FAILURE ATTRIBUTION WITH STEPWISE UNCERTAINTY IN MULTI AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Failure attribution in agent systems aims to identify the steps that critically contribute to task failure. Existing approaches either rely on large language models (LLMs) as judges, incurring substantial inference costs, or require additional data and training to build dedicated attribution models. We propose a training-free method for failure attribution that identifies critical failure steps using the outputs of a fixed, lightweight observer model. We find that decisive failure steps leave a measurable signature in the observer's continuous uncertainty. Conventional binary readouts that collapse predictions into binary outcomes discard relative uncertainty across steps, but the proposed method directly uses the observer model's pre-threshold uncertainty as a continuous attribution score, identifying the first step whose uncertainty enters a high-uncertainty regime as the critical failure step. It requires no modification to the observer model or its inference procedure, and introduces no additional training data, step-level failure labels, or auxiliary evaluation models. Recovering the continuous information enables fine-grained, step-level failure attribution at minimal additional cost. Experiments on the Who & When benchmark demonstrate that this attribution signal persists across multiple unsupervised localization rules, and that the proposed method outperforms closed-model LLM-as-a-Judge baselines by up to 10.3%p, while avoiding $2.11-$14.44 in frontier-model API costs per corpus without requiring any additional model calls.

1 INTRODUCTION

As multi-agent systems (MAS) become increasingly capable, understanding why they fail is as important as measuring whether they succeed (Hong et al., 2024; Cemri et al., 2025). Failure attribution is to identify the agents and steps that critically contribute to unsuccessful trajectories, providing information for diagnosing and improving system failures (Zhang et al., 2025). Existing approaches either use LLM judges to attribute failures from execution trajectories (Zheng et al., 2023) or train dedicated attribution models on labeled failure trajectories (Zhang et al., 2026). However, it remains challenging to accurately identify critical failure steps (Zhang et al., 2025), and existing approaches often incur additional inference or training costs or require fine-grained failure annotations (Zhang et al., 2026; 2025; Cemri et al., 2025).

In this paper, we explore observer uncertainty as a signal for failure attribution. Prior work on LLM uncertainty estimation shows that uncertainty reflects prediction reliability and can help identify failures (Xiong et al., 2024; Xie et al., 2025; Sheng et al., 2025). Recent work extends uncertainty estimation to multi-step reasoning, showing that uncertainty at intermediate steps can reflect the reliability of the reasoning process as it unfolds (Yin et al., 2024). Motivated by these findings, we investigate whether observer uncertainty can provide step-level information about failures along an execution trajectory. Instead of reducing the observer's confidence to a binary judgment, we retain its continuous uncertainty and examine whether variations across steps can localize critical failure steps.

Figure 1 illustrates our key observations. On the left, our uncertainty-based approach achieves higher agent- and step-level attribution accuracy than the closed- and open-model LLM judges shown. On the right, observer uncertainty rises around the annotated failure step and peaks immediately afterward, revealing a localized pattern that is obscured by binary judgments. This pattern indicates

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

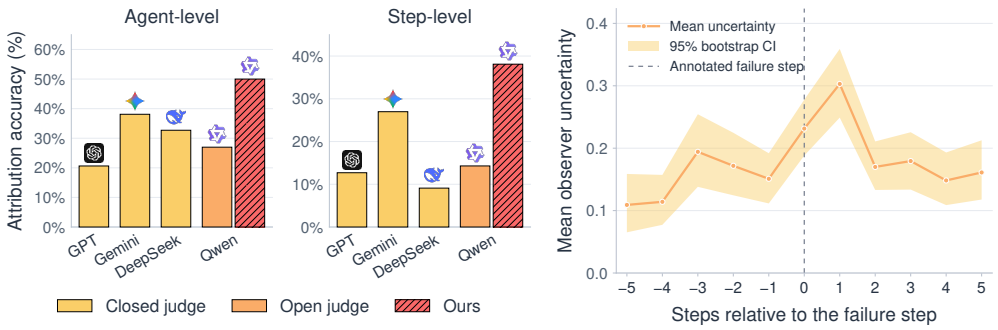

Figure 1: **Attribution performance and failure-aligned uncertainty.** Left: Agent- and step-level attribution accuracy on the Algorithm Generated subset of Who & When, without ground-truth (GT) information. Right: Observer uncertainty around the annotated failure step (dashed line), with 95% bootstrap confidence intervals.

that even an observer not specifically trained for failure attribution can exhibit uncertainty that is informative about failure location. We observe similar patterns across different uncertainty measures and localization methods.

Based on this observation, we use the continuous uncertainty of a fixed observer to identify critical failure steps. At each step, the observer evaluates the execution trajectory up to that point, and we retain the continuous information underlying its judgment rather than relying solely on the binary output. This yields a step-wise failure-evidence signal, which is combined with an unsupervised localization rule to identify the critical failure step. In our implementation, we use $(p(\mathrm{True}))$ to derive the uncertainty signal and Otsu thresholding for localization. The overall approach requires no modification to the observer, step-level failure annotations, or attribution-specific training.

We evaluate our approach on failure attribution benchmarks against LLM Judges and binary observer judgments (Zheng et al., 2023). Our approach improves step-level attribution over binary observer judgments and achieves competitive performance against LLM Judges. Ablations across different uncertainty measures and unsupervised localization methods further show that the observed gains are not tied to a particular choice of uncertainty measure or localization rule. Our main contributions are as follows:

- **Uncertainty as an attribution signal.** We show that continuous observer uncertainty contains step-level information about critical failures that is not captured by binary judgments, establishing uncertainty as a useful signal for failure attribution.
- **Training-free failure attribution.** We introduce an approach that leverages the continuous uncertainty of a fixed observer to localize critical failure steps, requiring no modification to the observer, step-level failure annotations, or attribution-specific training.
- **Robustness across uncertainty and localization choices.** We demonstrate that the effectiveness of uncertainty-based attribution persists across different uncertainty measures and unsupervised localization methods, indicating that the observed benefit is not tied to a particular implementation choice.

