A Reasonable Approach to Hallucination Mitigation on HotPotQA

Columbia COMS4705 Final Project Report
Keywords: *hallucination, Q&A, reinforcement learning, decoding, abstension*

Joshua Hegstad
Department of Computer Science
Columbia University
jlh2288@columbia.edu

Ahmed Jaber
Department of Computer Science
Columbia University
amj2234@columbia.edu

Farhaan Siddiqui
Department of Computer Science
Columbia University
fs2872@columbia.edu

Abstract

Closed-book question answering (QA) systems must answer questions using only parametric knowledge, which makes them prone to *closed-book hallucination*: confident but factually incorrect answers with no supporting evidence [1]. We view this behavior as arising from two failure modes: (1) the model lacks the relevant knowledge, and (2) the knowledge is implicitly present in its parameters, but the model does not reliably retrieve or apply it at inference time [2, 3]. We study these failure modes on HotPotQA using Qwen2.5-7B and Qwen3-8B and propose a three-stage mitigation pipeline.

To address the "cannot access knowledge" case, we perform Chain-of-Thought (CoT) distillation from a larger Qwen3-235B teacher to unlock latent parametric knowledge. To address the "no knowledge" case, we train the model to abstain. We replace incorrect predictions with "I don't know" and finetune on a class-balanced mix of correct answers and abstentions. Finally, we apply Reinforcement Learning from Verifier Feedback (RLVF) with an NLI-derived factuality score to heavily penalize confident errors. On HotPotQA, this pipeline reduces hallucinated responses and increases appropriate refusals. Furthermore, we demonstrate that unsupervised Semantic Entropy (SE) effectively flags residual hallucinations, enabling a consensus-based rejection strategy that improves selective F1 score from 0.50 to 0.70.

1 Key Information to include

Mentor: Melody Ma **External Collaborators:** None **Sharing project:** N/A

2 Introduction

Large Language Models (LLMs) frequently suffer from *hallucination* in closed-book settings, generating plausible but factually incorrect assertions [1]. This issue is particularly acute in Small Language Models (SLMs), which often lack the capacity to distinguish between what they know and what they have forgotten. This unreliability creates a critical barrier for deployment in safety-critical or privacy-constrained environments where Retrieval-Augmented Generation (RAG) is not feasible.

Columbia COMS4705: Natural Language Processing, Fall 2025. Adapted from Stanford CS224N Natural Language Processing with Deep Learning.

To address this, we propose a pipeline to enhance both capability and calibration in Qwen2.5-7B-Instruct. We employ **Chain-of-Thought (CoT) Distillation** from a massive teacher to "unlock" latent parametric knowledge, followed by **Abstention-Aware Fine-Tuning** and **Reinforcement Learning from Verifier Feedback (RLVF)** to teach the model to explicitly abstain when it lacks knowledge.

Our experiments on HotPotQA validate this multi-stage approach. We demonstrate that CoT distillation improves Exact Match accuracy from 18.86% to 24.31%, significantly outperforming standard SFT. Furthermore, our supervised abstention baseline achieves 97.5% precision in detecting knowledge boundaries. We find that RLVF effectively tunes the risk-coverage trade-off, increasing answer rates by $\sim 5\%$ while maintaining safety. Finally, we validate that unsupervised *Semantic Entropy (SE)* serves as a robust proxy for correctness, allowing for the rejection of residual hallucinations at inference time.

3 Related Work

Hallucination in LLMs: Hallucination refers to the generation of text that is nonsensical or unfaithful to the provided source or established world knowledge [1]. In closed-book settings, where models rely on parametric memory, hallucinations often stem from the model's inability to accurately assess the boundaries of its own knowledge [2]. While larger models exhibit emergent self-calibration, Small Language Models (SLMs) are particularly prone to confident errors, necessitating explicit intervention strategies.

Reasoning Distillation: Chain-of-Thought (CoT) prompting significantly enhances LLM reasoning but typically requires large model scales (>100B parameters) to be effective. To transfer this capability to smaller models, [4] and [5] proposed CoT distillation, where a student model is fine-tuned on reasoning traces generated by a large teacher. Our work extends this by imposing "internalized knowledge" constraints on the teacher, ensuring the distilled reasoning relies on facts the student can plausibly retrieve from its own weights rather than external context.

Abstention and Uncertainty: Teaching models to say "I don't know" is a classic problem in reliable QA [6]. Recent approaches have focused on unsupervised methods to detect uncertainty, such as analyzing the entropy of semantic clusters in model generations (Semantic Entropy) [7] or inspecting internal attention patterns (RAUQ) [8]. [3] demonstrated that models can be trained to verbalize this uncertainty. We combine these streams by using supervised abstention for base calibration and refining it with NLI-based Reinforcement Learning (RLVF) that directly rewards the alignment between confidence (low entropy) and factuality.

