ATTRIBUTION IS A PROPAGATION PROBLEM: END-TO-END CAUSAL EVIDENCE ATTRIBUTION IN MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models are increasingly used in hierarchical multi-agent systems, where external evidence can be transformed and passed through several agents before affecting a final decision. This raises a basic attribution question: *does evidence that appears important when it first enters the system also matter to the final outcome?* We study this question using controlled evidence-removal interventions and use TradingAgents, a 12-agent trading system, as our testbed. For each removable evidence item, we compare its sensitivity at the agent that first processes it, at a shared downstream checkpoint, and at the final decision. Across three LLM backends, local sensitivity is only weakly aligned with final causal influence, while downstream sensitivity is substantially more aligned. We introduce *Propagation-Aware Replay (PAR)*, which uses intermediate propagation states to decide which evidence items receive full end-to-end causal verification. Our study covers 25 ticker–date cases, with the main evaluation on 15 cases and 2,346 candidate–model observations. At the evaluated operating point, PAR retains 79.53% of evidence with measured final causal influence while reducing projected structural verification cost by 20.70%. These results support treating attribution in hierarchical multi-agent systems as an end-to-end propagation problem rather than a purely local one.

1 INTRODUCTION

Large language models are increasingly used in multi-agent systems, where different agents collect information, analyze it, pass outputs to other agents, and together produce a final decision (Guo et al., 2024; Wu et al., 2024; Hong et al., 2024). Explaining one agent is not necessarily enough to explain that decision. Existing attribution methods typically study which parts of a model's input matter or which agents contribute to the final outcome (Cohen-Wang et al., 2024; Liu et al., 2025; Lu et al., 2026). What is less clear is what happens to one piece of external information after it enters a multi-agent system. A news article or social-media post may look unimportant to the first agent that processes it, but still affect the final decision after passing through the rest of the system.

This matters in systems that make consequential decisions, such as multi-agent trading systems (Xiao et al., 2024). Here, the question is not only which information an individual agent reacts to, but which items actually affect the final BUY, HOLD, or SELL decision. We therefore ask: *does an item that looks important when it first enters the system also matter to the final decision?*

We observed a clear mismatch in TradingAgents (Xiao et al., 2024). In one evaluation case, a Reddit post ranked 29th out of 30 social-media items by local sensitivity at the Sentiment Analyst. At the downstream Research Manager, the same post ranked first among 55 candidates. Removing only this post changed the final decision from BUY to SELL in all three runs. This example shows why local importance alone may not be enough to explain the final system decision.

We study this with controlled removal interventions. For each news item or social-media post, we remove only that item, rerun the affected downstream agents, and compare the final decision with the corresponding clean run using the same random seed. Similar removal-based interventions have been used for context and agent-level attribution (Cohen-Wang et al., 2024; Liu et al., 2025; Lu et al., 2026). Repeating this for every item gives us an exhaustive reference of which items affect

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.

