Self-Evaluating LLMs for Multi-Step Tasks: Stepwise Confidence Estimation for Failure Detection

Vaibhav Mavi
Dyania Health
vaibhav@dyaniahealth.com

Shubh Jaroria
Dyania Health
shubh@dyaniahealth.com

Weiqi Sun
Dyania Health
weiqi@dyaniahealth.com

Abstract

Reliability and failure detection of large language models (LLMs) is critical for their deployment in high-stakes, multi-step reasoning tasks. Prior work explores confidence estimation for self-evaluating *LLM-scorer systems*, with confidence scorers estimating the likelihood of errors in LLM responses. However, most methods focus on single-step outputs and overlook the challenges of multi-step reasoning. In this work, we extend self-evaluation techniques to multi-step tasks, testing two intuitive approaches: holistic scoring and step-by-step scoring. Using two multi-step benchmark datasets, we show that stepwise evaluation generally outperforms holistic scoring in detecting potential errors, with up to 15% relative increase in AUC-ROC. Our findings demonstrate that self-evaluating LLM systems provide meaningful confidence estimates in complex reasoning, improving their trustworthiness and providing a practical framework for failure detection.

1 Introduction

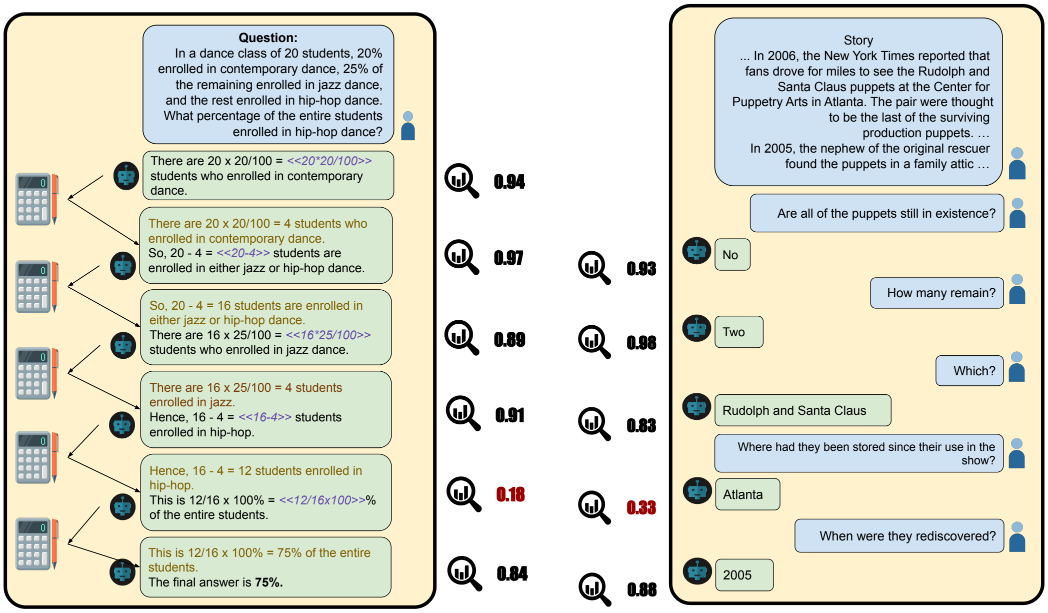
Large language model (LLM) agents are increasingly deployed in complex applications such as task-planning [29], dialog systems [28], collaborative problem-solving [24] and multi-hop question answering [16] where detecting errors and failures is a critical challenge. A common strategy for detecting failures is to extend the system with a self-evaluation component, where either the agent itself or an auxiliary evaluator assigns a confidence score to the response [7].

Failure detection through confidence estimation has been extensively studied in single-step prediction tasks [7, 26, 18, 2], but its role in multi-step reasoning remains largely underexplored. Multi-step interactions pose unique challenges: reasoning chains can be arbitrarily long, errors may occur at any step, and later steps often depend on earlier ones. Consequently, direct application of existing methods often fails to identify errors in a multi-step task reliably. For example, self-certainty [12] directly applied to CoQA (Conversational Question Answering) [19] yields poor performance (AUC-ROC 0.523, FPR@0.9 recall 0.95).