4 Approach

Key Models:

- (BASELINE) Qwen2.5-7B-Instruct [9]: Referred to as **Qwen2.5-Instruct**.
- (BASELINE) Qwen2.5-7B-Instruct (SFT): Referred to as **Qwen2.5-SFT**. Finetuned using LoRA on 10,000 question-answer pairs in HotPotQA
- Qwen2.5-7B-Instruct (Finetune-CoT): Referred to as **Qwen2.5-FCoT**. Finetuned on Dataset B to generate a reasoning trace and answer for each question, without the context of supporting paragraphs.
- Qwen2.5-7B-Instruct (Abstention Finetuned): Referred to as **Qwen2.5-Abstain**. Finetuned using QLoRA on the question-answer/abstention pairs in Dataset C.
- Qwen2.5-7B-Instruct (Abstention Finetuned + RLVF): Referred to as **Qwen2.5-RLVF**.

Finetune-CoT To enable complex reasoning capabilities in our smaller student model, we employed Fine-tune-CoT [5], a framework that leverages the capabilities of large language models (LLMs) to supervise the training of smaller models. While massive LLMs can solve complex tasks using Chain-of-Thought (CoT) prompting, they are often computationally prohibitive to deploy. Fine-tune-CoT circumvents this by using a large "Teacher" model to generate synthetic reasoning traces, which are then used to fine-tune a "Student" model. This effectively transfers the reasoning ability, allowing the smaller model to learn how to solve the problem.

The standard Fine-tune-CoT pipeline consists of three stages: (1) prompting the teacher to generate step-by-step rationales, (2) curating the data by filtering out samples with incorrect answers, and (3) fine-tuning the student on the validated reasoning traces.

In our implementation, we adapted the Generation and Curation stages to create a specialized "Internalized Knowledge" dataset (Dataset B). We utilized Qwen3-235B as our teacher model. To address the common issue of hallucination during reasoning generation, we provided the teacher with the ground-truth paragraphs as context. However, to ensure the student model learns to function in a closed-book setting, we imposed strict constraints on the teacher's output: it was prompted to frame its reasoning as a step-by-step recall of facts, explicitly forbidding citation markers (e.g., "according to the passage"). We then implemented a validation loop that filtered samples not only for answer correctness (as in standard Fine-tune-CoT) but also for adherence to this "internalized" stylistic constraint.

Abstention-Aware Fine-Tuning To mitigate hallucinations, we employed a specialized fine-tuning strategy designed to calibrate the model's confidence. Rather than forcing the model to answer every query—which encourages hallucination on out-of-distribution or forgotten knowledge—we fine-tuned the model to explicitly output a refusal phrase ("I don't know") when it lacks the internal knowledge to answer correctly.

We first aimed to map the "knowledge boundary" of Qwen2.5. We hypothesize that if the Qwen2.5-FCoT generates an incorrect answer for a training example x despite having access to its internal weights, that example represents a hallucination-prone region. We performed inference on the first $N = 40,000$ examples of the HotpotQA training set. For every example, we compared the model's predicted answer $\hat{y}$ against the ground truth y using Exact Match (EM) scoring. This effectively partitioned our training data into two sets:

- Correct Set ($S_{correct}$): Samples where the model's internal knowledge was sufficient (EM$(\hat{y}, y) = 1$).
- Hallucination Set ($S_{hallucination}$): Samples where the model confidently generated an incorrect answer (EM$(\hat{y}, y) = 0$)

Using the partitioned data, we constructed a new supervised fine-tuning dataset (Dataset C) designed to reinforce known knowledge while penalizing hallucinations. First, we relabeled the ground truth for $S_{hallucination}$ to "I don't know", and kept the ground truth for $S_{correct}$. However, a significant risk in abstention training is the "laziness" objective, where a model learns that outputting "I don't know" is a global minimum for loss reduction. To prevent the model from defaulting to refusal, we enforced a strict class balance by randomly sampling 8000 examples from $S_{correct}$ and 2000 examples from $S_{hallucination}$ where the model should now abstain. This resulted in a final dataset of 10,000 examples will an 80/20 class balance. We then finetuned Qwen2.5-Instruct using QLoRA.

Reinforcement Learning from Verifier Feedback (RLVF): To refine the abstention boundary learned during supervised fine-tuning, we train the Qwen2.5-Abstain model with policy-gradient RL using a reward signal derived entirely from verifier factuality and model confidence. Although inference is closed-book, the reward model is *not*: the **FactualityVerifier**—a DeBERTa-based NLI model used in prior factuality and hallucination work [10, 11] —receives the full HotPotQA supporting paragraphs [12]. For each example, we concatenate all supporting titles and sentences into a single premise and compute:

- p_{ent}: entailment probability,
- p_{cont}: contradiction probability,
- $f = p_{\text{ent}} - p_{\text{cont}}$: grounded factuality score.

