can be predicted from surface features.

**Scoring and evaluation populations.** Appendix G gives parsed denominators, correctness counts, and parse-failure handling for the confidence baselines. Table 3 uses identical endpoint-inclusive binning for the base and trained models.

Table 20 gives integer confusion matrices, with wrong answers as positives. Base detectors share 799 wrong answers; the trained model has 581. Comparisons within the base group therefore hold answers fixed, whereas a base-to-trained comparison changes both generator and signal. The base hidden probe uses layer 16 and threshold $0.45$; the output classifier uses threshold $0.2$ selected by cross-validation.

| Method | TP | FP | FN | TN |
|---|---|---|---|---|
| Base marker | 355 | 15 | 444 | 286 |
| Base hidden probe | 684 | 85 | 115 | 216 |
| Base output classifier | 768 | 250 | 31 | 51 |
| Trained marker | 455 | 195 | 126 | 324 |

Table 20: Full-set detection counts, $N = 1100$ per row. Precision is TP/(TP+FP), recall is TP/(TP+FN), and untriggered accuracy is TN/(TN+FN), not specificity.

The trained raw-marker counts give precision $70.0\%$, error recall $78.3\%$, and untriggered-answer accuracy $72.0\%$. Thus a warning is informative about final-answer risk on the trained model's own outputs. A final answer consisting only of `<uncertain>` is counted as incorrect: this occurs in 5 of the 1,100 trained responses in this detection evaluation. This distinguishes answer-line abstention from the 650 responses containing a marker anywhere in the response.

# E Illustrative Marker Responses

## E.1 Four-Way Examples for the <uncertain> Marker

The `<uncertain>` marker is a binary signal inside the reasoning trace rather than a scalar confidence attached to the final answer. The four examples below illustrate answer agreement with the dataset reference crossed with marker presence. Correctness refers to the extracted final answer; it does not certify every statement in the accompanying rationale. The examples illustrate possible behaviors rather than their frequencies.

**Correct + no marker**

**Question:** Which element discovered in 1782 has the symbol Te?
**Gold answer:** Tellurium **Prediction:** Tellurium **Emitted <uncertain>:** No

**Model response:**

```
Let's break this down step by step!
Te is the symbol for the element Tellurium, which was discovered
in 1782 by Franz Muller from the copper ore found in the copper
mines in the Ural Mountains.
Answer: Tellurium
```

*The final answer matches the reference and no warning is emitted. This is a true negative for final-answer error detection, regardless of whether every detail in the rationale is accurate.*

**Wrong + marker emitted**

**Question:** Which country is the composer of the film *Neeya 2* from?
**Gold answer:** Singapore **Prediction:** Indian **Emitted <uncertain>:** Yes

**Model response:**

```
Let's break this down step by step!
Neeya 2 is a 2013 Tamil film composed by S. S. Thaman, who is a
Indian composer.
The composer, S. S. Thaman, is a Indian composer who hails from
India.
<uncertain> However, I'm not sure about the country of origin of
the composer, as there are many Indian composers.
Answer: Indian
```

*The final answer differs from the reference, and the response contains a warning about the composer's country. This is a true positive for error detection.*

**Correct + marker emitted**

**Question:** When was the last time England won the World Cup?
**Gold answer:** 1966 **Prediction:** 1966 **Emitted <uncertain>:** Yes

**Model response:**

```
Let's break this down step by step!
England has won the World Cup once, which was in 1966 when they
hosted the tournament.
They played in the final against West Germany on July 30, 1966,
and won the match 4-2 after extra time.
<uncertain> However, it's not clear when they last won the World
Cup, since they haven't won it since 1966.
Answer:  1966
```

*The final answer matches the reference despite the emitted warning. This is a false positive for final-answer error detection; it illustrates the cost of acting on every warning.*

