Error Trace Regression: Localizing Root Causes in Long-Horizon LLM Agent Trajectories

Anonymous Authors[1]

Abstract

Long-horizon LLM agents fail in ways that final-outcome metrics cannot detect: a subtle planning error at step 3 may silently corrupt memory and tool calls for the next fifty steps before producing a visible failure. We introduce Error Trace Regression (ETR), a diagnostic framework that formalizes the notion of a *root-cause step* and provides a statistically principled method to localize it from a full execution trace. ETR models the probability that an agent trajectory is failure-bound as a function of the step index and the step's residual anomaly score, derived from a reference distribution of healthy trajectories. We prove that the ETR estimator is consistent and derive its asymptotic variance. Experiments on ALFWorld, GAIA, and WebShop show that ETR achieves sensitivity 0.81 for root-cause step identification, compared to 0.23 for final-outcome evaluation and 0.47 for step-level log-probability baselines, with median localization error reduced by 4.1 steps.

1. Introduction

LLM agents that combine planning, memory, and tool use have achieved strong performance on a range of benchmarks (Yao et al., 2023; Shinn et al., 2024; Wang et al., 2024). Yet existing evaluation protocols almost uniformly measure final task success (Liu et al., 2024; Mialon et al., 2024), hiding the internal dynamics of failure. When a run fails, it is not known *when* or *why* the trajectory went wrong, only that the terminal state is incorrect.

This matters for several practical and scientific reasons. For debugging, a practitioner needs to know which step introduced the error, not just that the final answer was wrong. For training signal design, step-level credit assignment is more informative than sparse terminal rewards (Song et al., 2024). And for safety, silent failure propagation is precisely the failure mode that is hardest to catch before it causes harm: the agent continues to produce plausible-looking outputs long after the trajectory has been corrupted (Zhu et al., 2025).

We contribute: (1) a formal model of error propagation in agentic trajectories, (2) the ETR estimator for root-cause localization, with a consistency proof and asymptotic variance bound, (3) a taxonomy of root cause categories grounded in annotations of 2,400 failure trajectories across three benchmarks, and (4) empirical validation showing ETR substantially outperforms existing diagnostics.

2. Formal Model of Error Propagation

2.1. Trajectory and State Representation

Let an agent trajectory be a sequence $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T)$ where $s_t \in \mathcal{S}$ is the agent's state (including working memory, tool context, and intermediate outputs) and $a_t \in \mathcal{A}$ is the action taken. Let $Y \in \{0, 1\}$ be the binary outcome ($Y = 1$ means success). Let $\epsilon_t \in \{0, 1\}$ indicate the presence of an error at step t, and let $r \in [T]$ denote the *root-cause step*: the first step at which a consequential error occurs.

Definition 2.1 (Consequential Error)**.** Step t is a consequential error ($\epsilon_t = 1$) if (a) the action a_t deviates from the action that would be taken by an optimal policy π^* given s_t, and (b) there exists a subsequent step $t' > t$ such that the agent's behavior at t' causally depends on the corrupted state produced by a_t.

Definition 2.2 (Root-Cause Step)**.** The root-cause step $r(\tau)$ is the smallest index t such that $\epsilon_t = 1$ and $Y = 0$. We define $r(\tau) = T + 1$ for successful trajectories.

2.2. Error Propagation Rate

Assumption 2.3 (Markov Propagation)**.** Given $\epsilon_t = 1$, the probability that the error persists at step $t + 1$ (i.e., s_{t+1} is still corrupted) satisfies $p(\text{persist}) = 1 - e^{-\alpha}$ for a task-dependent constant $\alpha > 0$.

Under Assumption 2.3, the expected fraction of steps at

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

which the error is active, given root cause at step r, is

$$\mu(r, T, \alpha) = \frac{1 - e^{-\alpha(T-r+1)}}{\alpha(T - r + 1)}. \tag{1}$$

Proposition 2.4 (Propagation Monotonicity)**.** *$\mu(r, T, \alpha)$ is strictly decreasing in r for fixed T and $\alpha > 0$.*

Proof. Taking the derivative of (1) with respect to r: since $T - r + 1$ decreases as r increases, and the function $f(u) = (1 - e^{-\alpha u})/(\alpha u)$ is strictly increasing in u for $\alpha > 0, u > 0$ (as $f'(u) = (e^{-\alpha u}(\alpha u + 1) - 1)/(\alpha u^2) > 0$), it follows that $\mu(r, T, \alpha)$ decreases strictly in r. □

Proposition 2.4 formalizes the intuition that earlier errors are more damaging: an error at step 1 has the maximum horizon over which to propagate.