Thus RLVF optimizes semantic factuality rather than gold-label EM.

Confidence Term: We include a normalized entropy confidence score

$$\text{conf} = 1 - \frac{H_{\text{avg}}}{H_{\max}},$$

following semantic-entropy–based uncertainty estimation [7]. Confidence increases reward only when supported by positive factuality.

Abstention Detection: Abstentions are not identified via string matching. Instead, the **Abstention-Classifier** applies the same NLI verifier to check whether the model's answer entails an abstention-style statement (e.g., "I don't know"), similar in spirit to prior work on verbalized uncertainty [3]. If so, the model receives a fixed penalty of -1.0.

Reward Function: For non-abstaining outputs, reward depends solely on verifier factuality and confidence:

$$R = \begin{cases} 10\, f(\lambda_{\text{base}} + \lambda_{\text{conf}} \cdot \text{conf}) & f \geq 0, \\ 5\, f(\lambda_{\text{base}} + \lambda_{\text{conf}} \cdot \text{conf}) & f < 0. \end{cases}$$

Positive factuality is up-weighted (10×), negative factuality down-weighted (5×).

We underscore four key design choices in our RLVF implementation. **First**, by grounding the reward in verifier evidence rather than Exact Match (EM), we optimize for semantic correctness, avoiding the brittleness of string matching for valid paraphrases [1]. **Second**, we incorporate an entropy-normalized confidence term into the reward, mirroring recent uncertainty-based shaping techniques [7] to encourage calibration. **Third**, we employ asymmetric scaling—applying larger multipliers for positive factuality—to bias the policy toward answering while still strictly penalizing contradictions. **Finally**, we "warm start" the RL process from our supervised abstention checkpoint; this ensures the model possesses a pre-established abstention boundary, significantly stabilizing the early stages of policy optimization compared to training from scratch [5].

4.1 Scalable Implementation and Model Transition

Platform Migration and Architecture Upgrade Our Initial experiments utilized Qwen2.5-7B on a standard virtual machine infrastructure. However, to address training instability and accelerate throughput (achieving a $\sim 10\times$ speedup), we migrated our training pipeline to the **Tinker** high-performance compute platform. As the platform does not offer a Qwen2.5 model, we transitioned to **Qwen3-8B**. Validations confirmed that Qwen3-8B performs comparably to Qwen2.5-7B on our base benchmarks, ensuring that our architectural conclusions remain robust.

Data Scaling Leveraging this improved throughput, we scaled our supervised fine-tuning (SFT) from the initial 10,000-sample subset to the **full 80,000-sample HotPotQA training set**. Similarly, we re-distilled the CoT and Abstention models using the complete teacher-generated dataset, significantly expanding the breadth of training data compared to the pilot experiments.

Uncertainty Estimation: RAUQ → Semantic Entropy While our initial design proposed using the RAUQ metric [8], which relies on attention weight introspection, the abstracted inference API of our scaled platform precludes access to internal hidden states. Adopting a black-box approach, we implemented **Discrete Semantic Entropy (SE)** [7].

To estimate SE without access to token probabilities, we sample $M = 10$ stochastic generations per question and cluster them based on semantic equivalence using a bidirectional NLI entailment model (`microsoft/deberta-large-mnli`). The probability of each semantic cluster C is approximated via Monte Carlo integration:

$$P(C|x) \approx \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}(s_i \in C) \tag{1}$$

The total uncertainty is then given by the entropy over these semantic clusters: $H(x) = -\sum_C P(C|x) \log P(C|x)$. This metric serves as our primary signal for evaluating the efficacy of our abstention mechanism in calculating and rejecting hallucinations.

5 Experiments

5.1 Data

We use the the training and validation splits of the HotPotQA [12] distractor dataset. The distractor setting has question-answer pairs with ten paragraphs as context, two of which are relevant to answering the question. Since we ran our tests in a closed-book setting, we discarded all ten context paragraphs for our finetuning, RL, and evaluation methods. We define 3 training datasets:

Dataset A1/A2: The first and second 10,000 question-answer pairs of HotPotQA (distractor setting, train split), respectively.

Dataset B: The first 10,000 question-answer pairs of HotPotQA (distractor/train) with reasoning traces generated according to our "Finetune-CoT" section. 4

- **Q:** Which magazine was started first Arthur's Magazine or First for Women?
- **R:** **Step 1**: The question asks ... **Step 2**: Arthur's Magazine was published starting in 1844... **Step 3**: Since 1844 is earlier than the 1980s ...the answer is Arthur's Magazine. [1]
- **A:** Arthur's Magazine

Dataset C: 10,000 question-answer or question-abstion pairs in the first 40,000 samples of HotPotQA (distractor/train). See our "Abstention-Aware Fine-Tuning" section 4 for details on generation.