2 Related Work

Failure attribution in multi-agent systems. Failure attribution in LLM-based multi-agent systems (MAS) aims to identify the agent and execution step responsible for task failure. Who&When (Zhang et al., 2025) formalized this task as responsible-agent identification and decisive-step localization. Existing approaches broadly fall into non-learning and learning-based paradigms. Non-learning methods rely on LLM-based trajectory analysis, including full-trajectory inspection, sequential or search-based localization, and scope narrowing (Zhang et al., 2025; Sun et al., 2026). Learning-based methods instead train dedicated attribution models on labeled failure trajectories, using automatically constructed supervision and specialized training objectives (Zhang et al., 2026;

Kong et al., 2026). Both paradigms introduce additional overhead.Non-learning methods require costly inference, while learning-based methods require attribution-specific data and training. This motivates failure attribution methods that avoid both sources of overhead.

Uncertainty estimation in LLMs. LLM uncertainty provides a continuous signal of prediction reliability beyond discrete model outputs. Prior work primarily studies uncertainty at the response level, using token probabilities, entropy-based measures, semantic uncertainty, or self-reported confidence to assess prediction correctness and calibration (Malinin & Gales, 2021; Kuhn et al., 2023; Kadavath et al., 2022; Tian et al., 2023). Sampling-based approaches similarly estimate uncertainty from consistency across multiple responses (Manakul et al., 2023). Recent work extends uncertainty estimation beyond final responses to intermediate states in multi-step reasoning and agentic execution. Step-level uncertainty has been used to characterize the reliability of evolving reasoning processes (Yin et al., 2024), while SAUP propagates uncertainty across sequential agent steps to assess agent reliability (Zhao et al., 2025). These studies suggest that uncertainty can provide information not only at the final output but also throughout a sequential process. Our work builds on this perspective but focuses on failure attribution. Rather than using uncertainty to assess overall response or trajectory reliability, we examine whether variations in a fixed observer's continuous uncertainty can localize critical failure steps. In particular, we recover the continuous information underlying binary observer judgments and use it as a step-level attribution signal, without additional uncertainty estimation or model training.

3 METHOD

3.1 PROBLEM FORMULATION

We consider a multi-agent system that produces a trajectory of T sequential steps,

$$\tau = (z_1, z_2, \ldots, z_T)\,, \tag{1}$$

where each step $z_t = (a_t, x_t)$ consists of the acting agent a_t and its generated action, message, or tool interaction x_t. We focus on trajectories that ultimately result in task failure. For a failed trajectory, let $t^\star \in \{1, \ldots, T\}$ denote the decisive failure step, defined as the earliest erroneous step that critically contributes to the final task failure (Zhang et al., 2025). To avoid using future consequences when evaluating an earlier step, a fixed observer $\mathcal{O}$ assesses step t using only the trajectory prefix

$$\tau_{\le t} = (z_1, \ldots, z_t)\,. \tag{2}$$

Failure attribution is then formulated as estimating

$$\hat{t} = \mathcal{A}\left(\{\mathcal{O}(\tau_{\le t})\}_{t=1}^{T}\right), \tag{3}$$

where $\mathcal{A}$ denotes the attribution procedure. We require $\mathcal{O}$ to remain fixed and $\mathcal{A}$ to operate without step-level failure labels or attribution-specific training on the target trajectories.

3.2 METHOD OVERVIEW

Figure 2 summarizes our approach. For each step, a fixed observer evaluates the trajectory prefix up to that point and produces a binary judgment together with its underlying confidence. Instead of discarding this confidence after thresholding, we retain it as a continuous step-wise uncertainty score and form an uncertainty sequence over the trajectory. We then separate the sequence into low- and high-uncertainty regimes using an unsupervised localization rule. The earliest step entering the high-uncertainty regime is selected as the decisive failure step, and the agent responsible for that step is returned as the failure agent. This contrasts with binary judgments, which retain only discrete outcomes, and full-trajectory LLM judges, which directly predict the failure location from the execution trace.

3.3 STEP-WISE OBSERVER UNCERTAINTY

For each step t, we query a fixed observer using only the trajectory prefix $\tau_{\le t}$ and ask whether the current step is correct given the preceding interaction history. Rather than using the observer's

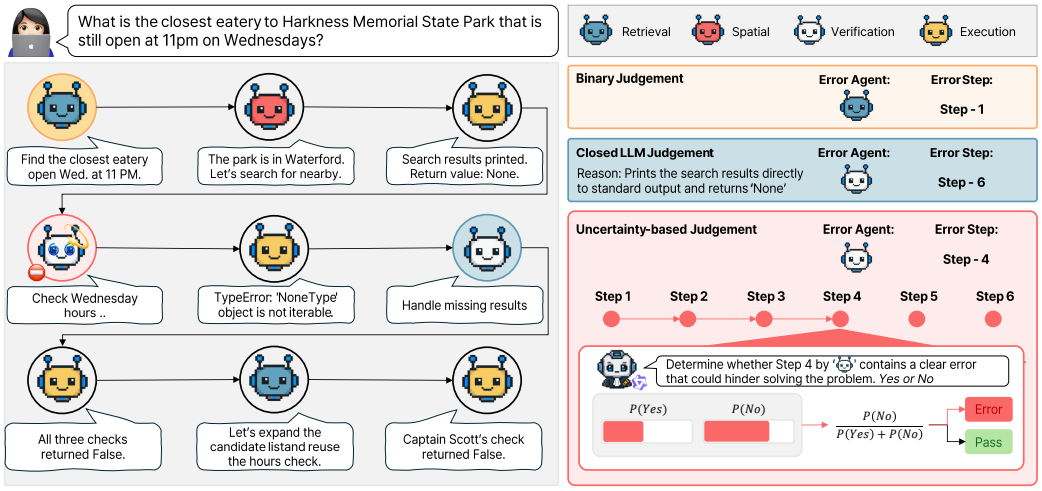

Figure 2: **Overview of failure attribution in a multi-agent trajectory.** The proposed method preserves the observer's continuous uncertainty and identifies the earliest step entering the high-uncertainty regime, in contrast to binary and LLM-based judgments.

