WHEN AND HOW DO SKILLS FAIL AGENTS? TRACING FAILURES IN SKILL-AUGMENTED AGENT TRAJECTORIES TO THEIR SOURCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Skills have become an important mechanism for improving the ability of large language model (LLM) agents to perform complex tasks. When a skill-augmented execution fails, however, the failure signal itself says little about the cause: a defective skill and an agent's misuse of a correct skill produce the same outcome but demand opposite repairs. This makes failure attribution over long-horizon trajectories of skill-augmented agents an important research problem. To study this problem, we develop a fine-grained attribution taxonomy and construct a human-annotated benchmark containing 194 trajectories from 30 task groups to evaluate the attribution capabilities of leading models. Our evaluation shows that many mainstream reasoning models have not yet reached a practically useful level of attribution performance. To further advance research on this capability, we propose SkillTracer, a 9B model specialized for skill attribution. SkillTracer improves attribution pass@1 by 11.3 points over its base model and outperforms frontier models evaluated without explicit reasoning at a fraction of their inference cost. Benchmark and code are available at `https://anonymous.4open.science/r/SkillAttribution-5906`.

1 INTRODUCTION

Large language model (LLM) agents increasingly rely on external skills to acquire specialized capabilities (Anthropic, 2025; Wang et al., 2023; Yao et al., 2023). A skill consists of organized folders of documents, scripts, and reference materials that an agent can discover and load dynamically (Anthropic, 2025). Compared with retraining a model or modifying a general-purpose agent harness, updating a skill is less costly and has a more controllable scope of impact. Skills are therefore becoming important engineering assets in agent systems, and recent benchmarks show that their quality and coverage directly affect agent performance across diverse tasks (Li et al., 2026; Zhong et al., 2026).

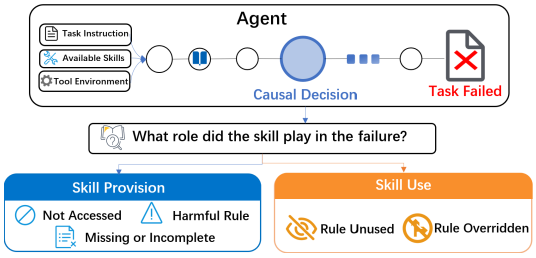


Figure 1: Causal attribution traces a failed execution to skill provision or skill use, supported by auditable evidence.

Yet a failed execution reveals the outcome, not its cause: the decisive error may stem from deficient skill guidance or from the agent's use of adequate guidance, which demand different fixes. Existing skill- and prompt-optimization methods use failure feedback to propose updates (Yang et al., 2026b; Shen et al., 2026; Zhou et al., 2023b; Pryzant et al., 2023; Yang et al., 2024), while agent failure-analysis methods primarily catalog failure patterns or localize responsible steps (Cemri et al., 2025; Deshpande et al., 2025; Zhang et al., 2025c;a). Neither directly distinguishes the causal role played by supplied skills. Consequently, similar failure symptoms may lead to the same update even when they arise from different problems in skill access, content, or use.

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.

To address this gap, we introduce SkillAttributionBench for evidence-grounded causal attribution in skill-augmented agents. Given the task instruction, supplied skills, execution trajectory, and verifier feedback, a model must identify the causal decision, attribute it to skill provision or skill use, and cite the governing text when available. The benchmark contains 194 human-adjudicated trajectories spanning 30 task groups. Evaluation on SkillAttributionBench shows that a range of existing models struggle to distinguish failures caused by skill content from those caused by agent behavior. Correct attribution requires identifying the decision that changed the execution outcome within a long trajectory and tracing its relationship to the skill. Existing models, however, often rely on local textual cues or terminal actions (Liu et al., 2024; Deshpande et al., 2025; Abdollahi et al., 2025) and fail to reconstruct this causal chain reliably.

To investigate whether targeted training can improve this capability, we construct a training set that is strictly disjoint from the benchmark and train a lightweight attribution model based on Qwen3.5-9B (Qwen Team, 2026). The model takes the task instruction, skills, execution trajectory, and verifier feedback as input, and outputs a failure category together with supporting textual evidence. Training combines supervised fine-tuning and GRPO (Shao et al., 2024). The reward evaluates category correctness, source and quotation alignment, and output validity, thereby providing partial credit to partially correct predictions. Relative to the base model, the trained model improves attribution performance by 11.3 points.

We further test whether causal attribution can guide downstream repair using human-adjudicated attributions as the diagnosis. For benchmark tasks with observed execution failures, a repair model receives the human-adjudicated attribution and makes at most one atomic edit to the relevant skill, without observing post-edit feedback or iteratively revising the result. Across fresh task executions, this single-step repair increases success from 43.3% to 63.3%. The result shows that the benchmark diagnoses can narrow the repair target and translate observed failures into measurable improvements.

Our main contributions are as follows:

- We formulate evidence-grounded causal failure attribution for skill-augmented agents, distinguishing failures in skill provision from failures in skill use.
- We introduce SkillAttributionBench, comprising 194 human-adjudicated trajectories from 30 task groups with auditable evidence, and reveal that current models remain unreliable at this task.
- We propose SKILLTRACER, a 9B attribution model that improves pass@1 by 11.3 points over its base model, and evaluate how causal attribution can guide downstream repair.

2 RELATED WORK

2.1 SKILL-AUGMENTED AGENTS AND SKILL OPTIMIZATION

LLM agents increasingly acquire reusable capabilities from external skills containing instructions, scripts, and references. Agent Skills organize such artifacts for on-demand loading (Anthropic, 2025), while Voyager builds an expanding library of executable skills from experience (Wang et al., 2023). Recent benchmarks measure how skill quality, coverage, and retrieval affect downstream performance (Li et al., 2026; Zhong et al., 2026; Tan et al., 2026). Related systems retain reusable experience as reflections, distilled lessons, or workflow memories (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2025; Yu et al., 2025; Yang et al., 2026a). Together, they establish reusable guidance as an editable layer of agent systems. SkillOpt and SkillOpt-Lite revise skills from recurring execution feedback (Yang et al., 2026b; Shen et al., 2026). Related methods optimize prompts using natural-language feedback, generated candidates, or evolutionary search (Zhou et al., 2023b; Yang et al., 2024; Guo et al., 2024; Fernando et al., 2024; Khattab et al., 2024; Yuksekgonul et al., 2024); ProTeGi, for example, converts minibatch errors into textual gradients (Pryzant et al., 2023). Optimization has also expanded to complete agent programs and workflows (Cheng et al., 2024; Zhang et al., 2025b; Hu et al., 2025). These approaches exploit failed executions but need not identify whether a symptom originates in skill access, content, or application. Our work isolates this diagnostic problem before deciding what to rewrite.