- **Q:** Who wrote the 1971 drama which stars the winner of the 1997 BAFTA Fellowship?
  A: I don't know.

5.2 Evaluation method

We use Exact Match and F1 between gold answers and the model's predicted answers to gauge performance. For abstention models, we also calculate Selective EM and Selective F1 by considering only the answers not abstained on. We also gauge the uncertainty of our models via RAUQ and semantic entropy during inference. By sorting model responses by their RAUQ score [8] and iteratively rejecting the most uncertain samples, we use this as a target performance for our model's abstention.

5.3 Experimental details

We evaluated **Qwen2.5-Instruct** (bfloat16) and its fine-tuned variants (**SFT, FCoT, Abstain**) using QLoRA (4-bit NF4 base, FP16 adapters). All fine-tuning experiments were run for **one epoch**.

Training Configuration: We applied LoRA ($r = 16, \alpha = 32$, dropout 0.05) to all linear projection layers. Optimization used `adamw_torch` with a cosine scheduler (3% warmup) and an effective batch size of 16. **Supervised Baselines (SFT, FCoT, Abstain):** Trained with a learning rate of 2e-4. **SFT** used standard chat formatting (max length 512), masking user prompts. **FCoT & Abstain** utilized the specific delimiter format (`Question ### Rationale -> Answer`) to minimize token overhead, with a max length of 2048 tokens to accommodate reasoning traces. Qwen2.5-SFT was trained on Dataset A1, Qwen2.5-FCoT was trained on dataset Dataset B, and Qwen2.5-Abstain was trained on Dataset C. **Qwen2.5-RLVF:** Initialized from the supervised abstention (Qwen2.5-Abstain) checkpoint. Trained using on-policy generation with Dataset A2 with a learning rate of 1e-5 and weight decay 0.01.

Inference: Experiments were conducted in a closed-book setting using greedy decoding. **Standard Models:** Used a concise system prompt restricting output to short entities (max gen: 50 tokens). **Reasoning Models:** Prompted with the `###` delimiter to trigger reasoning traces (max gen: 512 tokens); answers were parsed programmatically.

5.4 Results

Generation Performance and Error Correction: Our fine-tuning experiments demonstrate clear improvements over the Qwen2.5-Instruct baseline across all metrics. As shown in Table 7, the Chain-of-Thought (CoT) model achieved the highest exact match accuracy of 24.31%, a significant gain over the Base model's 18.86%. Notably, the CoT approach proved superior in error correction, successfully rectifying 697 questions that the Base model originally answered incorrectly, compared to 477 corrections by the SFT model. While both fine-tuning methods introduced regressions—breaking a smaller number of previously correct answers (249 for SFT and 294 for CoT)—the net performance gain validates the efficacy of distilling reasoning traces over standard supervised fine-tuning for this domain. Furthermore, our best performing model (Qwen2.5-FCoT) acheieved near SoTA performance compared to much larger models. Specifically, PaLM-62B acheieved EM 26.46 and F1 35.67, validating our method's effectiveness 5.

[1] Parts of the reasoning trace redacted for conciseness

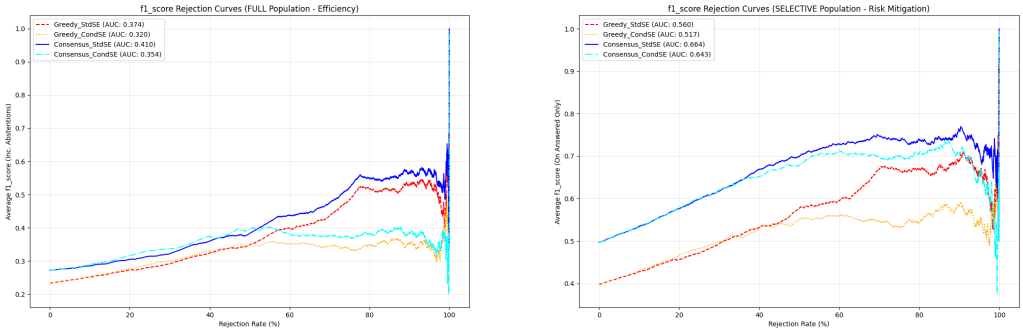
Abstention Mechanism and Reliability: Table 6 presents our abstention models' precision and recall metrics, where precision measures the percentage of abstentions that correspond to questions the reference model answered incorrectly (validating that the model abstains when appropriate), and recall measures the percentage of reference model errors that were successfully avoided through abstention.

The supervised model (Qwen2.5-Abstain) demonstrates highly conservative behavior, achieving 97.49% precision against the Base model, meaning its abstentions are almost invariably justified. Its recall ranges from 55-61% across reference models, performing best on the "Hardest" subset (60.85%)—questions where all baseline models failed. This confirms the mechanism activates most strongly on genuinely difficult queries.