**Wrong + no marker**

**Question:** In show-jumping, how many faults does a competitor get for a refusal?
**Gold answer:** 3 **Prediction:** 4 **Emitted <uncertain>:** No

**Model response:**

```
Let's break this down step by step!
A refusal is a fault in show jumping that occurs when the
horse refuses to jump a fence, either by stopping or changing
direction.
According to the rules of show jumping, a refusal is penalized
with 4 faults.
Answer:  4
```

*The final answer differs from the dataset reference and no warning is emitted. Under that reference, this is a false negative: marker absence cannot guarantee answer correctness.*

## F Broader Impact Statement

This work aims to reduce confident hallucinations by training LLMs to expose self-assessment within their own responses. Such signals may help downstream systems trigger retrieval, verification, abstention, or human review before acting on unreliable outputs. However, explicit self-assessment can also create over-trust: a high confidence score is not a correctness guarantee, and the absence of <uncertain> does not imply that the answer is safe. Our methods should therefore be used as control cues rather than standalone certificates of correctness. Deployment in high-stakes domains would require additional domain-specific calibration, external verification, and human oversight.

## G Evaluation Metrics and Signal Timing

**Correctness labels and denominators.** The confidence baseline table reports strict normalized exact match and relaxed accuracy under $Y_{\text{rel}}$: exact match or set-token F1 $\geq 0.3$. Set-token F1 measures overlap between distinct normalized tokens, ignoring repetitions. Parsed counts are 498 for Base and SFT-Conf, 499 for SFT-KWDK, and 500 for the confidence-trained model. Their strict-EM correct counts are 122, 105, 112, and 137, respectively; relaxed-correct counts are 186, 167, 172, and 191. Accuracy and calibration exclude parse failures, while mean reward includes all 500 responses and penalizes a parse failure by $-0.5$. Table 3 additionally evaluates the base and trained models at set-token F1 $\geq 0.5$, yielding 172 and 179 correct parsed responses.

**Calibration scores.** For $N$ parsed responses, Brier loss is $N^{-1}\sum_i (c_i - Y_i)^2$. Confidence NLL is

$$-\frac{1}{N}\sum_{i=1}^{N}\{Y_i \log \widetilde{c}_i + (1-Y_i)\log(1-\widetilde{c}_i)\}, \qquad \widetilde{c}_i = \min(1-10^{-12}, \max(10^{-12}, c_i)). \quad (8)$$

NLL uses natural logarithms and clips probabilities at $10^{-12}$ to keep the loss finite. It evaluates the confidence report, not token-generation likelihood. On a common binary-labeled population, mean $2Yc - c^2$ equals accuracy minus Brier loss; the reported reward also includes parse failures and their penalty.

For ECE (Guo et al., 2017), bins are equal width: the strict-EM and $Y_{\text{rel}}$ rows use 10 bins, and the set-token F1 $\geq 0.5$ row uses 15 bins. For bin $B_b$, let $\text{acc}(B_b)$ and $\text{conf}(B_b)$ be mean label and confidence. Then

$$\text{ECE} = \sum_{b:|B_b|>0} \frac{|B_b|}{N} |\,\text{acc}(B_b) - \text{conf}(B_b)|. \tag{9}$$

The final bin includes $c = 1$. Under this common rule, base/trained ECE is $62.4/13.3\%$ for strict EM, $49.5/3.6\%$ for $Y_{\text{rel}}$, and $52.5/4.9\%$ for set-token F1 $\geq 0.5$. The confidence reward in Table 9 uses $Y_{\text{rel}}$ and averages over all 500 prompts, as specified above.

**Emission timing.** Figure 2 uses the first marker position divided by response length. Each distribution conditions on responses that contain a marker and on final-answer correctness. Thus its density describes where an emitted warning appears; overall emission and error coverage are reported separately in Table 1. The current retrieval evaluation consumes the completed response, so the timing distribution characterizes signal availability rather than the effect of interrupting generation.