discrete verdict, we retain its confidence over the two candidate outcomes, `true` and `false`. Let $\mathcal{S}_{\mathrm{T}}$ and $\mathcal{S}_{\mathrm{F}}$ denote the token spellings corresponding to `true` and `false`. We aggregate their next-token probability mass as

$$M_{\mathrm{T},t} = \sum_{s \in \mathcal{S}_{\mathrm{T}}} \exp(\ell_{t,s}), \qquad M_{\mathrm{F},t} = \sum_{s \in \mathcal{S}_{\mathrm{F}}} \exp(\ell_{t,s}), \tag{4}$$

where $\ell_{t,s}$ is the log-probability assigned to token spelling s. Aggregating over spelling variants accounts for tokenizer-dependent realizations, such as tokens with or without leading whitespace. We normalize the probability mass over the two outcomes as

$$p_t(\mathrm{True}) = \frac{M_{\mathrm{T},t}}{M_{\mathrm{T},t} + M_{\mathrm{F},t}}, \tag{5}$$

and define the step-wise uncertainty score as

$$U_t = 1 - p_t(\mathrm{True}) = \frac{M_{\mathrm{F},t}}{M_{\mathrm{T},t} + M_{\mathrm{F},t}}. \tag{6}$$

Larger U_t indicates stronger evidence against the correctness of step t. Unlike a binary verdict, U_t preserves relative confidence across steps, allowing us to compare failure evidence along the trajectory. We do not require U_t to be a calibrated estimate of failure probability; it is used only as a relative failure-evidence score. Applying the same observer to all trajectory prefixes yields

$$\mathbf{U}(\tau) = (U_1, U_2, \ldots, U_T), \tag{7}$$

which is used for failure localization. Our main implementation derives U_t from $p_t(\mathrm{True})$, but the method is not tied to this particular uncertainty measure. We additionally evaluate semantic entropy, negative log-likelihood (NLL), and self-consistency in Figure 6.

3.4 Uncertainty-based Failure Localization

Given the uncertainty sequence $\mathbf{U}(\tau)$, we localize the decisive failure step by separating the scores into low- and high-uncertainty regimes. Since the scale and distribution of uncertainty may vary across trajectories, we estimate the threshold independently for each trajectory.

In our implementation, we use Otsu thresholding (Otsu, 1979), which selects the threshold that maximizes between-group separation. For a candidate threshold c, let $w_0(c)$ and $w_1(c)$ denote the proportions of scores below and above c, and let $\mu_0(c)$ and $\mu_1(c)$ denote their respective means. We choose

$$c^\star = \arg\max_c w_0(c) w_1(c) \left(\mu_0(c) - \mu_1(c)\right)^2. \tag{8}$$

As illustrated in Figure 3, the resulting threshold divides the observed scores into low- and high-uncertainty regimes. We then localize the decisive failure step as the first step entering the high-uncertainty regime:

$$\hat{t} = \min \{t : U_t \geq c^\star\} . \tag{9}$$

Our approach is not tied to Otsu thresholding. We additionally evaluate a two-component Gaussian mixture model (GMM-2) (McLachlan & Peel, 2000), change-point detection (Killick et al., 2012), a 60th-percentile threshold (Pct-60), Tukey's rule (Tukey, 1977), and a median absolute deviation (MAD) rule (Hampel, 1974). Figure 4 compares failure localization performance across these unsupervised localization rules.

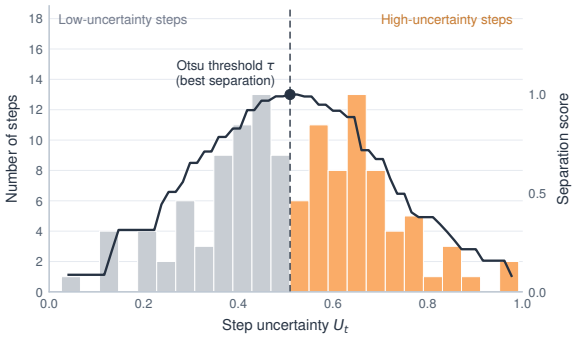

Figure 3: **Example of Otsu threshold selection.**

4 Experiments

We design our experiments to answer three questions: (i) Can a fixed observer without attribution-specific training localize failures competitively with LLM judges? (ii) Does retaining continuous uncertainty improve failure localization over binary judgments from the same observer? (iii) Does the benefit persist across localization rules and uncertainty measures?

4.1 Experimental Setup

Dataset and metrics. We evaluate on the *Algorithm-Generated* and *Hand-Crafted* subsets of Who&When (Zhang et al., 2025). We report agent-level and step-level attribution accuracy under both non-GT and GT settings.

Baselines and models. We compare against random prediction and LLM judges following the Step-by-Step protocol of Who&When (Zhang et al., 2025). The closed-model judges include seven models from the GPT, Gemini, and DeepSeek families, while the open-model judges include eight models from the Llama and Qwen families. To isolate the effect of continuous uncertainty, we additionally compare our uncertainty-based readout with the binary readout from the same observer. Our method uses seven open observers from the Llama and Qwen families.