The RLVF model exhibits a different profile: it exhibits higher precision (91-98%) but achieves lower recall (50-54%). This trade-off is intentional—the reward structure penalizes abstention modestly (-1) relative to hallucinations (-5), encouraging the model to attempt more questions. As shown in Table 5, this yields higher coverage: the RLVF model answers 58.83% of questions versus the supervised model's 53.65%.

However, this increased coverage comes at the cost of selective accuracy. The supervised model achieves 37.98% EM on attempted questions, compared to the RLVF model's 34.62%. This reflects the fundamental trade-off visualized in Figure 7: the supervised approach prioritizes safety (high precision, conservative abstention), while the RLVF approach prioritizes coverage (more attempts, lower selective accuracy). Both strategies are valid depending on application requirements—safety-critical domains benefit from the supervised model's conservatism, while general-purpose systems benefit from the RLVF model's higher answer rate.

5.4.1 Semantic Entropy and Abstention Results



Full Population (Efficiency) Selective Population (Risk Mitigation)

Figure 1: F1 Score Rejection Curves: Comparing efficiency on the full dataset vs risk mitigation on answered questions.

Uncertainty Quantification vs. Correctness We validated the efficacy of Semantic Entropy (SE) as a proxy for correctness. As shown in Figure 2 (Right), the distribution of SE for correct answers is heavily clustered around 0 (median ≈ 0), whereas incorrect answers exhibit significantly higher variance with a median of ~ 0.9. This confirm that high Semantic Entropy successfully signals likely hallucinations.

Abstention Calibration We analyzed whether the model's native abstention mechanism aligns with semantic uncertainty (Figure 2 Left/Center). We examined both **Standard Semantic Entropy (Std SE)**, which measures total uncertainty over all clusters, and **Conditional Semantic Entropy (Cond SE)**, which normalizes uncertainty by ignoring the probability mass assigned to the "I don't know" cluster. For both metrics, the "Abstained" distribution is consistently shifted higher than the "Answered" distribution, indicating the model is correctly choosing to abstain on questions where it is internally uncertain, though the overlap suggests further gains are possible by explicitly rejecting high-entropy answers.

Rejection Curve Efficiency Figure 1 demonstrates the trade-off efficiency of utilizing SE for rejection. The **x-axis** represents the *rejection rate*: the percentage of samples discarded by ranking all questions by their Semantic Entropy score (highest first) and removing the top-$k\%$. The **y-axis** shows the resulting F1 score on the retained samples. We compared two uncertainty estimation strategies: **Greedy**, which estimates confidence based solely on the single most likely token sequence, and **Consensus**, which marginalizes probability mass over all sampled generations that share the same semantic meaning (the "semantic cluster"). As shown in Table 4, the **Consensus** strategy consistently outperforms Greedy. At 0% rejection (all samples retained), Consensus achieves a Selective F1 of 0.50 compared to Greedy's 0.40. By rejecting the top 50% most uncertain samples, the Consensus strategy improves Selective F1 to **0.70**. This confirms that aggregating semantic consistency across multiple samples provides a more robust signal for hallucination detection than single-path generation.

6 Analysis

6.1 "Reasoning Trace" Qualitative Check

Evidence of Successful "Internalized Knowledge" Transfer: Qualitative inspection reveals that the CoT model successfully learned the 'internalized knowledge' style. As seen in Table 1, Qwen2.5-FCoT model bridged the gap where Qwen2.5-Instruct failed by explicitly retrieving the nationality of both entities in Step 2, correcting the Instruct model's factual error without using external citations.

Context (Base Wrong → CoT Correct)	**Generated Reasoning Trace**
Q: Were Scott Derrickson and Ed Wood of the same nationality? **Gold:** yes **Base Pred:** No (Wrong) **CoT Pred:** yes (Correct)	Step 1: The question asks about the nationalities of Scott Derrickson and Ed Wood. Step 2: Scott Derrickson is an American film director... while Ed Wood was an American filmmaker... Step 3: Since both individuals were born and active in the United States... the answer is yes.

Table 1: Example of Successful Knowledge Retrieval

Analysis of Regression & Hallucination: However, the reasoning process introduces two distinct failure modes where the Instruct model was originally correct but the FCoT model regressed (see Table 2). First, **Hallucination via Reasoning**: The step-by-step generation can induce false details. In the Rose McGowan example (Table 2, Top), the FCoT model hallucinated a specific role (Cordelia Chase) for the wrong actress, overriding the Instruct model's correct association. Second, **Granularity Mismatch**: The reasoning can lead to over-specification. In the "occupation" example (Table 2, Bottom), the FCoT model deduced "film director" based on specific evidence. While factually correct, this was penalized against the rigid gold label "director," highlighting a limitation in Exact Match (EM) metrics for reasoning tasks and the usefulness of incorporating F1 metrics.