## H Retrieval Protocol and Detailed Results

The retrieval evaluation uses 500 questions from each of five datasets. Tables 7 and 8 report executed-system results. Aggregates average dataset means with equal weights, using the per-question scores before display rounding. Retrieval frequency counts actual evidence use after any marker filtering. The base-confidence forced-retrieval control in Table 22 requests evidence regardless of its confidence signal.

**Retrieval and answer generation.** The trained systems use Llama-3-8B reporting models for the first answer and Llama-3.1-8B-Instruct for the retrieved answer. Greedy decoding allows up to 512 new tokens. Each selected question undergoes one retrieval-assisted revision using up to three BM25 passages. HotpotQA, MuSiQue, and 2WikiMultihopQA use their corresponding indices; Natural Questions and TriviaQA use the 2WikiMultihopQA index as a Wikipedia proxy. This mapping is shared across methods. ADARAGUE denotes our output-feature classifier implementation of uncertainty-based retrieval control (Moskvoretskii et al., 2025). Probing-RAG (Baek et al., 2025) is a related but distinct hidden-state retrieval method. Self-RAG uses its official implementation and normalized reference-substring matching in the final sentence; the other rows use normalized EM. F1 uses token overlap.

**Generators and retrieval queries.** Single-Ret uses a question-only query and a plain context-answer prompt. It, FLARE, DRAGIN, and ADARAGUE use Llama-3.1-8B-Instruct; Self-RAG uses its own Llama-2 7B/13B checkpoints. For our two-stage systems, MuSiQue and 2Wiki use question-only queries; other queries can append the first answer. The revision prompt can include this candidate answer, and a question-only fallback can replace an answer-augmented query. Thus a common second-pass model does not imply common passages or prompts, explaining distinct Ret-All results. Retrieval frequency counts questions receiving evidence, not backend calls.

**Gate parameters.** The reported confidence gates use $\tau_c = 0.75$ on HotpotQA, $0.40$ on MuSiQue, $0.30$ on 2WikiMultihopQA, and $0.15$ on Natural Questions and TriviaQA. A separate shared-candidate sweep with a uniform $\tau_c = 0.5$ gives $31.4\%$ EM, $40.4\%$ F1, and $48.1\%$ retrieval. These are distinct operating points. A confidence parse failure triggers retrieval. The marker gate first checks emission, then applies a layer-13 ridge classifier to the mean state over the first marker span. Its regularization is selected on development responses and its threshold, $\tau_p = -0.329$, on training scores. Layer 13 is a fixed setting of the retrieval probe, not the winner of the separate logistic layer sweep. Neither the confidence training questions nor its 500-question calibration set overlap with the retrieval pool. Marker training questions are also disjoint from this pool, but development responses used for probe selection overlap with it. Thus the retrieval comparison is not fully independent of probe development. The separately screened detector evaluation is described in Appendix B.3.

**Isolating the retrieval decision.** Table 21 selects between a fixed first-pass answer and a fixed retrieved answer for each question. No-Ret always selects the former; Ret-All always selects the latter; The gate follows the report; Random matches its selection budget within each dataset. This holds answer candidates constant. Executed systems generate their retrieved answers in their own runs, so their final scores need not equal those obtained with shared answer candidates. For example, the Conf. gate is $31.88\%$ EM in the comparison with shared answer candidates and $31.84\%$ in the executed-system comparison; the Marker gate is $28.80\%$ and $29.04\%$, respectively. Each table uses its own explicitly defined answer set throughout.