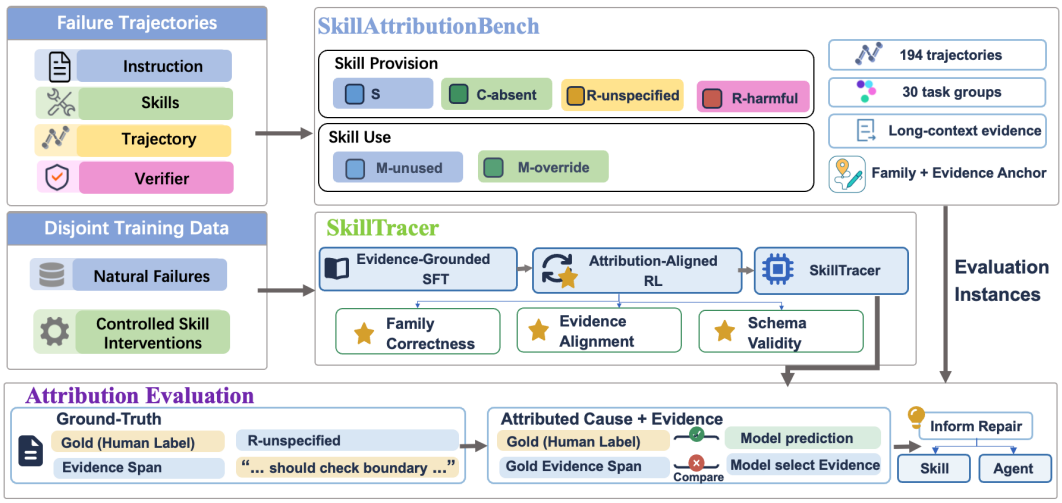


Figure 2: Overview of SKILLATTRIBUTIONBENCH and SKILLTRACER. The benchmark organizes human-adjudicated failure trajectories into six skill-provision and skill-use families with supporting evidence. A disjoint training set supports evidence-grounded supervised fine-tuning and attribution-aligned reinforcement learning of SKILLTRACER. At evaluation time, model predictions are compared with the adjudicated cause and evidence.

2.2 FAILURE ATTRIBUTION IN AGENTIC SYSTEMS

Prior work analyzes agent failures at several levels. MAST develops a taxonomy of recurring multi-agent failures (Cemri et al., 2025); TRAIL localizes errors inside agent traces (Deshpande et al., 2025); and Abdollahi et al. identify where reasoning first diverges from a correct solution (Abdollahi et al., 2025). Process-supervision work similarly evaluates intermediate reasoning errors (Uesato et al., 2022; Lightman et al., 2024; Zheng et al., 2025; Huang et al., 2024). These studies characterize what went wrong or where it became visible. Failure attribution further asks who or what is responsible. Who&When identifies the responsible agent and decisive step (Zhang et al., 2025c), while AgenTracer trains a lightweight model for agent–step localization using counterfactual replay and fault injection (Zhang et al., 2025a). GuideBench studies adherence to domain-specific guidelines (Diao et al., 2025), and broader evaluations examine instruction following, interactive tool use, and trajectory judging (Zhou et al., 2023a; Yao et al., 2025; Qi et al., 2025; Zhuge et al., 2024). Our formulation instead attributes failure to the causal role of the supplied skill. It distinguishes failures in skill provision from failures in skill use and grounds each applicable prediction in the governing rule, making the result auditable at the level where skills are inspected and revised.

3 SKILLATTRIBUTIONBENCH

Agent benchmarks are typically designed to measure task completion or instruction following (Li et al., 2026; Zhong et al., 2026). These evaluations establish what an agent can do, but do not aim to diagnose the role of a supplied skill after a failure. SKILLATTRIBUTIONBENCH addresses this complementary question by asking what in the skill-augmented execution should be repaired. To make the answer actionable, it distinguishes failures in skill selection and content from failures in the agent's use of available guidance. To make the answer auditable, each finding cites the governing rule verbatim whenever one exists. The task therefore requires tracing the verifier symptom back to the failure-producing decision and determining whether the relevant guidance was unavailable, defective, or incorrectly applied.

3.1 TASK DEFINITION

Judging an attribution requires seeing everything the agent saw and everything the environment reported; otherwise whether guidance was available becomes impossible to determine after the fact.

Table 1: The SkillAttributionBench taxonomy. The families are defined by the causal role of a skill in the failed decision, rather than by surface similarity between a rule and the final error.

Family	Failure locus	Operational criterion
`S`	Skill selection	Guidance needed for the failed decision exists in an available skill, but that skill (or the necessary part of it) does not enter the agent's context.
`C-absent`	Skill coverage	The consulted skills contain no reusable rule for the decision that causes the failure. Solving it requires adding a new, generalizable branch rather than merely clarifying existing text.
`R-unspecified`	Rule specification	The relevant rule is present and encountered, but leaves an operationally necessary mechanism, boundary, parameter, or interpretation unresolved.
`R-harmful`	Rule correctness	A faithful, literal application of the skill guidance itself leads to the failed behavior.
`M-unused`	Model application	Adequate guidance enters context and the decision point is reached, but the agent silently fails to operationalize it.
`M-override`	Model decision	The agent considers the applicable guidance and explicitly replaces it with another choice that causes the failure.

We therefore represent one failed execution as

$$x = (I, S, \tau, V), \tag{1}$$

where I is the task instruction, S is the set of skills supplied to the agent, τ is the complete reasoning-and-action trajectory, and V is the verifier feedback. The verifier establishes that the execution failed; the benchmark does not ask the model to predict this outcome again. Instead, the model must identify the causal decision that produced the failure and determine its relation to the supplied guidance, returning

$$z = \{(f_j, A_j)\}_{j=1}^{J}, \tag{2}$$

where f_j is a fault family and A_j is a set of supporting textual anchors. Each anchor consists of a canonical source and a verbatim quotation, so that the attribution can be checked against the input and mapped back to the system component that should be repaired. Most trajectories contain one finding, while the set-valued output allows multiple findings only when they describe distinct causal chains.

The central difficulty is separating a root cause from its downstream symptoms. For example, a missing output file may result from an ignored delivery rule, but it may also be the consequence of an earlier unresolved modeling decision that consumed the execution budget. We therefore reconstruct each failure backward:

$$\text{verifier symptom} \leftarrow \text{failed artifact or behavior} \leftarrow \text{producing action} \leftarrow \text{agent decision}. \tag{3}$$

Attribution stops at the earliest evidenced decision in the reconstructed causal chain whose correction would prevent the failure and whose cause lies in skill provision or skill use.