However, a trivial extension of detecting errors after each step improves the performance substantially (AUC-ROC **0.849**, FPR@0.9 recall **0.374**). This observation raises a key question: Should confidence estimation methods in multi-step tasks evaluate responses: i) after each reasoning step, enabling fine-grained error-detection, or ii) holistically, considering the final answer in full context?

We systematically investigate this question across two representative settings: tool-enhanced reasoning and LLM–user dialog. Our experiments reveal that step-level evaluation often provides superior error detection, though holistic evaluation can still be advantageous in certain contexts.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Multi-Turn Interactions in Large Language Models.



Question:
In a dance class of 20 students, 20% enrolled in contemporary dance, 25% of the remaining enrolled in jazz dance, and the rest enrolled in hip-hop dance. What percentage of the entire students enrolled in hip-hop dance?

There are 20 x 20/100 = <<20*20/100>> students who enrolled in contemporary dance.

There are 20 x 20/100 = 4 students who enrolled in contemporary dance.
So, 20 - 4 = <<20-4>> students are enrolled in either jazz or hip-hop dance.

So, 20 - 4 = 16 students are enrolled in either jazz or hip-hop dance.
There are 16 x 25/100 = <<16*25/100>> students who enrolled in jazz dance.

There are 16 x 25/100 = 4 students enrolled in jazz.
Hence, 16 - 4 = <<16-4>> students enrolled in hip-hop.

Hence, 16 - 4 = 12 students enrolled in hip-hop.
This is 12/16 x 100% = <<12/16x100>>% of the entire students.

This is 12/16 x 100% = 75% of the entire students.
The final answer is **75%.**

(a) step-level confidence scoring setup with GSM8K

Story
... In 2006, the New York Times reported that fans drove for miles to see the Rudolph and Santa Claus puppets at the Center for Puppetry Arts in Atlanta. The pair were thought to be the last of the surviving production puppets. …
In 2005, the nephew of the original rescuer found the puppets in a family attic …

Are all of the puppets still in existence?

No

How many remain?

Two

Which?

Rudolph and Santa Claus

Where had they been stored since their use in the show?

Atlanta

When were they rediscovered?

2005

(b) step-level confidence scoring setup with CoQA

2 Related Work

Prior work on confidence estimation for LLMs can be broadly divided into black-box and white-box approaches. Black-box methods assume no access to the underlying model and often rely on prompting-based strategies such as self-reflection, self-consistency, or generating multiple candidate answers [20, 25, 14, 21, 11, 30]. Another line of work employs external evaluators to assess responses post hoc, using features such as similarity measures or structured scoring models [15, 10, 18, 3].

White-box methods, in contrast, target open-source models where full access to parameters and activations is possible. These techniques include fine-tuning models to improve self-evaluation abilities [9, 22], leveraging log-probabilities for calibrated confidence estimation [23, 6, 12], and training regression models over hidden states to predict correctness [2, 4, 1].

Geng et al. [7] provide a comprehensive survey of these approaches, organizing calibration techniques across settings and highlighting key limitations. However, the vast majority of existing work studies single-step tasks, where the model outputs a single response to a single query. Extending these methods to multi-step reasoning remains largely unaddressed.

3 Problem Definition

We define failure detection as the task of estimating the probability $p \in [0, 1]$ that an agent's (A) response R to a given input I is incorrect: $p = \mathcal{F}(R \mid \text{I})$.

Multi-step interactions require a more general formulation: the input is defined as $I = (C, Q)$ where C denotes the initial context and Q is the sequence of queries. The LLM agent produces a sequence of responses R with interaction length $n = |Q| = |R|$. We consider the following two adaptations:

Response-level scoring: Treat the queries and responses in all the steps as a single sequence and assign one confidence score to the whole solution. This holistic approach captures global coherence.

$$p = \mathcal{S}_{whole}(R_{[1:n]} \mid C, Q_{[1:n]}) \tag{1}$$

Step-level scoring: The response R_i at a given step i is dependent on the prior queries and responses and scoring it requires all of the previous context $(C, Q_{[1:i]}, R_{[1:i-1]})$. Accordingly, we assign a separate score to each response R_i, conditioned on the previous queries and responses.

$$p_i = \mathcal{F}_{step}(R_i \mid C, Q_{[1:i]}, R_{[1:i-1]}) \tag{2}$$

If any individual score exceeds the threshold, the entire response can be flagged as potentially incorrect. This can be achieved by using $p = min(\{p_i\}_{i=1}^n)$.