Context (Base Correct → CoT Wrong)	**Reasoning Trace Fault**
Failure Mode 1: Hallucination	
Q: What WB supernatural drama series was Jawbreaker star Rose McGowan best known for being in? **Gold/Base:** Charmed **CoT Pred:** Buffy the Vampire Slayer	Step 2: Rose McGowan is best known for her role as Cordelia Chase in the WB supernatural drama series "Buffy the Vampire Slayer." *(Error: Hallucinated incorrect role/show.)*
Failure Mode 2: Granularity Mismatch	
Q: What occupation do Chris Menges and Aram Avakian share? **Gold/Base:** Director **CoT Pred:** film director	Step 3: Since both individuals are primarily recognized for directing films, their shared occupation is film director. *(Error: Trace is too specific for Gold Label.)*

Table 2: Analysis of CoT Failure Modes (Regression)

6.2 SFT vs. RLVF Abstention Behavior (Risk Comparison)

Comparing the two abstention strategies reveals clear behavioral boundaries. For the query, *"What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell?"*,

the SFT model responded with "I don't know," reflecting the conservative abstention style learned during supervised training. In contrast, the RLVF model, which is optimized under an asymmetric reward structure that penalizes abstention (-1) less severely than incorrect answers (-5), chose to attempt an answer and predicted "First Lady." Because the correct answer is "Chief of Protocol," this instance illustrates a *confident hallucination*: the model preferred risking an incorrect response rather than accepting the abstention penalty. Such cases demonstrate the "risk" side of the trade-off, where the model's incentive to avoid the abstention penalty leads to unreliable or overconfident guesses.

However, the same mechanism can also produce beneficial behavior. For example, in the query *"What army did the namesake of the ship launched as the München in 1930 fight in during the American Revolutionary War?"*, the SFT model again abstained with "I don't know." The RLVF model, facing the same penalty structure, elected to answer and produced the correct response: "Continental Army." This represents the "reward" side of the trade-off: the model takes a calculated risk and, in this case, succeeds in recovering information that the SFT baseline refuses to commit to. Instances like this show that the asymmetric reward can indeed encourage productive non-abstentions, enabling the model to fill gaps where supervised training remains overly cautious.

Together, these two examples highlight the central tension of abstention-aware RLHF. A more aggressive penalty on abstention encourages the model to answer more often, which can yield legitimate gains in coverage, but also increases the likelihood of confident hallucinations. Achieving the right balance requires careful tuning of the abstention penalty and, potentially, more structured confidence modeling to distinguish informative risk-taking from unreliable guessing.

7 Conclusion

In this work, we presented a comprehensive approach to mitigating hallucination in Small Language Models for closed-book question answering. By decomposing the hallucination problem into "access failures" and "knowledge gaps," we applied targeted interventions: Chain-of-Thought distillation to improve reasoning-based retrieval, and Abstention-Aware Fine-Tuning to calibrate the model's refusal behavior.

Our experiments on HotPotQA reveal that these strategies are highly effective. CoT distillation proved superior to standard fine-tuning, correcting 697 baseline errors by guiding the model to step-by-step derivational answers. For uncorrectable knowledge gaps, our supervised abstention mechanism achieved a 97.5% precision in flagging hallucinations, effectively converting "confident errors" into "safe refusals." Furthermore, we demonstrated that Reinforcement Learning with Verifier Feedback (RLVF) offers a dynamic method to tune the trade-off between caution and helpfulness, allowing system designers to optimize for coverage without sacrificing fundamental reliability.

However, limitations remain. Our RLVF approach, while increasing coverage, slightly degraded selective accuracy compared to the purely supervised baseline, highlighting the difficulty of optimizing reward landscapes where "safety" (abstention) is a local optimum. Additionally, our reliance on Semantic Entropy for post-hoc filtering, while effective, incurs a significant computational cost during inference.

Future work will focus on two directions: (1) developing lightweight, cascading uncertainty filters (e.g., using RAUQ as a pre-filter for Semantic Entropy) to reduce inference latency, and (2) extending our evaluation to open-book settings to assess whether these calibration gains transfer when the model must discriminate between parametric knowledge and retrieved context. Ultimately, our findings suggest that "knowing what you don't know" is a learnable capability for SLMs, paving the way for more trustworthy autonomous agents.

8 Team Contributions

Joshua wrote the code for training and evaluating the Tinker models and running semantic entropy evaluations, and wrote all analysis thereof.

Farhaan wrote code for training and evaluating the Qwen2.5 finetuned models, including generating the CoT and Abstention datasets and analyzing the reasoning traces.

Ahmed implemented the Reward Learning (RLVF) training and evaluation pipeline, developed the NLI-verifier and Abstention classification, and analyzed the abstention behavior.