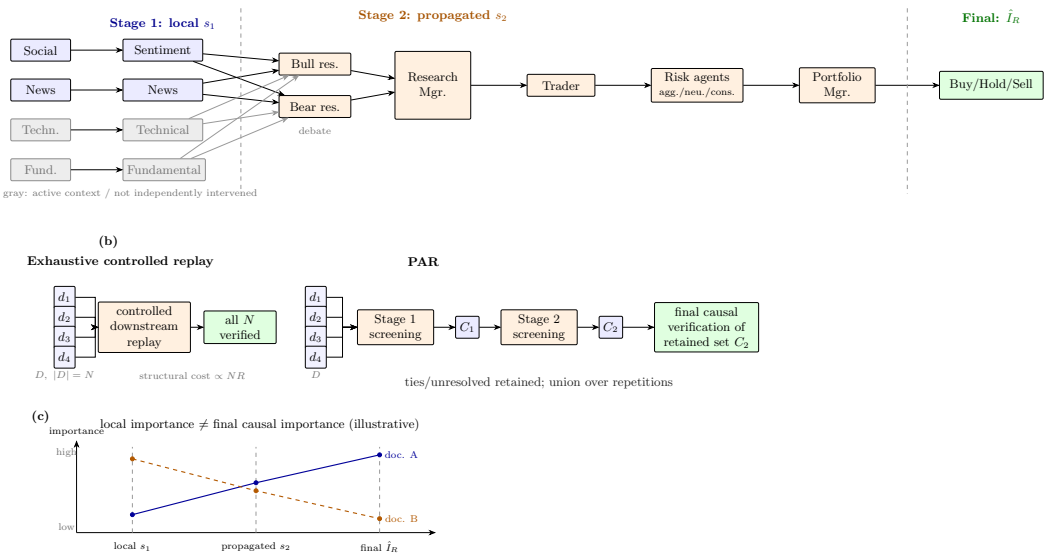


Figure 1: **End-to-end causal attribution and Propagation-Aware Replay (PAR).** (a) We track the same external information at the first agent, a shared downstream checkpoint, and the final decision. Gray branches remain active as context but are not independently intervened upon. (b) Exhaustive replay propagates every removal to the final decision, whereas PAR screens candidates at intermediate checkpoints and verifies only the retained set. (c) Local sensitivity can differ from downstream sensitivity and final causal influence: an item can rank low locally, much higher downstream, and still affect the final decision.

the final decision, but requires each removal to be propagated through several downstream agents and repeated across multiple runs.

We also look at what happens before the final decision. For each item, we consider three points in the system: the agent that first processes it, a shared downstream checkpoint, and the final decision. We refer to these as *local sensitivity*, *propagated sensitivity*, and *final causal influence*. Because the checkpoints expose different types of outputs, we compare their rankings rather than their raw scores.

We introduce **Propagation-Aware Replay (PAR)** to reduce the cost of exhaustive replay. PAR does not decide which items matter from the local agent alone. It first screens candidates using local sensitivity and then replays the retained items to a shared downstream checkpoint. Candidates are ranked separately in each run, ties and unresolved cases are kept, and the retained sets are combined across runs.

The remaining items are then ranked by their downstream sensitivity. An item that ranks low at the first agent may rank much higher downstream. PAR uses this second checkpoint to select candidates for full end-to-end causal verification, reducing the amount of final verification without relying only on local sensitivity.

Contributions. Our work makes three contributions:

- We study attribution across the full path from external information entering a multi-agent system to the final decision. We show that an item that looks weak at the first agent can still have a strong final causal effect.
- We introduce *Propagation-Aware Replay (PAR)*, which uses local and downstream sensitivity to decide which items receive full end-to-end causal verification.
- We evaluate PAR in TradingAgents across 15 cases and three LLM backends. We measure how local and downstream sensitivity relate to final causal influence, and how much verification cost can be reduced while retaining influential items.

2 Related Work

Our work is closest to research on context attribution, contribution estimation in multi-agent systems, counterfactual replay, causal analysis of agent failures, and information provenance.

Context and evidence attribution. Context-attribution methods ask which parts of an LLM's input contribute to its output. ContextCite (Cohen-Wang et al., 2024) attributes a generation to pieces of supporting context, while AttriBoT (Liu et al., 2025) reduces the cost of leave-one-out context attribution using screening, activation reuse, and proxy models. These methods mainly explain one model invocation. We instead follow the same external item through several agents and measure whether it affects the final system decision.

Credit assignment in multi-agent systems. Other work asks which agents contribute to the system-level outcome. Agents that Matter (Lu et al., 2026) studies agent removal using a cooperative-game formulation, while HiveMind (Xia et al., 2026) uses workflow structure to make agent-level contribution estimation more efficient. Our unit of analysis is different: we remove an individual news or social-media item rather than an agent.

Counterfactual replay and agentic attribution. Counterfactual interventions have also been used to analyze agent executions. The Why Behind the Action (Qian et al., 2026) attributes actions to parts of a hierarchical interaction. DoVer (Ma et al., 2026) uses targeted interventions to test debugging hypotheses, while Causal Agent Replay (Shah, 2026) re-executes downstream computation after intervening on trajectory steps. Related zero-replay methods (Kang et al., 2026) try to identify useful interventions without executing every counterfactual. We also use intervention and replay, but our intervention is on external information entering the system rather than on an internal message, trajectory step, or agent.