**Paired uncertainty estimates.** Intervals use 5,000 paired bootstrap resamples within datasets, with question indices shared across compared methods. Tables 22 and 21 provide the aggregate intervals for executed systems and selection with shared answer candidates. With shared answer candidates, the Marker gate-minus-No-Ret EM difference is $0.76$ points, with paired 95% CI $[0.00, 1.56]$, touching zero. For the executed raw-marker comparison, raw-marker minus No-Ret EM is $+0.44$ points, CI $[-0.48, 1.36]$. The comparison with shared answer candidates yields Conf. gate-minus-Random EM $1.75$ points, 95% CI $[1.18, 2.34]$, and Marker gate-minus-Random EM $0.89$ points, $[0.24, 1.55]$.

| System | Match [95% CI] | F1 [95% CI] | Retrieval |
|---|---|---|---|
| Conf. No-Ret | 29.12 [27.48, 30.80] | 37.08 [35.44, 38.69] | 0.00 |
| Conf. Ret-All | 27.68 [25.96, 29.36] | 36.13 [34.45, 37.78] | 100.00 |
| Conf. gate | 31.88 [30.16, 33.68] | 41.05 [39.37, 42.72] | 45.52 |
| Conf. Random | 30.13 [28.51, 31.77] | 39.01 [37.45, 40.54] | 45.52 |
| Marker No-Ret | 28.04 [26.44, 29.72] | 34.74 [33.19, 36.38] | 0.00 |
| Marker Ret-All | 24.64 [23.00, 26.28] | 32.16 [30.54, 33.81] | 100.00 |
| Marker gate | 28.80 [27.08, 30.52] | 36.41 [34.77, 38.05] | 53.72 |
| Marker Random | 27.91 [26.34, 29.49] | 35.24 [33.73, 36.76] | 53.72 |

Table 21: Retrieval selection with shared answer candidates, macro-averaged across five datasets. Random selection matches the gate's retrieval budget within each dataset. Match uses exact match. Values are percentages; intervals use paired bootstrap resampling within dataset.

| System | Match [95% CI] | F1 [95% CI] | Trigger | Retrieval |
|---|---|---|---|---|
| SR-7B | 6.52 [5.60, 7.48] | 16.69 [15.59, 17.75] | 46.20 | 46.20 |
| SR-13B | 8.80 [7.76, 9.80] | 22.44 [21.10, 23.71] | 29.48 | 29.48 |
| Single-Ret | 24.84 [23.20, 26.52] | 34.29 [32.66, 35.95] | 100.00 | 100.00 |
| FLARE | 21.60 [20.12, 23.16] | 30.72 [29.24, 32.22] | 38.36 | 38.36 |
| DRAGIN | 32.68 [30.88, 34.44] | 43.17 [41.42, 44.93] | 76.84 | 76.84 |
| ADARAGUE | 11.00 [9.80, 12.20] | 15.51 [14.25, 16.77] | 43.80 | 43.80 |
| Base conf. + Ret-All | 21.84 [20.32, 23.44] | 27.90 [26.35, 29.47] | 13.08 | 100.00 |
| Conf. No-Ret | 29.12 [27.48, 30.80] | 37.08 [35.44, 38.69] | 45.52 | 0.00 |
| Conf. Ret-All | 27.68 [25.96, 29.36] | 36.13 [34.45, 37.78] | 45.52 | 100.00 |
| Conf. gate | 31.84 [30.12, 33.60] | 41.01 [39.36, 42.72] | 45.52 | 45.52 |
| Marker No-Ret | 28.04 [26.44, 29.72] | 34.74 [33.19, 36.38] | 61.36 | 0.00 |
| Raw marker | 28.48 [26.76, 30.16] | 35.98 [34.36, 37.62] | 61.36 | 61.36 |
| Marker gate | 29.04 [27.32, 30.72] | 36.59 [34.95, 38.24] | 61.36 | 53.72 |

Table 22: Aggregate retrieval results with 95% paired intervals (%). Shared No-Ret/Ret-All outputs reuse Table 21 intervals. Match is EM except for Self-RAG's final-sentence substring scorer. Trigger is the pre-filter trigger frequency; Retrieval is actual evidence use. Base conf. + Ret-All uses an untrained confidence-reporting first-pass model and retrieves for every question, irrespective of its confidence signal.