References

[1] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Ho Shu Chan, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55, 2023.

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

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

[4] Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. *arXiv preprint arXiv:2306.14050*, 2023.

[5] Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers, 2023.

[6] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, 2018.

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

[8] Artem Vazhentsev, Lyudmila Rvanova, Gleb Kuzmin, Ekaterina Fadeeva, Ivan Lazichny, Alexander Panchenko, Maxim Panov, Timothy Baldwin, Mrinmaya Sachan, Preslav Nakov, and Artem Shelmanov. Uncertainty-aware attention heads: Efficient unsupervised uncertainty quantification for llms, 2025.

[9] Qwen: An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

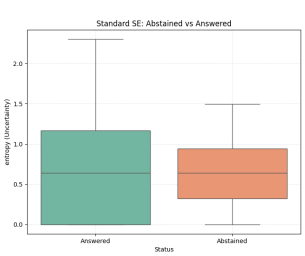
[10] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021.

[11] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021.

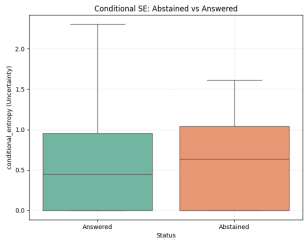
[12] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018.

A Appendix

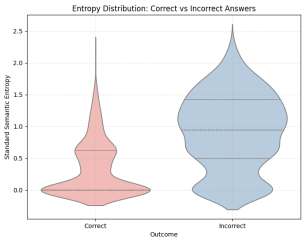
A.1 Abstention Model vs. SE Analysis



Std SE: Abstained vs Answered



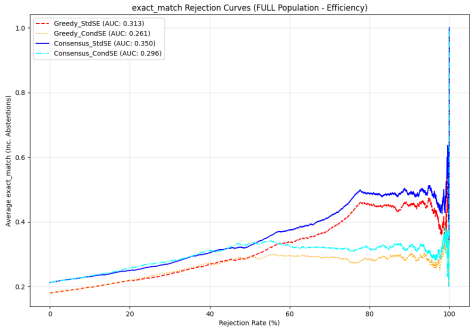
Cond SE: Abstained vs Answered



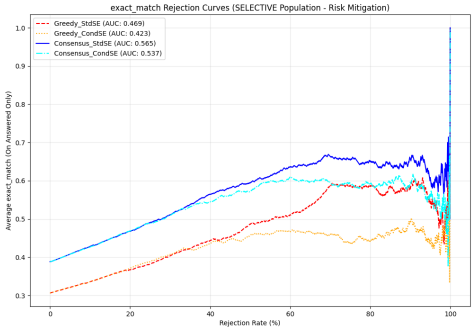
Std SE: Correct vs Incorrect

Figure 2: Distribution of Uncertainty (Entropy) across outcomes.

A.2 Additional SE Rejection Curves



Full Population (Efficiency)



Selective Population (Risk Mitigation)

Figure 3: Exact Match (EM) Rejection Curves.

Table 3: Impact of Semantic Entropy-based Rejection on Exact Match ($N = 2500$)

Strategy	**Rejection Threshold**	**Full EM**	**Selective EM**
Greedy	0%	0.1796	0.3065
Greedy	50%	0.2912	0.4863
Consensus	0%	0.2124	0.3884
Consensus	50%	0.3256	**0.5944**

Table 4: Impact of Semantic Entropy-based Rejection on F1 Scores ($N = 2500$)

Strategy	**Rejection Threshold**	**Full F1**	**Selective F1**
Greedy	0%	0.2329	0.3974
Greedy	50%	0.3482	0.5771
Consensus	0%	0.2719	0.4972
Consensus	50%	0.3828	**0.6991**

A.3 Supplementary RAUQ Rejection Curves

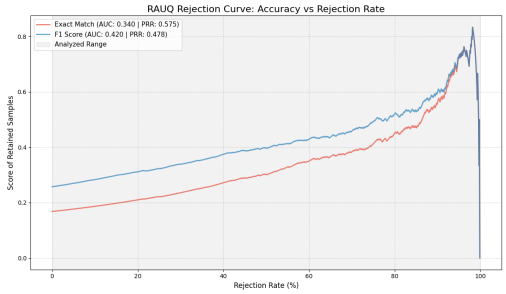


Figure 4: Vanilla Qwen2.5-7B Rejection Curve

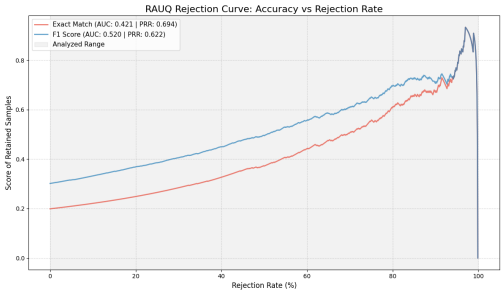


Figure 5: Finetuned Rejection Curve

Figure 6: RAUQ Rejection Curves Comparison. Both models show improved accuracy as high-uncertainty samples are rejected, with the finetuned model achieving higher overall retention quality.