4 Data

4.1 Agent Inputs

To test error detection, we focus on tasks where correctness can be objectively defined at each step. Accordingly, we select the following datasets.

GSM8K (Grade School Math - 8K) [5] is a collection of grade-school math word problems that require multi-step reasoning and computation. At each step, the agent generates an intermediate formula, queries an expression evaluator, and incorporates the tool's response into subsequent steps (Figure 1a). Problems in the GSM8K test set require an average of 5.1 steps.

CoQA (Conversational Question Answering) [19] contains over 127,000 question–answer pairs spanning 8,000 conversations that are context-grounded, with later questions often depending on previous queries and answers (Figure 1b). Conversations have an average of 13.5 steps.

Responses:
For both tasks, we fine-tune Llama-3.2-11B-Instruct [8] for two epochs. Because several confidence scoring methods require training, we hold out subsets of training and test splits for confidence estimation, while using the remainder to train the LLM agent. For GSM8K specifically, we use two sets of labels: Answer labels — assess whether the final answer matches the ground truth, and Reasoning labels — assess whether each intermediate reasoning step is correct. Further details on response labels and accuracy are included in Appendix A.

5 Experiments

5.1 Confidence Estimation Methods

We evaluate several confidence scoring methods, under both formulations response-level (Equation 1) and step-level (Equation 2). For methods requiring training, we use the instruction-tuned Llama-3.2-11B as the base model, replacing its generation head with a regression head for classification objectives. Details on each algorithm are mentioned in the Appendix C

5.2 Evaluation Metrics

We frame error detection as a binary classification task and report the following metrics:

AUC-ROC (Area Under the Receiver Operating Characteristic Curve): Measures how well the model separates correct from incorrect responses across thresholds. Higher is better.

FPR@0.9 Recall: Since the goal is to reliably flag potentially incorrect responses while minimizing false alarms, we measure the false positive rate (FPR) of the model at a threshold where it identifies the incorrect responses with at least 0.9 recall. Some approaches fail to reach the target recall without trivially classifying all responses as incorrect. In these cases, we report FPR@0.9 recall as 1 and additionally report the maximum achievable recall.

5.3 Results

Failure detection in multi-step interactions: For both tasks, the best performing methods achieve an AUC-ROC of 0.9 and a recall of 0.9 with FPR below one-third. Across techniques, regression model performs the best for both tasks. Interestingly, preference-based reward models perform poorly, suggesting that PRMs are better suited for ranking responses by quality, rather than tasks that have objective correctness labels [27].

Performance across granularity and task: For CoQA, step-level scoring significantly outperforms response-level scoring across all methods. For GSM8K, the difference is smaller and trends are less consistent. Notably, the step-level performance of self-certainty is significantly worse, likely due to tool interactions that alter the agent's responses at each step, thereby distorting the logits. This

Table 1: Evaluation results of different techniques on GSM8K and CoQA. An FPR@0.9 Recall of 1.0 (mr: *x*) means that the recall does not exceed *x* without flagging everything as low confidence.