3.2 Attribution Taxonomy

The taxonomy is derived from the space of repairs, not from surface error types. A failed skill-augmented execution admits three repair actions: bring the needed skill into context, modify the skill's content, or correct the agent's use of an adequate skill. The six families refine these actions into checkable conditions; they are summarized in Table 1.

Each family maps to a distinct intervention. Fixing an `S` failure requires changing retrieval or loading, whereas modifying the skill body cannot help if the skill is never read. The content families `C-absent`, `R-unspecified`, and `R-harmful` require adding, completing, or correcting skill knowledge, respectively. By contrast, `M-unused` and `M-override` attribute the failure to the agent's use of correct in-context guidance. The boundary between the two model-side families is semantic: a different final action alone does not prove an override; the trajectory must show that the agent considered and displaced the rule.

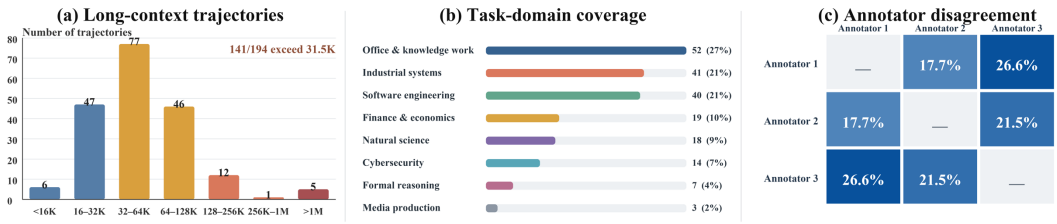


Figure 3: Overview of SKILLATTRIBUTIONBENCH. (a) Distribution of the 194 original trajectories across context-length intervals; 141 exceed the 31.5K tokens. (b) Distribution across primary task domains, counted by trajectory. (c) Pairwise family-set disagreement rates among three annotators over their shared annotation set. Each off-diagonal cell reports the proportion for one annotator pair.

Some executions are filtered out before inclusion: runs that ended early for environmental or infrastructural reasons, or whose task or verifier proved invalid, never become benchmark instances. This keeps the benchmark focused on failures in which a skill genuinely played a causal role.

3.3 TRAJECTORY COLLECTION AND INSTANCE CONSTRUCTION

The benchmark is built from failed executions of tool-using agents on heterogeneous, verifier-backed tasks adapted from SkillsBench (Li et al., 2026) and SkillLearnBench (Zhong et al., 2026). To cover failures of diverse agent systems rather than a single stack, executions run under Harbor with several agent models (including GLM-5.2, Kimi-K2.6, and DeepSeek-V4-Flash) and harnesses (opencode, claude-code, and Codex CLI). The resulting 194 trajectories span 30 task groups covering office and knowledge work, industrial systems, software engineering, finance and economics, natural science, cybersecurity, formal reasoning, and media production. Each instance retains the task instruction, the supplied skills, the agent's reasoning and tool interactions, and the verifier outcome, because skill entry, the causal decision, and the observed failure often occur at different points in the trajectory.

3.4 HUMAN CAUSAL ADJUDICATION

Long agent trajectories often support several plausible explanations, so adjudication requires a more detailed and precise causal analysis than matching surface symptoms. Annotators trace each observed failure back to the decision that produced it and assess the causal role of the relevant skill at that decision, distinguishing defective guidance from incorrect use of adequate guidance; the latter judgment asks whether a faithful application of the rule would itself have failed.

All trajectories are annotated independently by three annotators, each spending about 30 hours on average, and disagreements are resolved through a final adjudication pass rather than majority voting. When a governing text exists, the final finding must quote it verbatim, so that the evidence behind each label can be inspected and verified against the input; none is required for `S`, where the skill never entered context, or `C-absent`, where the claim is precisely that no reusable rule exists.

3.5 BENCHMARK PROPERTIES

The benchmark covers all six fault families, and `C-absent` and `R-harmful` carry a deliberately high burden of proof: the former requires evidence that a reusable rule is genuinely missing, the latter that faithful use of the rule itself leads to failure. Evidence is also distributed across long executions: the original trajectories average 135K model tokens, with a median of 49K and a maximum above 4.37 million. Evidence that a skill entered context may appear early, the decisive choice much later, and the observable symptom only in the final verifier output.

The task distribution spans software and cyber systems, physical control, scientific analysis, finance, formal reasoning, document work, and media production. The pairwise disagreement rates in Figure 3(c) show that independently produced family sets can diverge substantially, which is why each claim is checked against its own causal evidence before entering the benchmark.

Together, the adjudication process and the long-context structure make SKILLATTRIBUTIONBENCH a test of causal diagnosis rather than error-message classification.

4 SKILLTRACER: EVIDENCE-GROUNDED ATTRIBUTION

Failure attribution may be invoked repeatedly as skills and agents are improved, making exclusive reliance on frontier reasoning models costly. Yet our benchmark shows that attribution remains difficult even for such models (Section 5), so it is unclear whether a compact model can acquire this capability. We therefore train SKILLTRACER, a 9B model initialized from Qwen3.5-9B (Qwen Team, 2026). Given the task instruction, supplied skills, execution trajectory, and verifier feedback, it predicts both the fault family and the textual evidence supporting that decision. Requiring both prevents a plausible label from being treated as a verifiable attribution without evidence.

Figure 2 summarizes the training pipeline. We first build causal supervision from natural failures and controlled skill interventions, then use the same family–evidence annotations to construct SFT targets and RL rewards.

4.1 DUAL-SOURCE CAUSAL SUPERVISION

Natural failures provide realistic agent behavior, but assigning the correct causal label is itself the hard attribution problem: a long failure trajectory often admits several plausible explanations, so a single-pass label may simply be the most plausible-sounding one. We therefore use GPT-5.6-sol in a two-stage annotation pipeline:

$$\begin{aligned} \hat{z}_i &= \mathrm{Annotate}(x_i), \\ z_i &= \mathrm{CausalReview}(x_i, \hat{z}_i), \\ \mathcal{D}_{\mathrm{nat}} &= \{(x_i, z_i) \mid z_i \neq \perp\}. \end{aligned} \tag{4}$$

Here $\hat{z}_i$ is the candidate family and evidence anchors, while the second stage confirms or revises them through root-cause-oriented review. It returns $\perp$ for samples that remain uncertain, arise from infrastructure faults, or come from invalid tasks, so only causally supported annotations enter $\mathcal{D}_{\mathrm{nat}}$. Natural failures alone, however, contain relatively few clear examples in which skill content is missing, incomplete, or harmful.