4.2 Main Results

A training-free observer is competitive with frontier LLM judges. Table 1 shows that our method, using only open-weight observers and no API calls, achieves higher best scores than closed LLM judges in six of eight settings. It also exceeds the best open-LLM judge across all eight settings, with gains of $+5.5$ to $+19.9$ percentage points (pp). The gains are particularly large for step-level attribution on Algorithm-Generated trajectories: our method reaches 38.1% both without and with GT, compared with 27.8% and 31.0% for the best closed judge and 18.3% and 19.1% for the best open judge. Notably, strong performance does not require a large observer: Qwen3.5-9B reaches 32.5% without GT, outperforming all closed judges in this setting. Closed judges remain stronger in two settings without GT: Hand-Crafted step-level attribution (27.0% vs. 23.6%) and Algorithm-Generated agent-level attribution (60.0% vs. 50.0%).

Continuous uncertainty retains information lost by discrete verdicts. The comparison with open-LLM judges also provides a controlled test of the readout: our method uses the same underlying models and prompts, but replaces their discrete verdicts with continuous uncertainty. This change improves failure attribution in most settings. For example, Llama-3.3-70B improves from 38.2% to 65.5% on Hand-Crafted agent-level attribution without GT, while Qwen3.6-35B-A3B improves from 14.3% to 38.1% on Algorithm-Generated step-level attribution without GT. These

Table 1: **Failure attribution performance on Who&When.** Agent- and step-level accuracy (%) with and without ground-truth (GT) information. Bold indicates the best score in each setting, and Δ denotes the gain of our best result over the best closed/open LLM judge.

Type	Model	Handcraft				Automated			
		Agent-level		Step-level		Agent-level		Step-level	
		w/o GT	w/ GT	w/o GT	w/ GT	w/o GT	w/ GT	w/o GT	w/ GT
Baseline	Random	29.10	29.10	3.90	3.90	29.10	29.10	12.00	12.00
Closed LLM	GPT-4o	52.70	54.60	16.40	16.40	20.60	31.00	12.70	21.40
	GPT-4o-mini	20.60	61.80	12.70	14.60	52.70	28.60	16.40	15.90
	GPT-4.1	26.20	65.50	21.40	18.20	**60.00**	35.70	18.20	27.00
	Gemini 3.1 Flash Lite	55.20	55.20	19.00	15.50	38.90	37.30	27.00	27.00
	Gemini 3.5 Flash Lite	47.30	52.70	9.10	12.70	38.10	43.70	27.80	31.00
	DeepSeek V4 Flash	32.50	52.70	22.20	20.00	54.50	44.40	16.40	31.00
	DeepSeek V4 Pro	38.10	41.80	**27.00**	12.70	32.70	38.90	9.10	29.40
	Best Score	55.20	65.50	**27.00**	20.00	**60.00**	44.40	27.80	31.00
Open LLM	LLaMA-3.2-3B	18.20	18.20	5.50	1.80	11.90	11.10	4.00	2.40
	Llama-3.1-8B	36.40	36.40	1.80	1.80	19.10	28.60	9.50	11.90
	Llama-3.3-70B	38.20	38.20	5.50	9.10	8.70	16.70	7.90	14.30
	Qwen3-8B	49.10	41.80	14.60	9.10	17.50	15.10	11.90	8.70
	Qwen3-32B	45.50	58.20	7.30	20.00	13.50	26.20	7.90	18.30
	Qwen3.5-4B	30.90	32.70	7.30	1.80	21.40	33.30	10.30	17.50
	Qwen3.5-9B	36.40	41.80	7.30	7.30	36.50	26.20	18.30	13.50
	Qwen3.6-35B-A3B	36.40	36.40	5.50	10.90	27.00	34.90	14.30	19.10
	Best Score	49.10	58.20	14.60	20.00	36.50	34.90	18.30	19.10
Ours	Llama-3.1-8B	54.60	52.70	16.40	20.00	21.40	22.20	13.50	13.50
	Llama-3.3-70B	**65.50**	**69.10**	21.80	**25.50**	21.40	21.40	19.80	19.10
	Qwen3-8B	47.30	43.60	5.50	5.50	25.40	24.60	16.70	16.70
	Qwen3-32B	58.20	54.60	20.00	20.00	27.00	27.00	23.00	21.40
	Qwen3.5-4B	50.90	50.90	20.00	14.60	39.70	39.70	24.60	25.40
	Qwen3.5-9B	60.00	54.60	23.60	20.00	38.10	37.30	32.50	29.40
	Qwen3.6-35B-A3B	41.80	38.20	12.70	10.90	50.00	**54.80**	**38.10**	**38.10**
	Best Score	**65.50**	**69.10**	23.60	**25.50**	50.00	**54.80**	**38.10**	**38.10**
Δ Ours − Closed LLM		+10.30	+3.60	-3.40	+5.50	-10.00	+10.30	+10.30	+7.20
Δ Ours − Open LLM		+16.40	+10.90	+9.10	+5.50	+13.50	+19.80	+19.90	+19.10

within-model gains suggest that continuous uncertainty preserves attribution-relevant information that is lost when the model's judgment is reduced to a discrete verdict.

4.3 Is Continuous Uncertainty the Source of the Gain?

Retaining continuous uncertainty consistently improves over binary verdicts. Table 2 isolates the readout by comparing our method with the binary readout obtained from the *same* observer call, so the two differ only in whether the verdict is taken as a hard decision or as a continuous score. The continuous readout improves accuracy in 51 of 56 observer-setting pairs, and the average improvement is positive in all eight settings, ranging from $+5.4$pp to $+18.5$pp. On Algorithm-Generated trajectories, it improves over the binary readout for every observer in every setting, and the five negative cases on Hand-Crafted trajectories are confined to two observers. Since the observer, prompt, and inference are identical, this gain can be attributed to the information that hard thresholding discards.