		GSM8K		**CoQA**	
Technique	**granularity**	AUC (↑)	FPR@0.9 rec (↓)	AUC (↑)	FPR@0.9 rec (↓)
Self-verbalized	response	0.556	1.0 (mr: 0.13)	0.502	1.0 (mr: 0.77)
	step	0.546 (-0.2%)	1.0 (mr: 0.10)	0.624 (+24%)	0.587
Llama-3.2-11B	response	0.586	1.0 (mr: 0.52)	0.522	1.0 (mr: 0.73)
	step	0.676 (+15%)	1.0 (mr: 0.75)	0.613 (+12%)	0.81
GPT-4.1-mini	response	0.880	1.0 (mr: 0.81)	0.548	0.88
	step	0.670 (-24%)	1.0 (mr: 0.48)	0.665 (+21%)	0.476
Regression	response	0.843	0.441	0.689	0.732
	step	**0.907** (+7%)	**0.314**	**0.952** (+38%)	**0.169**
PRM	response	0.450	0.928	0.381	1.0 (mr: 0.57)
	step	-	-	0.493 (+30%)	0.887
Self-certainty	response	0.649	0.812	0.523	0.95
	step	0.395 (-40%)	0.945	0.849 (+62%)	0.374
Activations	response	0.608	1 (mr: 0.77)	0.792	0.643
	query	0.750 (+23%)	0.647	0.919 (+16%)	**0.169**

Table 2: Recall for cases with incorrect reasoning steps but correct answer. Higher the better

	Self-eval	Llama-3.2-11B	GPT-4.1-mini	Regression	Self-certainty	Activation
response	0.05	0.133	0.50	0.367	0.133	0.167
step	0	0.40	0.30	**0.60**	0.217	0.267

degradation is not observed for the activations-based regressor, since it only relies on hidden states from the final token.

Most techniques perform better on CoQA than GSM8K, suggesting that reasoning-intensive math problems are more challenging for evaluators than context-grounded QA. Interestingly, GPT-4.1-mini shows significantly improved performance on GSM8K, reflecting its superior reasoning ability.

Relation to final answer accuracy: For GSM8K, the agent reached the correct answer despite flawed intermediate reasoning in **60/879** test cases (Figure 2). Table 3 shows that step-level performance of all methods against final answers is slightly lower, while the response-level performance improves. This is expected since step-level scoring penalizes intermediate mistakes more strongly, while response-level scoring focuses on the overall outcome.

Identifying cases where the agent reaches the correct answer through flawed reasoning is crucial for trustworthy deployment. Table 2 shows that for most methods, step-level scoring is more effective at detecting such cases.

Case study on real world data We also test the effectiveness of this approach on a private dataset with real clinical notes and questions. Consistent with the analysis on public datasets, a regression model generating step-level scores achieves the best performance with **AUC-ROC** of $= \mathbf{0.940}$ and **FPR@0.9 rec** $= \mathbf{0.152}$. We include further details in Appendix D.

6 Conclusion

We extended confidence estimation to multi-step tasks in dialogue and tool-assisted reasoning, where maintaining consistency across steps is especially challenging. Through experiments on two multi-step tasks, we find that step-level scoring, though harder to implement in some cases, generally improves error detection and reveals when correct answers emerge from faulty reasoning. Our study highlights the limits of current methods and provides a basis for developing confidence estimators better suited to multi-step reasoning.

References

[1] Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying, 2023. URL https://arxiv.org/abs/2304.13734.

[2] Mohammad Beigi, Ying Shen, Runing Yang, Zihao Lin, Qifan Wang, Ankith Mohan, Jianfeng He, Ming Jin, Chang-Tien Lu, and Lifu Huang. InternalInspector i^2: Robust confidence estimation in LLMs through internal states. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12847–12865, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.751. URL https://aclanthology.org/2024.findings-emnlp.751/.

[3] Debarun Bhattacharjya, Balaji Ganesan, Junkyu Lee, and Radu Marinescu. Assessing confidence in large language models by classifying task correctness using similarity features. In *ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI*, 2025. URL https://openreview.net/forum?id=DirbdPbGhv.

[4] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision, 2024. URL https://arxiv.org/abs/2212.03827.

[5] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

[6] Gianluca Detommaso, Martin Bertran, Riccardo Fogliato, and Aaron Roth. Multicalibration for confidence scoring in llms, 2024. URL https://arxiv.org/abs/2404.04689.

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

[8] Aaron Grattafiori, Abhimanyu Dubey, and ... The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[9] Haixia Han, Tingyun Li, Shisong Chen, Jie Shi, Chengyu Du, Yanghua Xiao, Jiaqing Liang, and Xin Lin. Enhancing confidence expression in large language models through learning from past experience, 2024. URL https://arxiv.org/abs/2404.10315.