A.4 Confusion Matrix

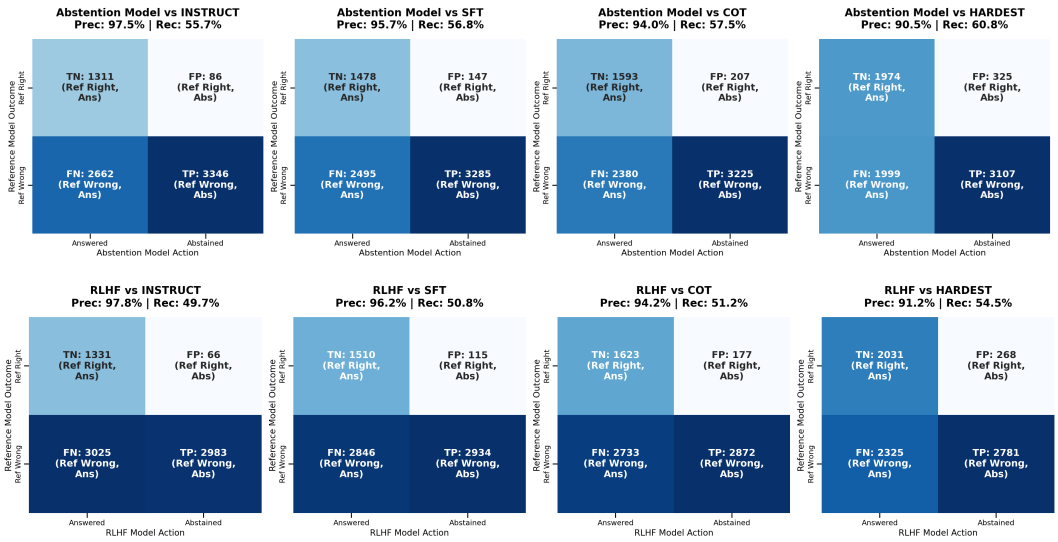


Figure 7: Abstention Performance Comparison: Supervised Fine-Tuning vs. RLHF. **Top**: Confusion matrices for the supervised abstention model (Qwen2.5-Abstain SFT), which was trained with explicit abstention labels on questions where baseline models failed. **Bottom**: Confusion matrices for the RLHF model, trained using policy gradient reinforcement learning with reward shaping. Each panel shows how well the model's abstention decisions align with questions that reference models answered incorrectly.

A.5 Additional Tables

Table 5: Comparison of Experimental and State-of-the-Art Results on the HotpotQA Dataset (Distractor/Validation) (EM and F1 scores are percentages).

Model	EM	F1	Abs. Rate	Sel. EM	Sel. F1	Avg. RAUQ
Experimental Results ($N = 7405$ Samples)						
Qwen2.5-Instruct	18.86	27.02	–	–	–	1.25
Qwen2.5-SFT	21.94	31.41	–	–	–	1.62
Qwen2.5-FCoT	**24.31**	33.87	–	–	–	–
Qwen2.5-Abstain	19.47	25.78	46.34	37.98	48.46	1.32
Qwen2.5-RLVF	20.37	26.24	41.17	34.62	44.60	1.40
SOTA Closed-Book Results (RECITE Paper)						
Codex-002	37.11	48.37	N/A	N/A	N/A	N/A
PaLM-62B	26.46	35.67	N/A	N/A	N/A	N/A

Table 6: Performance Analysis of Abstention Mechanism Across Reference Models

Model	Ref. Model	Precision	Recall	TP (Good Abs)	FP (Bad Abs)	FN (Missed)
4*Qwen2.5-Abstain	Base	**97.49%**	55.69%	3,346	**86**	2,662
	SFT	95.72%	56.83%	3,285	147	2,495
	CoT	93.97%	57.54%	3,225	207	2,380
	Hardest	90.53%	**60.85%**	3,107	325	**1,999**
4*Qwen2.5-RLVF	Base	**97.84%**	49.65%	2,983	**66**	3,025
	SFT	96.23%	50.76%	2,934	115	2,846
	CoT	94.19%	51.24%	2,872	177	2,733
	Hardest	91.21%	54.47%	2,781	268	2,325

Table 7: Overall Model Performance and Error Analysis Relative to Base Model

Metric	Base Model	SFT Model	CoT Model
Overall Accuracy (Exact Match)	18.87%	21.94%	**24.31%**
Corrections (Base Wrong $\rightarrow$ Right)	–	477	**697**
Regressions (Base Right $\rightarrow$ Wrong)	–	**249**	294