Uncertainty separates failure steps even without a localization rule. Table 3 evaluates whether U_t ranks the decisive step above the other steps, independently of any threshold. Binary verdicts are close to chance, with an average AUROC of 52.3 on Hand-Crafted and 52.0 on Algorithm-Generated trajectories. The continuous score raises these to 58.7 and 57.3, improves AUROC for six of seven observers in each subset, and reaches up to 73.5 for Llama-3.3-70B on Hand-Crafted trajectories.

Table 2: **Continuous vs. binary readout.** Accuracy difference (pp) between the two readouts using the same observer and inference.

Observer	Handcraft				Automated			
	Agent-level		Step-level		Agent-level		Step-level	
	w/o GT	w/ GT	w/o GT	w/ GT	w/o GT	w/ GT	w/o GT	w/ GT
Llama-3.1-8B	+18.20	+21.80	+9.10	+10.90	+11.90	+18.30	+7.10	+11.10
Llama-3.3-70B	+30.90	+34.50	+16.40	+18.20	+11.90	+5.60	+11.90	+4.80
Qwen3-8B	+10.90	+7.30	-3.60	-5.50	+6.40	+14.30	+4.80	+10.30
Qwen3-32B	+21.80	+3.60	+9.10	+1.80	+16.70	+7.90	+16.70	+7.90
Qwen3.5-4B	+36.40	+36.40	+18.20	+12.70	+37.30	+34.90	+22.20	+21.40
Qwen3.5-9B	+14.60	+3.60	+7.30	+1.80	+15.90	+12.70	+19.10	+13.50
Qwen3.6-35B-A3B	-3.60	-9.10	+1.80	-1.80	+10.30	+3.20	+10.30	+0.80
Avg. Δ	**+18.40**	**+14.00**	**+8.30**	**+5.50**	**+15.80**	**+13.80**	**+13.20**	**+10.00**

Table 3: Failure-step AUROC (%) for binary and continuous uncertainty scores.

Observer	Hand-Crafted		Algorithm-Generated	
	Binary	Continuous	Binary	Continuous
Llama-3.1-8B	46.9	**48.0**	**46.0**	40.3
Llama-3.3-70B	50.0	**73.5**	49.7	**59.3**
Qwen3-8B	49.8	**52.7**	48.8	**58.7**
Qwen3-32B	52.1	**52.7**	52.0	**54.3**
Qwen3.5-4B	54.5	**67.0**	53.2	**58.3**
Qwen3.5-9B	60.9	**71.6**	59.9	**62.2**
Qwen3.6-35B-A3B	**51.7**	45.3	54.6	**68.0**
Mean	52.3	**58.7**	52.0	**57.3**

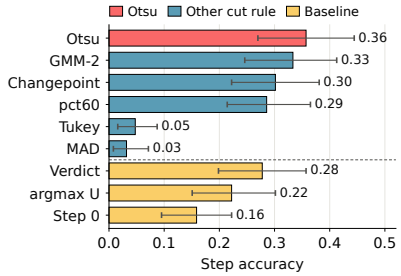

Figure 4: Step-level failure localization across unsupervised localization rules.

This confirms that the attribution signal resides in the uncertainty itself rather than in a particular decision procedure.

4.4 Robustness Analysis

Failure localization is robust to the choice of cut rule. Figure 4 examines whether step-level failure localization depends on a particular rule for converting continuous uncertainty into a discrete failure location. Across several alternative cut rules, including GMM-2, changepoint detection, and percentile-based thresholding (pct60), we observe comparable step-level accuracy (29–33%) to Otsu (36%). These alternatives also remain competitive with or outperform the baseline localization rules, such as verdict (28%), argmax U (22%), and Step 0 (16%). While not every statistical rule is equally effective—Tukey and MAD perform poorly in this setting—the results indicate that the localization benefit is not tied to Otsu specifically, but persists across multiple reasonable ways of discretizing the continuous uncertainty signal.

P(True) provides a strong and efficient uncertainty measure. Figure 6 compares P(True) with semantic entropy (Kuhn et al., 2023), negative log-likelihood (NLL), and self-consistency (Manakul et al., 2023) across three observer models. P(True) achieves the best or tied-best step-level attribution score for all three observers, with the largest margin on Qwen3.6-35B-A3B (0.325 vs. 0.270 or lower). Moreover, P(True) requires only a single forward pass per step, whereas semantic entropy and self-consistency require multiple samples. This makes P(True) a strong and computationally efficient default.

The benefit holds across localization rules. Table 4 compares Otsu with a percentile rule calibrated on ALFWorld and HotpotQA. Both continuous rules outperform the binary readout across all