Skill intervention provides a complementary, controlled source. Given a successful host trajectory τ^+ produced with skill S, we select a task-relevant decision d and apply a family-specific mutation m_f to obtain a paired execution:

$$\begin{aligned} \tau^+ &= \mathrm{Exec}(I, S), & \Omega(\tau^+) &= 1, \\ \tau_f^- &= \mathrm{Exec}(I, m_f(S; d)), & \Omega(\tau_f^-) &= 0. \end{aligned} \tag{5}$$

Here m_f removes, weakens, or replaces the rule governing decision d for the target family f. We then re-execute the task with the mutated skill. Pairing the successful host τ^+ with the failed intervention τ_f^- isolates the changed skill content and supplies clearer causal supervision for otherwise rare skill-side failures. These reviewed pairs form $\mathcal{D}_{\mathrm{inj}}$. Natural and intervention data use the same family–anchor schema, and their task groups are disjoint from the benchmark.

After review, the resulting pool contains 1,118 training instances from 62 task groups and 116 validation instances from 7 disjoint groups.

4.2 EVIDENCE-GROUNDED SFT

SFT trains the model to produce a causal explanation before the structured family–anchor annotation. We use Kimi-K3 to construct or revise this reasoning under the fixed, reviewed ground truth. The explanation must connect the verifier symptom to the producing decision and the role of the skill, while Kimi-K3 cannot alter the reviewed family or evidence. We retain only targets whose reasoning agrees with the annotation and whose quotations are recoverable from their declared sources, yielding 660 SFT examples.

4.3 ATTRIBUTION-ALIGNED RL

We further optimize the SFT model with GRPO (Shao et al., 2024). The reward mirrors the two requirements of the task: selecting the correct causal family and grounding that decision in the

correct text. For examples with evidence anchors, it is

$$R = 0.55F_{0.5}^{\text{family}} + 0.40F_{1}^{\text{anchor}} + 0.05\,\mathbb{I}_{\text{schema}}. \tag{6}$$

For a predicted anchor $\hat{a} = (\hat{s}, \hat{q})$ and reference anchor $a = (s, q)$, similarity is explicitly source-gated:

$$\text{sim}(\hat{a}, a) = \mathbb{I}[\hat{s} = s]\, F_{1}^{\text{token}}(\hat{q}, q). \tag{7}$$

Anchor matches are formed within each family, so a quotation receives credit only when both its source and textual content align with the reference.

For `S` and `C-absent`, which have no positive rule to quote, the anchor weight is reassigned to family correctness and correctly leaving the anchor set empty. Overall, the reward favors the causal decision and its supporting evidence rather than surface similarity to a reference answer.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Data sources. The full benchmark contains 194 human-adjudicated failure trajectories from 30 task groups adapted from SkillsBench (Li et al., 2026) and SkillLearnBench (Zhong et al., 2026). They are produced under Harbor using several agent models (including GLM-5.2, Kimi-K2.6, and DeepSeek-V4-Flash) and harnesses (opencode, claude-code, and Codex CLI), and retain the instruction, supplied skills, execution trace, and verifier outcome (Section 3). The training data use entirely disjoint task groups and contain 1,118 RL instances. Applying the stricter reasoning and evidence checks required for SFT yields 660 examples (Section 4.1).

Table 2: Single-pass attribution results (%) on the 32K benchmark, grouped by inference mode. Anchor-F1 is computed on instances with quotable gold evidence; parsing failures count as incorrect. Cost is the estimated total for the full evaluation.

	Family Accuracy			
Model	pass@1	pass@4	Anchor-F1	Cost ($)
Thinking mode				
Kimi-K2.6	48.44	66.67	20.70	46.30
DeepSeek-V4-Flash	48.70	69.79	21.35	5.56
GLM-5.2	50.78	70.31	23.19	53.20
GLM-5.3-Flash	55.21	66.67	26.41	65.78
GPT-5.6-sol (low)	51.56	67.19	23.09	77.68
GPT-5.6-sol (high)	56.90	73.96	28.54	130.68
Kimi-K3	**62.50**	**75.00**	**31.25**	114.36
No-thinking mode				
DeepSeek-V4-Flash	32.42	45.31	**9.79**	2.83
GLM-5.2	26.95	42.71	8.44	26.23
Kimi-K2.6	**37.24**	**54.69**	9.75	18.82
Small models				
Qwen3-8B	17.71	33.85	5.31	0.51
Qwen3.5-9B	29.95	52.08	12.36	0.52
+ EG-SFT	34.38	**63.54**	11.23	0.49
SKILLTRACER	**41.28**	60.42	**14.09**	0.50

Evaluation set. We evaluate on a standardized 32K version of SKILLATTRIBUTIONBENCH. It preserves the task instruction, skill documents, skill-access records, verifier evidence, and adjudicated evidence spans while compacting repetitive tool observations. This compression leaves attribution accuracy essentially unchanged while roughly halving input-token cost (Appendix B), and brings 192 of the 194 trajectories within the 32K budget; for a controlled comparison, the main table uses these 192 trajectories, and all reported models have completed four samples.

Models and protocol. To separate the effects of explicit reasoning and targeted training, we evaluate three groups of systems. The thinking-mode comparison covers Kimi-K2.6, DeepSeek-V4-Flash, GLM-5.2, GLM-5.3-Flash, Kimi-K3, and GPT-5.6-sol at two reasoning-effort levels. We additionally evaluate DeepSeek-V4-Flash and matched no-thinking configurations of GLM-5.2 and Kimi-K2.6 in no-thinking mode. To measure the effect of attribution training, we compare Qwen3.5-9B with Evidence-Grounded SFT and the final SKILLTRACER, using Qwen3-8B as an additional small-model reference. Every configuration receives the same single-pass prompt and standardized 32K instance. We sample four responses per trajectory at temperature 0.7; parsing failures are retained and counted as incorrect predictions. The complete evaluation prompt is given in Appendix D.

Table 3: Per-family attribution F1 (%) on the benchmark, reported for thinking-mode, no-thinking, and small-model settings. **Bold** marks the best result within each setting.