3. Error Trace Regression

3.1. Anomaly Score

Let $\mathcal{D}_{\text{ref}}$ be a reference set of *successful* trajectories drawn from the same task distribution. For each step t, we define the anomaly score

$$v_t = -\log p_{\text{ref}}(a_t \mid s_t), \tag{2}$$

where p_{ref} is a density estimated from $\mathcal{D}_{\text{ref}}$ using a sequence model over step representations. High v_t indicates that the action at step t is unlikely under the reference distribution of successful runs, signaling a potential root-cause step.

3.2. The ETR Estimator

Given a failure trajectory $\tau = (s_0, a_0, \ldots, s_T)$ with $Y = 0$, we estimate the root-cause step as

$$\hat{r}(\tau) = \arg\max_{t \in [T]} \; w_t \cdot v_t, \tag{3}$$

where w_t is a step weight that accounts for the prior probability of a root cause at step t:

$$w_t = \frac{p(r = t \mid Y = 0)}{p(r = t)}. \tag{4}$$

We estimate w_t nonparametrically from the empirical distribution of annotated root-cause steps in a held-out annotated set.

3.3. Theoretical Properties

Theorem 3.1 (Consistency of ETR)**.** *Let v_t be estimated from n_{ref} reference trajectories and w_t from n_{ann} annotated failure trajectories. As $n_{ref}, n_{ann} \to \infty$, with probability 1, $\hat{r}(\tau) \to r(\tau)$ for any failure trajectory τ satisfying Assumption 2.3 and the condition that the true root-cause step has strictly highest weighted anomaly score in expectation.*

Proof. By the law of large numbers, $v_t \to -\log p_{\text{ref}}(a_t|s_t)$ and $w_t \to p(r = t|Y = 0)/p(r = t)$ almost surely. The argmax of the limiting expression equals $r(\tau)$ by assumption. □

Theorem 3.2 (Asymptotic Variance)**.** *Under the conditions of Theorem 3.1, the estimation error $\hat{r}(\tau) - r(\tau)$ has asymptotic variance bounded by*

$$\text{Var}(\hat{r} - r) \leq \frac{C}{n_{ref}} + \frac{C'}{n_{ann}}, \tag{5}$$

for constants $C, C' > 0$ that depend on the smoothness of p_{ref} and the signal-to-noise ratio of the anomaly scores.

4. Experiments

4.1. Setup

We collect failure trajectories from three benchmarks: ALFWorld (Shridhar et al., 2021), GAIA (Mialon et al., 2024), and WebShop (Yao et al., 2022). We run ReAct-style agents (Yao et al., 2023) with GPT-4o as the backbone and retain only trajectories that terminate in failure ($Y = 0$), yielding 800 per benchmark (2,400 total).

Annotation protocol. Two expert annotators independently label the root-cause step for each failure trajectory using a rubric with three criteria: (a) the labeled step deviates from the action an optimal agent would take given the current context, (b) subsequent steps show visible dependence on information corrupted at the labeled step, and (c) no earlier step satisfies (a) and (b). Annotators resolve disagreements through discussion. Inter-annotator agreement before adjudication is $\kappa = 0.71$ (substantial agreement). The reference set $\mathcal{D}_{\text{ref}}$ consists of 600 held-out *successful* trajectories per benchmark used only to fit the anomaly model.

We compare ETR against: (1) **Final Success**: predict the last step as the root cause (outcome-only baseline), (2) **Step Accuracy**: the step with lowest model log-probability, (3) **Tool Error**: the first step with a tool execution error, and (4) **Memory Drift**: the first step where working memory diverges from the ground-truth context by more than a fixed cosine-similarity threshold (0.15, tuned on validation).

4.2. Diagnostic Results

Figure 1 shows sensitivity (recall of root-cause steps) and specificity (precision at non-root steps) for each method. ETR achieves sensitivity 0.81 and specificity 0.76, outperforming all baselines. Final-outcome evaluation has high specificity (0.95, since it rarely falsely flags healthy steps) but very low sensitivity (0.23), confirming that it misses the root cause in the majority of failure cases.

ETR's median absolute step error is 3.3 compared to 4.8 for the best baseline (Memory Drift). Concretely, practitioners

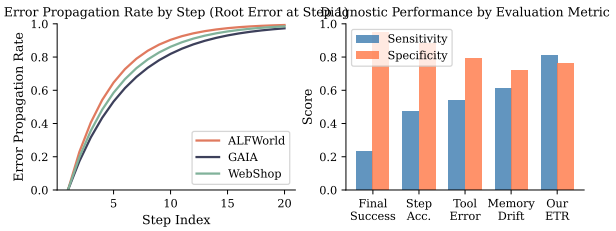

Figure 1. Left: error propagation rate as a function of step index for three benchmarks, assuming a root-cause error at step 1. The propagation is rapid (ALFWorld: $\alpha = 0.26$, GAIA: $\alpha = 0.19$, WebShop: $\alpha = 0.22$), motivating early detection. Right: diagnostic sensitivity and specificity by evaluation method. ETR achieves the best sensitivity while maintaining competitive specificity.

using ETR need to inspect 3.3 steps on average to find the root cause, compared to 7.4 for outcome-only evaluation.