Rule	Observer	Handcraft		Automated	
		w/o GT	w/ GT	w/o GT	w/ GT
Binary	Llama3.1-8B	7.30	9.10	6.40	2.40
	Llama3.3-70B	5.50	7.30	7.90	14.30
	Qwen3-8B	9.10	10.90	11.90	6.40
	Qwen3-32B	10.90	18.20	6.40	13.50
	Qwen3.5-4B	1.80	1.80	2.40	4.00
	Qwen3.5-9B	16.40	18.20	13.50	15.90
	Qwen3.6-35B-A3B	10.90	12.70	27.80	37.30
	Avg.	8.80	11.20	10.90	13.40
Percentile	Llama3.1-8B	9.10	7.30	18.30	13.50
	Llama3.3-70B	14.60	16.40	13.50	19.10
	Qwen3-8B	5.50	9.10	27.00	24.60
	Qwen-32B	16.40	16.40	20.60	23.00
	Qwen3.5-4B	7.30	21.80	26.20	29.40
	Qwen3.5-9B	18.20	23.60	21.40	25.40
	Qwen3.6-35B-A3B	9.10	16.40	32.50	37.30
	Avg.	11.40	15.80	22.80	**24.60**
Otsu	Llama3.1-8B	16.40	20.00	13.50	13.50
	Llama3.3-70B	21.80	25.50	19.80	19.10
	Qwen3-8B	5.50	5.50	16.70	16.70
	Qwen3-32B	20.00	20.00	23.00	21.40
	Qwen3.5-4B	20.00	14.60	24.60	25.40
	Qwen3.5-9B	23.60	20.00	32.50	29.40
	Qwen3.6-35B-A3B	12.70	10.90	38.10	38.10
	Avg.	**17.10**	**16.60**	**24.00**	23.40

Table 4: **Robustness across unsupervised localization rules.** Step-level attribution under Binary, Percentile, and Otsu localization.

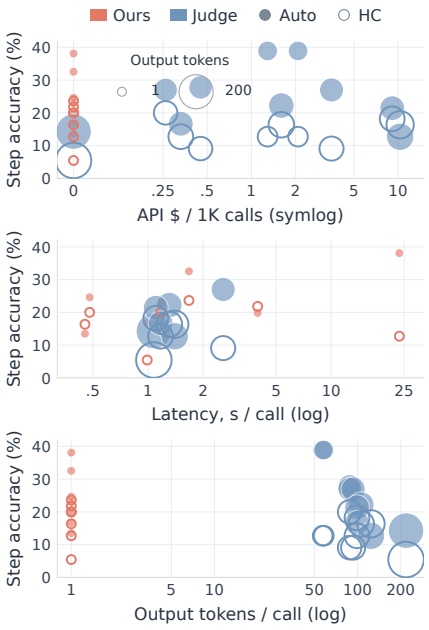

Figure 5: **Inference cost versus step-level attribution performance.** Comparison of our method and LLM judges across API cost, latency, and output length on the Auto and HC subsets.

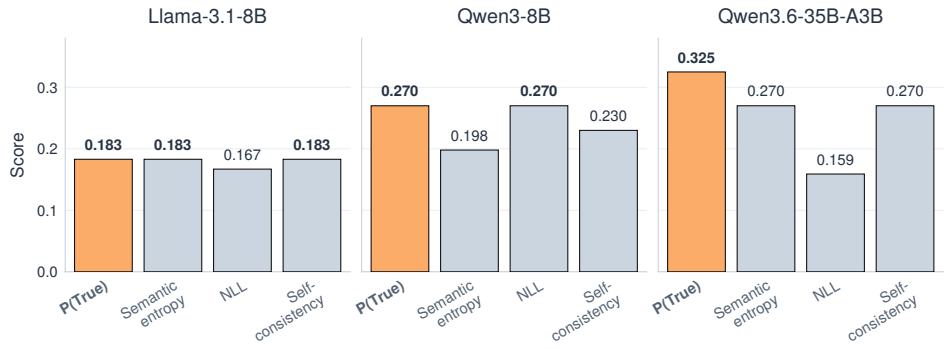

Figure 6: **Step-level attribution across uncertainty measures.** Comparison of $P(\text{True})$, semantic entropy, NLL, and self-consistency across three observer models.

four evaluation conditions, with particularly large gains on Algorithm-Generated trajectories without GT (10.9% for binary vs. 22.8% for percentile and 24.0% for Otsu). Otsu achieves the highest average in three of four conditions, while percentile is slightly better on Algorithm-Generated with GT. These results suggest that the gain is not specific to a particular thresholding rule. We use Otsu by default because it requires no external calibration data.

4.5 Inference Cost

Our method removes dependence on frontier APIs. Figure 5 plots inference cost against step-level accuracy. Our observers require 0.050 to 2.527 GPU-hours per 1K steps and no API calls, whereas closed judges cost approximately $0.26 to $10.34 per 1K steps. For example, Qwen3.5-9B requires 0.150 GPU-hours per 1K steps while outperforming every closed judge on Algorithm-Generated step-level accuracy without GT. Because the observer is queried once per step with a

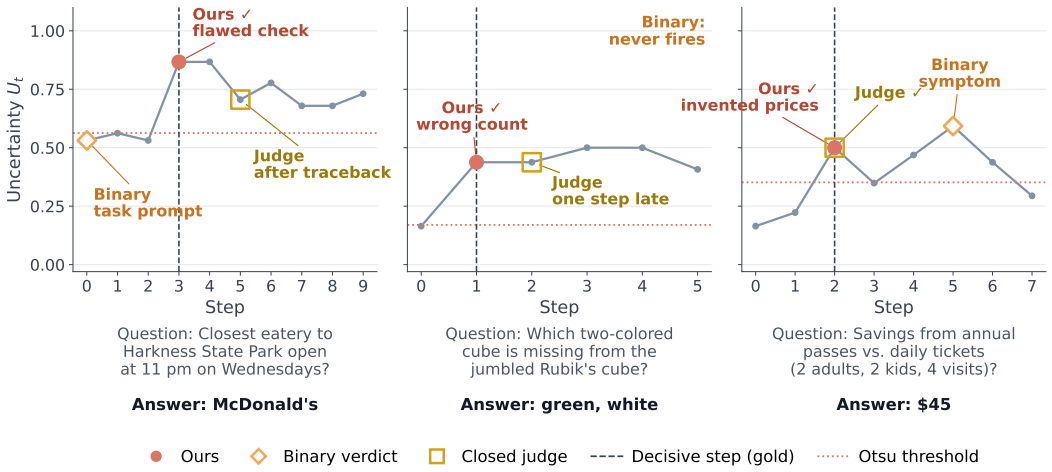

Figure 7: **Representative failure attribution examples.** Our uncertainty-based method identifies the annotated decisive step, while binary verdicts and closed LLM judges may miss or mislocalize the failure.

growing prefix, our method issues more calls with longer inputs than the judge baselines. It therefore does not reduce total inference volume, but replaces frontier-API dependence with single-token local inference that can run on a single GPU for models up to 35B parameters.