Model	S	C-absent	M-unused	M-override	R-unspecified	R-harmful
Thinking mode						
Kimi-K2.6	82.71	15.50	49.89	31.85	23.71	42.53
DeepSeek-V4-Flash	79.80	18.71	47.03	35.67	14.77	64.19
GLM-5.2	71.22	36.97	48.67	40.49	17.34	61.62
GLM-5.3-Flash	87.78	30.56	54.67	38.10	**48.34**	62.44
GPT-5.6-sol (low)	93.73	34.34	46.06	45.13	21.46	39.57
GPT-5.6-sol (high)	**95.60**	34.04	50.12	**51.91**	36.65	54.63
Kimi-K3	95.47	**44.59**	**61.54**	34.94	45.00	**74.31**
No-thinking mode						
DeepSeek-V4-Flash	60.25	11.11	39.85	30.87	3.86	0.00
GLM-5.2	56.10	15.93	29.53	27.40	**5.26**	4.03
Kimi-K2.6	**69.05**	**18.39**	**40.63**	**36.30**	2.21	**10.19**
Small models						
Qwen3-8B	25.64	16.67	28.66	0.00	**14.43**	17.16
Qwen3.5-9B	55.81	5.66	**37.89**	22.93	5.62	9.21
+ EG-SFT	80.60	16.97	29.95	16.15	9.49	37.29
SKILLTRACER	**92.10**	**20.38**	29.31	**27.88**	9.28	**44.44**

Metrics. Family Accuracy requires the predicted family set to exactly match the human annotation; we report it as pass@1 and pass@4 for different sampling budgets. A correct label alone, however, does not imply that the supporting evidence is right. To measure this, we report Anchor-F1, the token-level F1 between the predicted quotations and the adjudicated evidence anchors, matched greedily and gated by source identity. Instances whose gold finding carries no quotable rule (`S` and `C-absent`) are excluded from this measure.

5.2 MAIN FINDINGS

Frontier reasoning models remain unreliable for instance-level skill attribution. Table 2 shows that failure attribution remains difficult even for strong reasoning models: the best pass@1 is 62.50%, leaving more than one third of trajectories incorrectly attributed. Current reasoning models therefore do not yet make instance-level skill attribution reliable enough to replace human adjudication. Evidence grounding is weak in absolute terms. Anchor-F1 requires the correct source and verbatim overlap, and the best model reaches only 31.25, so quotations are frequently missing, misattributed, or paraphrased. Since the task requires label and evidence together, such predictions can hardly be audited. In no-thinking mode, Anchor-F1 falls below 10 for all three models, leaving the evidence effectively unverifiable.

Models are better attributors than any single answer suggests. Every model improves by 11 points or more from pass@1 to pass@4 (Table 2). For a substantial fraction of trajectories, a correct attribution appears only when several samples are drawn. By itself, this gap leaves open how the model weighs candidate causes. It establishes that single-pass accuracy understates what these models can produce, and that the answer obtained from one sample can differ from the answer obtained from another. The spread is even wider for the small models examined in Section 5.3, where the choice between SFT and RL directly trades pass@4 against pass@1. Since downstream repair consumes a single answer, this sample-to-sample variability makes any individual response undependable without an aggregation or confidence mechanism.

Models tend to reduce the failure to generic non-use of an available rule. Table 3 and the pooled confusion matrix (Figure 6, Appendix C) reveal a consistent directional bias. Most models predict `M-unused` more often than it appears in the benchmark, and every model underpredicts `M-override`. Aggregated across models, 51% of single-label `M-override` cases are reduced to `M-unused`; the same fallback is also the most frequent error for `C-absent` and `R-unspecified`. This fallback collapses failures caused by absent or incomplete guidance, as well as explicit rule replacement, into a generic agent-use error.

5.3 ANALYSIS

Targeted training improves the reliability of a compact attribution model. Targeted supervision is the key design choice for a compact attributor, since practical accuracy must be obtained without frontier-scale inference cost. We compare Qwen3.5-9B with Evidence-Grounded SFT and SKILLTRACER under the identical protocol (*Small models* block of Table 2). Targeted training raises pass@1 by 11.3 points over the base model and 6.9 over the EG-SFT, with the largest gains on the skill-side families targeted by the intervention data (`S` and `R-harmful`; Table 3), while `M-unused` F1 decreases, consistent with the fallback bias being suppressed rather than amplified. The SFT checkpoint attains the higher pass@4 while SKILLTRACER attains the higher pass@1, consistent with reinforcement sharpening favors a single dependable answer over a broader spread of plausible ones (Yue et al., 2025), the property that matters when a repair loop consumes one attribution. SKILLTRACER thus provides the practical operating point for routine attribution, surpassing every frontier model evaluated without explicit reasoning at significantly lower inference cost, with the remaining gap to thinking-mode models being an expected trade-off.

Trajectory length does not explain attribution difficulty. A natural concern is that the benchmark is hard merely because its trajectories are long. All models receive inputs compacted into the same 32K budget, and splitting instances at 32K source-trajectory tokens changes accuracy by at most 3.3 points for the three representative reasoning models (Figure 4). At the same time, accuracy on the shorter group remains far from saturation (below 65% for all three). The benchmark's difficulty therefore survives length normalization, which supports interpreting it as a test of causal diagnosis.

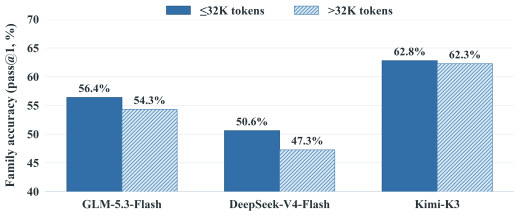


Figure 4: Family accuracy by source-trajectory length on the evaluation set. Stronger models exhibit a smaller accuracy drop.

A single attribution-guided edit can improve downstream task success. We finally test whether a causal attribution can be turned into an actionable skill update. We select the ten benchmark tasks on which DeepSeek-V4-Flash produced a verifier-confirmed failure, ensuring that every repair targets an observed benchmark failure. Kimi-K3 receives the failed trajectory and its human-adjudicated attribution, then makes at most one atomic edit to one contiguous skill span. This is a single-step optimization with no post-edit feedback or revision. We evaluate both skill versions with three fresh runs of the same agent and harness, yielding 30 executions per condition. As shown in Figure 5, the original skills succeed in 13 of 30 executions (43.3%), whereas the skills produced by the single attribution-guided edit succeed in 19 of 30 (63.3%), an improvement of 20 percentage points. The benchmark labels can therefore serve as an interface from observed failures to targeted skill repair.

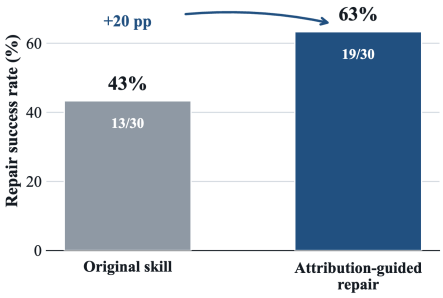


Figure 5: Original versus one-step attribution-guided skill repair, aggregated over ten tasks with three runs each.