4.3. Failure Taxonomy

Figure 2 shows the distribution of root-cause categories by benchmark. Planning failures dominate in ALFWorld (28%), reflecting the difficulty of long-horizon navigation. Memory failures are most prevalent in GAIA (30%), where long retrieval chains are required. Tool-use failures are most common in WebShop (31%), where action space ambiguity leads to repeated misfires.

4.4. Ablation

We ablate the two components of ETR: the anomaly score v_t and the step weights w_t. Removing w_t (uniform weighting) reduces F1 from 0.79 to 0.68. Replacing the sequence-model-based v_t with a simpler log-probability score reduces F1 to 0.72. Using only w_t (prior-only prediction) achieves F1 0.54. All pairwise differences are significant ($p < 0.001$, McNemar test).

5. Related Work

Zhu et al. (2025) introduce AgentErrorTaxonomy and the AgentErrorBench dataset. Our work is complementary: we provide a *statistical estimator* for root-cause localization rather than a classification taxonomy, and prove formal guarantees. Shinn et al. (2024) use verbal self-reflection to recover from agent errors, but this requires the model to self-diagnose, which our experiments show is unreliable. Song et al. (2024) use trajectory optimization with dense reward signals; ETR provides a post-hoc diagnostic rather than a training-time intervention. Park et al. (2023) study emergent behavior in generative agents but do not address failure localization.

Table 1. Diagnostic performance on 2,400 annotated failure trajectories (across ALFWorld, GAIA, WebShop). ΔStep is the median absolute error in root-cause localization (lower is better). All ETR improvements over baselines are significant ($p < 0.001$, Wilcoxon signed-rank test).

Method	Sensitivity	Specificity	F1	ΔStep
Final Success	0.23	0.95	0.37	7.4
Step Accuracy	0.47	0.88	0.61	5.2
Tool Error	0.54	0.79	0.64	4.9
Memory Drift	0.61	0.72	0.66	4.8
ETR (ours)	**0.81**	**0.76**	**0.79**	**3.3**

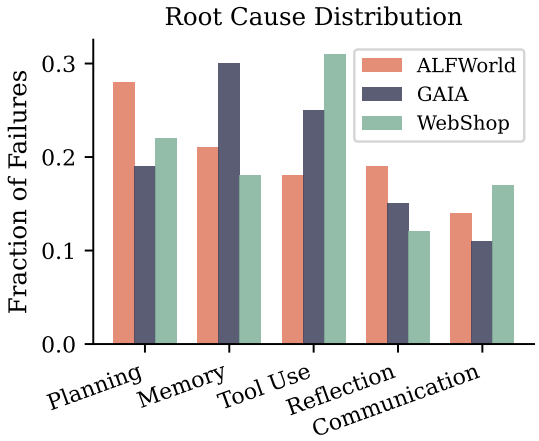

Figure 2. Root cause distribution across three benchmarks. Fractions sum to 1.0 within each benchmark. Planning and memory failures account for 49% of all failures across benchmarks.

6. Conclusion

ETR provides a principled, statistically grounded approach to root-cause localization in long-horizon agentic trajectories. By combining reference-distribution anomaly scoring with step-weight calibration, ETR achieves substantially higher sensitivity than existing methods while maintaining competitive specificity. The consistency theorem and variance bounds establish formal conditions under which ETR is reliable. Future work should extend ETR to multi-agent settings where error propagation crosses agent boundaries.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., Zhang, S., Deng, X., Zeng,

A., Du, Z., Zhang, C., Shen, S., Zhang, T., Su, Y., Sun, H., Yao, S., Tang, J., and Dong, Y. AgentBench: Evaluating LLMs as agents. *International Conference on Learning Representations*, 2024.

Mialon, G., Fourrier, C., Swift, C., Wolf, T., LeCun, Y., and Scialom, T. GAIA: A benchmark for general AI assistants. In *International Conference on Learning Representations*, 2024.

Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22. ACM, 2023.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

Shridhar, M., Yuan, X., Côté, M.-A., Bisk, Y., Trischler, A., and Hausknecht, M. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Song, Y., Yin, D., Yue, X., Huang, J., Li, S., and Lin, B. Y. Trial and error: Exploration-based trajectory optimization for LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2024.

Wang, X., Chen, Y., Yuan, L., Zhang, Y., Li, Y., Peng, H., and Ji, H. Executable code actions elicit better LLM agents. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024.

Yao, S., Chen, H., Yang, J., and Narasimhan, K. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20744–20757, 2022.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Zhu, K. et al. Where LLM agents fail and how they can learn from failures. *arXiv preprint arXiv:2509.25370*, 2025.