4.6 Qualitative Analysis

Figure 7 illustrates three representative cases where continuous uncertainty helps recover the decisive failure step. In the first example, our method localizes the flawed verification at Step 3, while the binary readout fires at an earlier non-failure step and the closed judge points to a later consequence. In the second example, uncertainty rises at the annotated error in Step 1, which our method correctly identifies; the binary readout never triggers, and the closed judge detects the failure one step later. In the third example, our method identifies the incorrect price information introduced at Step 2, whereas the binary readout selects a later downstream symptom at Step 5. These examples show that the continuous uncertainty trajectory can expose the onset of failure evidence more precisely than discrete verdicts.

5 Concluding Remarks

In this paper, we propose a training-free approach to failure attribution using the continuous uncertainty of a fixed observer. Rather than reducing the observer's output to a binary judgment, we retain its continuous confidence across trajectory steps and use it to localize decisive failures. Experiments on Who&When show that this continuous readout consistently improves over binary judgments from the same observer and achieves competitive performance against LLM judges without attribution-specific training or additional failure annotations. The gains persist across different uncertainty measures and unsupervised localization rules, suggesting that the benefit is not tied to a particular scoring or thresholding choice. Overall, our results indicate that continuous observer uncertainty provides useful step-level attribution information that is otherwise lost through binary decisions, offering a simple and low-overhead basis for failure attribution in multi-agent systems.

Limitations and Future Work. Our evaluation is limited to Who&When and a fixed set of observer models, leaving generalization to other multi-agent environments for future work. In addition, prefix-wise evaluation can become costly for long trajectories. Future work could evaluate broader agentic settings and develop more efficient ways to extract uncertainty from long trajectories.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing, including improving grammar, clarity, and readability. We did not use generative AI tools to generate research ideas, conduct analyses, interpret results, or produce research findings. Other disclosure tasks are not applicable to this work. All AI-assisted text was reviewed and revised by the authors to ensure accuracy and consistency with the intended meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems*, 2025.

Frank R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974. ISSN 01621459, 1537274X. URL http://www.jstor.org/stable/2285666.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Saurav Kadavath, Tom Conerly, Amanda Askell, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, October 2012. ISSN 1537-274X. doi: 10.1080/01621459.2012.737745. URL http://dx.doi.org/10.1080/01621459.2012.737745.

Fanqi Kong, Ruijie Zhang, Huaxiao Yin, Guibin Zhang, Xiaofei Zhang, Ziang Chen, Zhaowei Zhang, Xiaoyuan Zhang, Song-Chun Zhu, and Xue Feng. Aegis: Automated error generation and attribution for multi-agent systems. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zqcYoxXiN3.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Andrey Malinin and Mark J. F. Gales. Uncertainty estimation in autoregressive structured prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=jN5y-zb5Q7m.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 9004–9017. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.557. URL https://doi.org/10.18653/v1/2023.emnlp-main.557.

Geoffrey J. McLachlan and David Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics, 2000. ISBN 0-471-00626-2. URL https://www.bibsonomy.org/bibtex/1f4ae707df979236ba50173e48e851415.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, January 1979. doi: 10.1109/TSMC.1979.4310076. URL http://dx.doi.org/10.1109/TSMC.1979.4310076.

Huanxin Sheng, Xinyi Liu, Hangfeng He, Jieyu Zhao, and Jian Kang. Analyzing uncertainty of LLM-as-a-judge: Interval evaluations with conformal prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 11286–11328. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.569. URL https://aclanthology.org/2025.emnlp-main.569/.

Kai Sun, Wenqiang Li, Bo Dong, Yuxin Lin, Jingyao Zhang, and Bin Shi. Scope delineation before localization: A two-stage framework for enhancing failure attribution in multi-agent systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33108–33116, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/40594.

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 Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 5433–5442. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.330. URL https://doi.org/10.18653/v1/2023.emnlp-main.330.

John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley Publishing Company, Reading, MA, 1977.

Qiujie Xie, Qingqiu Li, Zhuohao Yu, Yuejie Zhang, Linyi Yang, and Yue Zhang. An empirical analysis of uncertainty in large language model evaluations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=J4xLuCt2kg.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *International Conference on Learning Representations*, 2024.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Zhiyuan Zeng, Xiaonan Li, Junqi Dai, Qinyuan Cheng, Xuanjing Huang, and Xipeng Qiu. Reasoning in flux: Enhancing large language models reasoning through uncertainty-aware adaptive guidance. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2401–2416, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.131. URL https://aclanthology.org/2024.acl-long.131/.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. Agentracer: Who is inducing failure in the LLM agentic systems? In Carl Vondrick, Bharath Hariharan, Colin Raffel, Lerrel Pinto, Diyi Yang, and Aleksandra Faust (eds.), *The Fourteenth International Conference on Learning Representations, ICLR 2026, Rio de Janeiro, Brazil, April 23-27, 2026*. proceedings.iclr.cc, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/134ed7a477770f227f12450ef0cbb8f4-Abstract-Conference.html.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. Which agent causes task failures and when? on automated failure attribution of LLM multi-agent systems. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/zhang25cq.html.