6 CONCLUSION

A failed skill-augmented execution does not reveal what should be fixed. The problem may lie in how the skill was provided, in what it says, or in how the agent used otherwise adequate guidance. We make this diagnostic step measurable through SKILLATTRIBUTIONBENCH, which evaluates the attributed cause and the textual evidence supporting it. Our results show that current models still struggle to distinguish closely related failure modes, especially silent rule non-use from explicit rule replacement, and often fail to ground their decisions in relevant evidence. At the same time, SKILLTRACER shows that this capability is trainable: targeted supervision substantially improves the attribution reliability of a compact model. Together, these findings establish evidence-grounded failure attribution as a distinct, measurable, and trainable capability for skill-augmented agents.

AI USE STATEMENT

Generative AI tools were used in both the research workflow and manuscript preparation. GPT-5.6-sol produced preliminary family–evidence annotations and performed causal review for training set; annotations that remained uncertain were excluded. Then Kimi-K3 was used to construct and revise evidence-grounded reasoning targets under fixed, reviewed ground truth. Generative AI tools were additionally used to improve the clarity and language of the manuscript. They were not treated as autonomous annotators or authors. Final benchmark labels were human adjudicated; evidence quotations were checked against their declared sources; and reported results, citations, and AI-assisted text were reviewed by the authors. The authors take responsibility for all claims, artifacts, and content in this work.

ETHICS STATEMENT

This work studies agent-generated execution traces rather than interactions with research participants. Human involvement is limited to annotation and adjudication, and no annotator-sensitive attributes are analyzed. Some source tasks imitate real operational settings and contain synthetic identifiers that resemble personal information; these are task content rather than participant data. The public release is limited to redistributable materials and is screened for accidental credentials and internal service information. The benchmark and SKILLTRACER are intended to support diagnosis and improvement of skill-augmented agents, but their predictions can be incorrect and should not be used as the sole basis for modifying high-stakes production systems. Attribution quality may also vary across task domains and agent implementations; these limitations should be considered when applying the model or extending the benchmark.

REPRODUCIBILITY STATEMENT

Section 3 describes the benchmark taxonomy, collection, and human adjudication procedure; Section 4 specifies the training-data construction, SFT targets, and attribution-aligned reward; and Section 5 reports the evaluation protocol and model configurations. Appendix D provides the complete benchmark evaluation prompt. The anonymous supplementary material includes the released benchmark and training splits, preprocessing and evaluation scripts, reward implementation, and commands needed to reproduce the reported experiments. Task-group-disjoint splits and fixed instance identifiers are provided to prevent train–test leakage and to support direct comparison with future work.

REFERENCES

Mohammad Abdollahi, Khandaker Rifah Tasnia, Soumit Kanti Saha, Jinqiu Yang, Song Wang, and Hadi Hemmati. Demystifying errors in LLM reasoning traces: An empirical study of code execution simulation, 2025. URL https://arxiv.org/abs/2512.00215.

Anthropic. Equipping agents for the real world with Agent Skills. https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills, 2025. Anthropic Engineering Blog, October 2025.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Kannan Ramchandran, Dan Klein, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? *arXiv preprint arXiv:2503.13657*, 2025. URL https://arxiv.org/abs/2503.13657.

Ching-An Cheng, Allen Nie, and Adith Swaminathan. Trace is the next AutoDiff: Generative optimization with rich feedback, execution traces, and LLMs. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2024)*, 2024. URL https://arxiv.org/abs/2406.16218.

Darshan Deshpande et al. TRAIL: Trace reasoning and agentic issue localization. *arXiv preprint arXiv:2505.08638*, 2025. URL https://arxiv.org/abs/2505.08638. Patronus AI.

Lingxiao Diao, Xinyue Xu, Wanxuan Sun, Cheng Yang, and Zhuosheng Zhang. GuideBench: Benchmarking domain-oriented guideline following for LLM agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11361–11399. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.557.

Chrisantha Fernando, Dylan Sunil Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. In *Forty-first International Conference on Machine Learning (ICML)*, 2024. URL https://openreview.net/forum?id=9ZxnPZGmPU.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=ZG3RaNIsO8.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2408.08435.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.01798.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.03714.

Xiangyi Li, Wenbo Chen, Yimin Liu, Shenghan Zheng, Xiaokun Chen, Yifeng He, Yubo Li, Bingran You, Haotian Shen, Jiankai Sun, Shuyi Wang, Qunhong Zeng, Di Wang, Xuandong Zhao, Yuanli Wang, Roey Ben Chaim, Zonglin Di, et al. SkillsBench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026. URL https://arxiv.org/abs/2602.12670.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2305.20050.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. URL https://arxiv.org/abs/2307.03172.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7957–7968. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.494.

Yunjia Qi, Hao Peng, Xiaozhi Wang, Amy Xin, Youfeng Liu, Bin Xu, Lei Hou, and Juanzi Li. AGENTIF: Benchmarking instruction following of large language models in agentic scenarios. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025) Datasets and Benchmarks Track*, 2025. URL https://arxiv.org/abs/2505.16944.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, 2026. February 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Yifei Shen, Bo Li, and Xinjie Zhang. SkillOpt-Lite: Better and faster agent self-evolution via one line of vibe. *arXiv preprint arXiv:2607.03451*, 2026. URL https://arxiv.org/abs/2607.03451.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023. URL https://arxiv.org/abs/2303.11366.

Boyin Tan, Xiaowei Huang, and Youcheng Sun. Skill coverage: A test adequacy metric for agent skills. *arXiv preprint arXiv:2606.20659*, 2026. URL https://arxiv.org/abs/2606.20659.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022. URL https://arxiv.org/abs/2211.14275.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2305.16291.

Zhiruo Wang, Jiayi Pan, Xingyao Wang, Xingchen Ma, Simon Li, Heng Ji, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2409.07429.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=Bb4VGOWELI.

Ke Yang, Zixi Chen, Xuan He, Jize Jiang, Michel Galley, Chenglong Wang, Jianfeng Gao, Jiawei Han, and ChengXiang Zhai. PlugMem: A task-agnostic plugin memory module for LLM agents, 2026a. URL https://arxiv.org/abs/2603.03296.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. SkillOpt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026b. URL https://arxiv.org/abs/2605.23904.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.03629.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2406.12045.