[10] Jinyi Han, Tingyun Li, Shisong Chen, Jie Shi, Xinyi Wang, Guanglei Yue, Jiaqing Liang, Xin Lin, Liqian Wen, Zulong Chen, and Yanghua Xiao. Mind the generation process: Fine-grained confidence estimation during llm generation, 2025. URL https://arxiv.org/abs/2508.12040.

[11] 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, 2022. URL https://arxiv.org/abs/2207.05221.

[12] Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty, 2025. URL https://arxiv.org/abs/2502.18581.

[13] Alex Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks, 2016. URL https://arxiv.org/abs/1610.09038.

[14] Moxin Li, Wenjie Wang, Fuli Feng, Fengbin Zhu, Qifan Wang, and Tat-Seng Chua. Think twice before trusting: Self-detection for large language models through comprehensive answer reflection, 2024. URL https://arxiv.org/abs/2403.09972.

[15] Yukun Li, Sijia Wang, Lifu Huang, and Li-Ping Liu. Graph-based confidence calibration for large language models, 2025. URL https://arxiv.org/abs/2411.02454.

[16] Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. Multi-hop question answering. *Foundations and Trends® in Information Retrieval*, 17(5):457–586, 2024. ISSN 1554-0669. doi: 10.1561/1500000102. URL http://dx.doi.org/10.1561/1500000102.

[17] OpenAI, Josh Achiam, and ... Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[18] Tejaswini Pedapati, Amit Dhurandhar, Soumya Ghosh, Soham Dan, and Prasanna Sattigeri. Large language model confidence estimation via black-box access, 2025. URL https://arxiv.org/abs/2406.04370.

[19] Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge, 2019. URL https://arxiv.org/abs/1808.07042.

[20] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36 – 37th Conference on Neural Information Processing Systems, NeurIPS 2023*, Advances in Neural Information Processing Systems. Neural Information Processing Systems Foundation, 2023.

[21] Vaishnavi Shrivastava, Ananya Kumar, and Percy Liang. Language models prefer what they know: Relative confidence estimation via confidence preferences, 2025. URL https://arxiv.org/abs/2502.01126.

[22] Paul Stangel, David Bani-Harouni, Chantal Pellegrini, Ege Özsoy, Kamilia Zaripova, Matthias Keicher, and Nassir Navab. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models, 2025. URL https://arxiv.org/abs/2503.02623.

[23] Yi-Jyun Sun, Suvodip Dey, Dilek Hakkani-Tur, and Gokhan Tur. Confidence estimation for llm-based dialogue state tracking, 2024. URL https://arxiv.org/abs/2409.09629.

[24] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. Multi-agent collaboration mechanisms: A survey of llms, 2025. URL https://arxiv.org/abs/2501.06322.

[25] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL https://arxiv.org/abs/2203.11171.

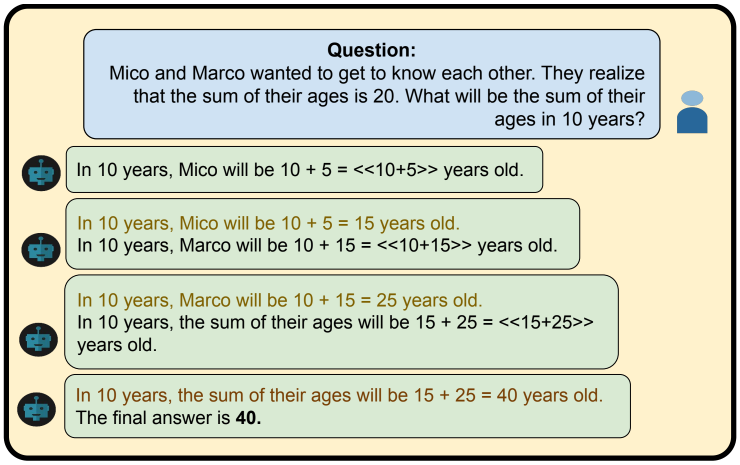
[26] Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms, 2024. URL https://arxiv.org/abs/2306.13063.