Qiwei Zhao, Dong Li, Yanchi Liu, Wei Cheng, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Huaxiu Yao, Chen Zhao, Haifeng Chen, and Xujiang Zhao. Uncertainty propagation on LLM agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 6064–6073, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

A Experimental Setup

A.1 Benchmark and Metrics

We use Who&When zhang2025whowhen. A log is gradable if it has at least two scored steps and its gold step is among the scored steps. This gives 126 Automated and 55 Hand-crafted logs. *Step accuracy* is exact match with the gold step, and *agent accuracy* is exact match with the gold agent. An abstention counts as a miss. We report 95% confidence intervals from a percentile bootstrap over logs with 2,000 resamples. Unless stated otherwise, no prompt contains the reference answer (the without-GT setting).

A.2 Models

Open observers. We use Qwen3.5-4B, Qwen3-8B, Llama-3.1-8B-Instruct, Qwen3.5-9B, Qwen3-32B, Qwen3.6-35B-A3B and Llama-3.3-70B-Instruct. Each is served with vLLM through its own chat template, with thinking disabled.

Closed judges. We use GPT-4o, GPT-4o-mini, GPT-4.1, Gemini 3.1 Flash Lite, Gemini 3.5 Flash Lite, DeepSeek V4 Flash and DeepSeek V4 Pro.

A.3 Uncertainty Readout

We score each step t with one request. The request contains the task and the transcript up to and including step t. It uses greedy decoding with `max_tokens` $= 1$ and `top_logprobs` $= 20$. Let p_{err} be the probability mass that these top-20 log-probabilities give to the answer meaning "this step is an error", summed over its spellings. Let p_{ok} be the same mass for the other answer. Then

$$U_t = \frac{p_{\text{err}}}{p_{\text{err}} + p_{\text{ok}}}, \qquad P(\text{True})_t = 1 - U_t.$$

The argmax token of the same distribution is the *binary verdict*. If a transcript exceeds 60,000 characters, we replace the content of the oldest steps with an omission marker. The judged step is never shortened.

A.4 Localization Rules

Rule	Definition
First crossing	$\hat{t} = \min\{t : U_t \geq c\}$; abstain if no step reaches c.
Map	$c = \text{quantile}(U, \hat{\pi})$, where $\hat{\pi} = h(\bar{U})$ (App. A.5).
Otsu	Per-trajectory cut that maximizes between-class variance.
GMM-2	Per-trajectory boundary of a two-component Gaussian mixture.
MAD / Tukey	$\text{median} + 3 \cdot \text{MAD}$ / $Q_3 + 1.5 \cdot \text{IQR}$.
Pooled pct	Percentile p of the pooled corpus U.
Changepoint	Single split that minimizes within-segment squared error.
Binary verdict	First step whose argmax token means "error".
argmax U	$\arg\max_t U_t$.

Table 5: Localization rules. The accused step's author is the fault agent.

A.5 Operating-Point Map

Reference corpus. We use single-agent ReAct trajectories from ALFWorld and HotpotQA that have step-level correctness labels. Each environment has four target agents (Llama-3.3-70B-Instruct, Mistral-7B-Instruct-v0.3, Phi-4-mini-instruct and gemma-3-4b-it) with 100 episodes each, for 8 cells in total. We score these trajectories with the same prompt and readout as Who&When.

Fit and placement. For each cell, $\bar{U}$ is the mean of U and needs no labels. $\pi^\star$ is the percentile at which the balanced-accuracy-optimal cut lies. The map h is the least-squares line through the 8 points $(\bar{U}, \pi^\star)$. We fit one map per observer and per estimator. On Who&When, we set $\hat{\pi} = h(\bar{U})$ and cut at that percentile of the target's own U, without using any Who&When label. When the target's $\bar{U}$ falls outside the range of $\bar{U}$ used in the fit, we mark the result as out of band.

B PROMPTS

B.1 STEP-BY-STEP PROMPT

This is the Step-by-Step prompt released with Who&When, used verbatim. Our method prefills the assistant turn with `1.` and reads the next token. `Yes` means the step is an error. The generative judge answers the same prompt without the prefill. It stops at the first `1. Yes`; if it never answers Yes, it abstains.

System.

```
You are a precise step-by-step conversation evaluator.
```

User.

```
You are an AI assistant tasked with evaluating the correctness of
each step in an ongoing multi-agent conversation aimed at solving
a real-world problem. The problem being addressed is: <problem>.
Here is the conversation history up to the current step:
Step 0 - <agent_0>: <content_0>
...
Step t - <agent_t>: <content_t>

The most recent step (<t>) was by '<agent_t>'.
Your task is to determine whether this most recent agent's
action (Step {idx}) contains an error that could hinder the
problem-solving process or lead to an incorrect solution. Please
respond with 'Yes' or 'No' and provide a clear explanation for
your judgment. Note: Please avoid being overly critical in your
evaluation. Focus on errors that clearly derail the
process.Respond ONLY in the format: 1. Yes/No.
2. Reason: [Your explanation here]
```

B.2 DECISIVE PROMPT

Here `True` means the step is an error.

System.

```
You are auditing a transcript from a multi-agent system that
failed to solve its task. You will see the task and the transcript
up to and including one step. Judge only that step, given
everything before it. Do not use hindsight you do not have.
```

User.

```
[task]
<question>

[transcript]
[step 0 | agent=<agent_0>]
<content_0>
...
[step t | agent=<agent_t>]
<content_t>

Question: Given the context above, is step <t> by '<agent_t>'
the point at which this task becomes unrecoverable, that is, the
first step whose error leads to the final failure?
Answer with exactly one word: True or False.
```