Chenglin Yu, Yang Yu, Songmiao Wang, Yucheng Wang, Yifan Yang, Jinjia Li, Ming Li, and Hongxia Yang. InfiAgent: Self-evolving pyramid agent framework for infinite scenarios, 2025. URL https://arxiv.org/abs/2509.22502.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 57654–57689. Curran Associates, Inc., 2025. doi: 10.52202/085713-1933. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/537d5aa768c2d534016a4d06f87bc8fb-Paper-Conference.pdf.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. TextGrad: Automatic "differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024. URL https://arxiv.org/abs/2406.07496.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. AgenTracer: Who is inducing failure in the LLM agentic systems? *arXiv preprint arXiv:2509.03312*, 2025a. URL https://arxiv.org/abs/2509.03312.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025b. URL https://arxiv.org/abs/2410.10762.

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 *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76583–76599, 2025c. URL https://arxiv.org/abs/2505.00212.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.50. URL https://arxiv.org/abs/2412.06559.

Shanshan Zhong, Yi Lu, Jingjie Ning, Yibing Wan, Lihan Feng, Yuyi Ao, Leonardo F. R. Ribeiro, Markus Dreyer, Sean Ammirati, and Chenyan Xiong. SkillLearnBench: Benchmarking continual learning methods for agent skill generation on real-world tasks. *arXiv preprint arXiv:2604.20087*, 2026. URL https://arxiv.org/abs/2604.20087.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023a. URL https://arxiv.org/abs/2311.07911.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023b. URL https://openreview.net/forum?id=92gvk82DE-.

Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-judge: Evaluate agents with agents. *arXiv preprint arXiv:2410.10934*, 2024. URL https://arxiv.org/abs/2410.10934.

A Case Study

To illustrate the difficulty of skill-failure attribution, we present a representative example from the `dialogue-parser` task. The task requires an agent to convert a dialogue script into a structured JSON graph while satisfying reachability and edge-target constraints. In this example, the agent successfully retrieved the relevant `dialogue-graph` skill and inspected its implementation. The skill demonstrates that transitions may target a terminal marker `End`, while the task instruction states that edge targets may be absent for the last node. However, neither the instruction nor the skill explicitly specifies whether `End`, as a terminal target, may also be serialized as an ordinary graph node.

During execution, the agent recognized the ambiguity and reasoned that "End need not exist as a node, but edge targets to End are fine." It then concluded that "including End node is fine either way" and added `End` to the output node list. The agent's own validation succeeded, and the supplied graph implementation also accepted the generated structure. However, the external verifier adopted

Table 4: Case study of attribution ambiguity in a failed agent trajectory. The same evidence leads to different causal interpretations.

Trajectory Evidence	Attribution Analysis
The agent loads the relevant skill and inspects its implementation. The guidance allows `End` as an edge target but does not explicitly define whether `End` can also appear as a serialized node.	**Human: R-unspecified.** The available guidance constrains the decision but leaves the terminal representation boundary unresolved. The failure is attributed to incomplete specification rather than failure to apply a complete rule.
The agent decides to explicitly add `End` into the node list after considering the last-node exception.	**GLM-5.2: M-unused.** The model treats the reachability requirement as a complete governing rule and attributes the failure to the agent not making the emitted `End` node reachable. This assumes an operational constraint that is not explicitly provided by the available guidance.
The agent's self-check passes, but the external verifier skips `End` during traversal and reports it as unreachable.	**DeepSeek-V4-Flash: M-unused.** The model interprets the last-node exception as sufficient evidence that `End` should not be emitted as a node. It therefore attributes the failure to ignoring an existing rule rather than an unresolved semantic boundary.

a different traversal convention, where edges pointing to `End` were not explored during reachability checking. Consequently, the explicitly emitted `End` node was judged unreachable and caused the task failure.

This example highlights a fundamental challenge in skill attribution: the presence of related guidance does not necessarily imply that the corresponding decision is fully specified. Both GLM-5.2 and DeepSeek-V4-Flash classify the failure as M-UNUSED, but their reasoning reveals different attribution shortcuts. GLM-5.2 assumes that the general reachability objective fully determines the required behavior, whereas DeepSeek-V4-Flash strengthens an existence exception into a prohibition on node emission. In contrast, the human-adjudicated label is R-UNSPECIFIED, emphasizing that attribution requires determining whether the available skill actually resolves the causal decision boundary, rather than merely matching a failed action to a related rule. The case also marks the filtering boundary described in Section 3.2: the verifier's traversal convention is consistent with the task specification but undocumented in the skill, so the instance is retained as R-unspecified rather than excluded.

B COMPRESSION RELIABILITY

The 32K evaluation version compacts repetitive tool observations while preserving task instructions, skill documents, skill-access records, verifier evidence, and adjudicated evidence spans. To verify that this compression does not distort attribution difficulty, we reran three models on the full, uncompressed trajectories of the same instances under the identical protocol (four samples per instance). Table 5 shows that pass@1 is essentially unchanged, while the uncompressed inputs cost 2.1 times as many input tokens per request. Instances whose uncompressed form exceeds a model's context window are excluded from both conditions.

Table 5: Attribution accuracy (%) and mean input size on the 32K evaluation version versus full, uncompressed trajectories, on the instances completed in both conditions and within context limits.

	pass@1		input tokens (K)	
Model	32K	full	32K	full
GLM-5.3-Flash	55.1	58.8	20.2	42.2
DeepSeek-V4-Flash	48.5	47.2	19.9	40.3
Kimi-K2.6	52.2	50.6	20.2	42.2
Mean	51.9	52.2	20.1	41.6

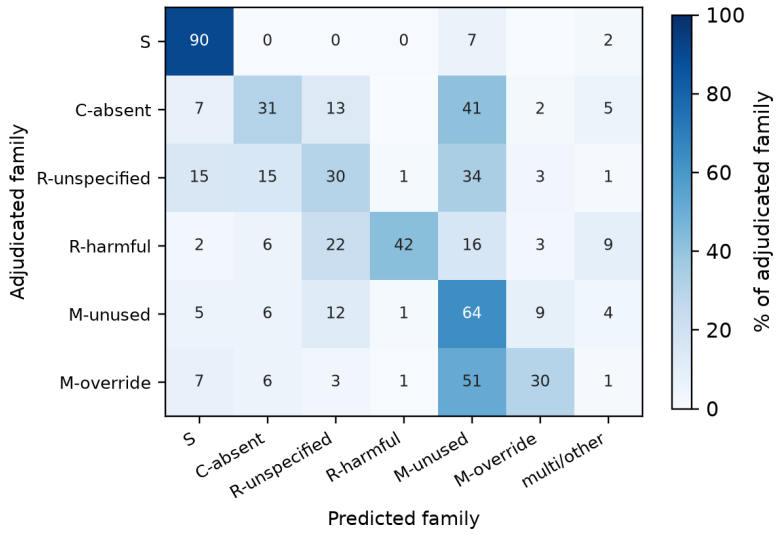


Figure 6: Confusion matrix pooled over the seven thinking-mode models on single-label instances, row-normalized. The last column aggregates multi-family predictions, empty outputs, and parsing failures.