[27] Yuhui Xu, Hanze Dong, Lei Wang, Caiming Xiong, and Junnan Li. Reward models identify consistency, not causality, 2025. URL https://arxiv.org/abs/2502.14619.

[28] Zihao Yi, Jiarui Ouyang, Zhe Xu, Yuwen Liu, Tianhao Liao, Haohao Luo, and Ying Shen. A survey on recent advances in llm-based multi-turn dialogue systems, 2025. URL https://arxiv.org/abs/2402.18013.

[29] Wenshuo Zhai, Jinzhi Liao, Ziyang Chen, Bolun Su, and Xiang Zhao. A survey of task planning with large language models. *Intelligent Computing*, 4:0124, 2025. doi: 10.34133/icomputing.0124. URL https://spj.science.org/doi/abs/10.34133/icomputing.0124.

[30] Ziang Zhou, Tianyuan Jin, Jieming Shi, and Qing Li. Steerconf: Steering llms for confidence elicitation, 2025. URL https://arxiv.org/abs/2503.02863.



Question:
Mico and Marco wanted to get to know each other. They realize that the sum of their ages is 20. What will be the sum of their ages in 10 years?

In 10 years, Mico will be 10 + 5 = <<10+5>> years old.

In 10 years, Mico will be 10 + 5 = 15 years old.
In 10 years, Marco will be 10 + 15 = <<10+15>> years old.

In 10 years, Marco will be 10 + 15 = 25 years old.
In 10 years, the sum of their ages will be 15 + 25 = <<15+25>> years old.

In 10 years, the sum of their ages will be 15 + 25 = 40 years old.
The final answer is **40.**

Figure 2: Case from GSM8K where the agent gets the answer correct through incorrect reasoning steps. The agent assumes the current ages of Mico and Marco to be 5 and 15 while the question does not mention it. The agent ends up getting to the correct answer nonetheless since it only concerns with the sum of their ages.

A Training details

A.1 Confidence Scorers: Teacher Forcing

For methods requiring supervised training, we adopt teacher forcing [13]. During training, the model receives the gold history (i.e., corrected responses) when evaluating the next response. The learning objective is:

$$p_i = \mathcal{F}(R_i, |C, Q_{[1:i]}, \hat{R}_{[1:i-1]}) = \mathbb{I}\{R_i \neq \hat{R}_i\} \tag{3}$$

where $\hat{R}$ is the list of ground truth responses and $\mathbb{I}\{.\}$ is the indicator function. During inference, we do not assume access to the ground truth. At inference time, however, no ground truth is available, and the evaluator must operate solely on the model's predictions.

B Data preparation

B.1 Agent: Training and Inference

For both tasks, we fine-tune Llama-3.2-11B-Instruct [8] for two epochs. Because several confidence scoring methods require training, we hold out subsets of train and test splits for confidence estimation, while using the remainder to train the LLM agent.

Performance varies across datasets and granularity: - On CoQA, the agent achieves$\mathbf{81}.\mathbf{2}\%$ step-level accuracy but only $\mathbf{16}.\mathbf{1}\%$ response-level accuracy. The large gap is expected, since even a single incorrect step can propagate errors downstream. - On GSM8K, the agent achieves $\mathbf{65}.\mathbf{6}\%$ answer accuracy and $\mathbf{47}.\mathbf{6}\%$ step-level accuracy. Here, answer accuracy is higher because the agent may arrive at correct final answers even if some intermediate steps are flawed (see Figure 2).

B.2 Labeling responses

We use GPT-5 to evaluate agent's responses against ground truth answers and intermediate steps, producing labels at both the step-level and response-level. To verify the label quality, we manually reviewed 100 samples from each dataset. We found labeling accuracy above 96% in both settings.

Table 3: Answer label performance on **GSM8K**. An FPR@0.9 Recall of 1.0 (mr: x) means that the recall does not exceed x without flagging everything as low confidence.