C Confusion Matrix

Figure 6 reports the pooled confusion matrix over the thinking-mode models, restricted to single-label instances and normalized per adjudicated family. The `M-unused` column absorbs a large share of misattributions from every other family, including 51% of adjudicated `M-override` cases, which supports the fallback-bias analysis in Section 5.

D Benchmark Evaluation Prompt

Tables 6–9 show the single-pass prompt used for all benchmark evaluations. Placeholders in braces are filled per instance; the taxonomy procedure and output requirements are shown verbatim.

Table 6: Single-pass attribution prompt used in benchmark evaluation (Part 1: input blocks, causal-decision localization, and the selection gate).

Prompts for SkillAttributionBench evaluation (Part 1)

```
System:  You are an expert diagnostician of LLM-agent failures.

User:
## Task instruction
=== source:  instruction ===
{task instruction}

## Skill documents
=== source:  {skill-name}/{filename} ===
{skill file text}

## Fault families and causal rules
Classify by the earliest skill-related decision that caused the
verifier
failure, using the following procedure in order.  Do not jump
directly from a
failed artifact to a family.

### Step 1:  localize the causal decision

Reconstruct the dependency chain from input discovery through
computation,
validation, and delivery.  Trace each verifier symptom backward to
the earliest
decision that would probably have prevented it.  A missing file or
field is a
downstream symptom when the trajectory actually stalled, timed out,
looped, or
failed earlier.

### Step 2:  apply the selection gate

Did the skill containing the rule needed at that decision point
actually enter
the trajectory through a skill invocation or file read?

- No -> S. Stop classification of that same fault.  Seeing related
terminology
or reading a sibling skill is not evidence that the needed text
entered.
- Yes -> continue.  Merely embedding a skill document in this
benchmark prompt
does not count; use only what the original agent read in the
trajectory.
```

Table 7: Single-pass attribution prompt used in benchmark evaluation (Part 2: distinguishing absent, unused, and attempted text; attempted-rule faults).

Prompts for SkillAttributionBench evaluation (Part 2)

```
### Step 3:  distinguish absent text, unused text, and attempted
text

- No rule in the selected skill governs the reached decision ->
C-absent.
- A correct, sufficiently specific rule was in the agent's context,
the agent
reached its decision point, but did not apply it -> M. Continue to
Step 5.
- The agent attempted to apply a rule and its behavior is traceable
to that
rule -> R. Continue to Step 4.

Do not infer that a rule was unused merely because the final
artifact is absent.
Prerequisite values being available is also not proof that the
agent reached
and passed the governed decision point.

### Step 4:  classify attempted-rule faults

- R-harmful:  a perfect, literal execution of the rule's actual
directive would
still deterministically reproduce the failure in this context.
- R-unspecified:  faithful execution still requires the agent to
invent a
decision-constitutive threshold, procedure, interpretation, or
applicability
boundary, and that improvisation failed.

R-harmful requires evidence of BOTH exact rule adoption and harmful
rule
semantics.  It is forbidden when the agent merely made its own
coding mistake,
ignored the rule, incompletely implemented it, or selected a
plausible detail
that the rule left open.  The presence of a relevant rule plus a
failed result is
not sufficient evidence for R-harmful.
```

Table 8: Single-pass attribution prompt used in benchmark evaluation (Part 3: unapplied correct-rule faults, boundary examples, and evidence requirements).

Prompts for SkillAttributionBench evaluation (Part 3)

```
### Step 5:  classify unapplied correct-rule faults

- M-override:  the reasoning explicitly references the correct rule
and then
rejects, negates, or substitutes another rule for it.
- M-unused:  the correct rule was verifiably in context and the
decision point
was reached, but the rule shows no application and no negation
narrative.

### Minimal boundary examples

- Needed PID skill never read; agent guesses a controller -> S.
- Skill has no guidance for choosing the aggregation boundary ->
C-absent.
- Skill specifies UTC days; agent reaches aggregation but silently
uses local
days -> M-unused.
- Agent cites the UTC rule, calls it unsuitable, and chooses local
days ->
M-override.
- Skill explicitly says to threshold the daily mean; agent does
exactly that
and the required event-level result fails -> R-harmful.
- Skill says to aggregate daily but does not define timezone; agent
guesses the
wrong boundary -> R-unspecified.

For every proposed family, identify concrete trajectory evidence
for skill
entry, decision-point reachability, attempted versus unapplied
behavior, and
the connection to the verifier symptom.  If a required fact is
unsupported, do
not use that family.  Text conflicts are not a category; classify
only the
concrete branch followed by the agent.
```

Table 9: Single-pass attribution prompt used in benchmark evaluation (Part 4: trajectory input and output requirements).

Prompts for SkillAttributionBench evaluation (Part 4)

```
## Trajectory digest
{compressed trajectory digest}

## Output requirements
Analyze the trajectory and output ONLY a JSON object:
{
"findings": [
{
"family": "S" | "M-unused" | "M-override" | "R-harmful" |
"R-unspecified" | "C-absent",
"anchors": [
{"source": "<skill-name>/<filename> or instruction",
"quote": "text copied from that source"}
]
}
]
}

For source, copy the canonical identifier shown in an input header
exactly.
Use "instruction" for the task instruction. Skill sources use only
the skill
directory name and concrete filename; do not invent or reconstruct
a relative
path. Copy quote text from that source rather than paraphrasing or
inventing it.

Output exactly one finding by default. Output multiple findings
only
when the trajectory contains multiple independent
skill-attributable faults,
each arising at a distinct causal decision point with its own
evidence. Do
not output multiple findings merely because several labels seem
plausible or
related. Each finding owns only the anchors supporting that
finding. If the
same evidence would support two labels, choose the single most
causally direct
family.

Before emitting JSON, silently verify:
1. The finding names the earliest causal decision, not a
downstream symptom.
2. S was checked before every content-level family.
3. M has proof that a correct rule was in context and its decision
was reached.
4. R has proof that the agent attempted the cited rule.
5. R-harmful passes the perfect-literal-follower test; otherwise
choose
R-unspecified, M, C-absent, or S as supported by the evidence.
6. Every quote is verbatim and every source exactly matches an
input header.
```