Technique	**granularity**	**AUC-ROC** (↑)	**ECE** (↓)	**FPR@0.9 Recall** (↓)
Self-eval	response	0.560	0.317	1.0 (mr: 0.15)
	step	0.559 (-0.2%)	0.3125	1.0 (mr: 0.12)
Llama-3.2-11B	response	0.590	0.291	1.0 (mr: 0.52)
	step	0.669 (+13%)	0.159	1.0 (mr: 0.76)
GPT-4.1-mini	response	**0.895**	0.088	1.0 (mr: 0.88)
	step	0.662 (-26%)	0.280	1.0 (mr: 0.49)
Regression	response	0.869	**0.075**	0.4385
	step	0.872 (+1%)	0.144	**0.369**
PRM	response	0.460	0.629	0.915
	step	-	-	-
Self-certainty	response	0.658	0.219	0.773
	step	0.342 (-48%)	0.320	0.958
Activations	response	0.605	0.339	1 (mr: 0.77)
	query	0.738 (+21%)	0.279	0.655

C Evaluated Confidence Estimation Methods

C.1 Black-box methods

C.1.1 Self-verbalized confidence

The LLM agent is prompted to verbalize its confidence in its own response. For step-level scoring, the agent outputs the confidence score at the end of each step.

C.1.2 Auxiliary evaluators

External models assess the agent's responses.

Pre-trained LLMs: Instruction-tuned LLMs are prompted to evaluate the agent's responses. We consider two evaluators: (a) Llama-3.2-11B (aligned with the agent's base model), and (b) OpenAI's GPT-4.1-mini [17] (independent of the agent).

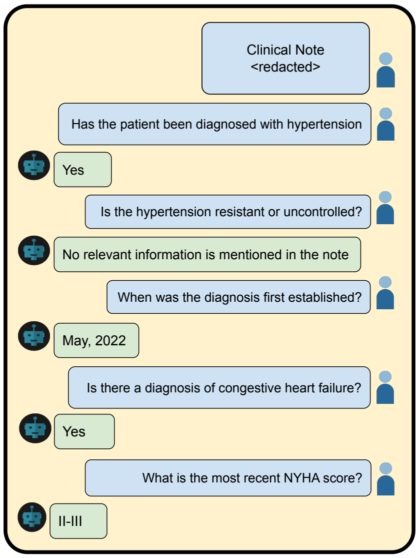
Regression model: We fine-tune a sequence classification model to regress confidence scores in the range $[0, 1]$.

Preference-based reward model (PRM): We train a reward model on preference data, treating completions with correct answers as "chosen" and incorrect agent outputs as "rejected." For GSM8K, multiple valid reasoning paths to solve the same problem make generating step-level preference data infeasible, since each incorrect step in the interaction would require a corrected version. Hence, we evaluate PRMs only at the response-level.

C.2 White-box methods

Logits: Following **Self-certainty** [12], we compute the KL divergence of the agent's output logits from the uniform distribution as a measure of certainty. Since this approach consistently outperforms other logit-based methods, we use it as the representative logit-based white-box baseline. Self-certainty scores are normalized to fall within $[0, 1]$.

Activations: Prior work [2, 4, 1] suggests that hidden states of the model's final LLM layer contain information on model's behavior and can be used to extract its confidence in its response. Following this, we train a 5-layer MLP classifier on the model's final hidden states to predict a correctness score.



Clinical Note
<redacted>

Has the patient been diagnosed with hypertension

Yes

Is the hypertension resistant or uncontrolled?

No relevant information is mentioned in the note

When was the diagnosis first established?

May, 2022

Is there a diagnosis of congestive heart failure?

Yes

What is the most recent NYHA score?

II-III

Figure 3: An example from the private clinical data.

D Evaluating on private dataset

To evaluate the applicability of confidence estimation methods in real-world settings, we tested some of the approaches on a private dataset consisting of conversational question-answering interactions over real patient clinical notes. A redacted example from this dataset is provided in Figure 3. We do not publicly release the data due to conflict of interest as well as HIPAA compliance, and the results are therefore not reproducible. Nevertheless, it provides a valuable demonstration in a domain where trustworthiness is critical. Consistent with the analysis on public datasets, a regression model generating step-level scores achieves the best performance with **AUC-ROC** $= \mathbf{0.940}$ and **FPR@0.9 rec** $= \mathbf{0.152}$. These results indicate that step-level confidence scoring with a regression model remains effective in complex, real-world interactions.