Causal and failure attribution in multi-agent systems. A related line of work studies where failures arise inside multi-agent systems. CHIEF (Wang et al., 2026a) models execution traces using hierarchical causal structure and uses counterfactual screening to identify root causes. DCFA (Wang et al., 2026b) combines causal-dependency analysis with local counterfactual refinement, while EDGE (Hou et al., 2026) models dependencies between errors and validates them through counterfactual interventions. Other examples include CDC-MAS (Ma et al., 2025), AgentRx (Barke et al., 2026), and failure-attribution benchmarks such as Who&When (Zhang et al., 2025).

Agent-Specific Effects (Triantafyllou et al., 2023) studies how the effect of one agent's action is carried through other agents. Counterfactual Effect Decomposition (Triantafyllou et al., 2024) further separates effects that pass through other agents from those that pass through the environment. These methods study internal agents, actions, reasoning steps, or failures. We instead track an external information item from the point where it enters the system to the final decision.

Information provenance. Another related question is how external information enters and is represented inside agentic systems. RAGentA (Besrour et al., 2025) links generated claims to retrieved documents, while the Misattribution Gap (Ahad et al., 2026) studies cases where information introduced through memory is later attributed incorrectly. This work is related to our setting, but it does not measure the effect of removing the same item at multiple points along an agent hierarchy.

Positioning. Prior work explains different parts of a multi-agent system: the input context, individual agents, internal execution steps, failures, or the provenance of external information. We focus on one external item and follow it through the system. We measure its sensitivity where it first enters, its sensitivity at a shared downstream checkpoint, and its causal influence on the final decision. This lets us test whether local importance agrees with what happens later in the system.

PAR uses this structure for screening. Instead of deciding which items deserve final verification from the first agent alone, it uses both local and downstream sensitivity to decide which candidates continue to full end-to-end causal verification.

3 Problem Formulation

Evidence removal. We use *evidence* to refer to the external information items consumed by the system, such as news articles or social-media posts. Let $\mathcal{M}$ be the set of evidence modalities and D_m the set of evidence items from modality $m \in \mathcal{M}$. The complete set of removable evidence is

$$D = \bigcup_{m \in \mathcal{M}} D_m = \{d_1, \ldots, d_N\},$$

where $N = |D|$ is the total number of removable items. In our evaluation, $\mathcal{M}$ contains social-media and historical-news evidence.

For each run $r \in \{1, \ldots, R\}$, let y_r be the clean final decision. To test the effect of item d_i, we remove only that item and replay the affected downstream part of the system. We denote the resulting decision by y_r^{-i}. The clean and removal runs use the same random seed, and all other evidence remains unchanged.

Final causal influence. We first measure how removing an item changes the final decision. In TradingAgents, the three possible actions are mapped to

$$q(\text{SELL}) = -1, \qquad q(\text{HOLD}) = 0, \qquad q(\text{BUY}) = +1.$$

For item d_i and run r, the change caused by its removal is

$$\Delta_{i,r} = q(y_r^{-i}) - q(y_r).$$

A positive value means that removal shifts the decision toward BUY, a negative value shifts it toward SELL, and zero means that the final action does not change.

We aggregate the effect across the R runs as

$$\widehat{I}_R(d_i) = \left| \frac{1}{R} \sum_{r=1}^{R} \Delta_{i,r} \right|.$$

We average the signed changes before taking the absolute value. Changes in the same direction reinforce each other, while changes in opposite directions partially cancel. We use $\widehat{I}_R(d_i)$ as our estimate of the final causal influence of item d_i under this intervention setup.

We call an item *final-important* when its removal has non-zero final causal influence:

$$F = \{d_i \in D : \widehat{I}_R(d_i) > 0\}. \tag{1}$$

An item is included only when all required clean and removal runs are valid. Missing runs are never treated as zero effect or as HOLD.

Attribution along the hierarchy. The final decision tells us whether removing an item matters at the end of the system. We also want to know what happens before that point. We observe the same item at three stages: (i) its *local sensitivity* $s_1(d_i)$ at the agent that first processes it; (ii) its *propagated sensitivity* $s_2(d_i)$ at a shared downstream checkpoint; and (iii) its *final causal influence* $\widehat{I}_R(d_i)$ on the final decision.

The three quantities are based on different native outputs and are not directly comparable in magnitude. We therefore compare their rankings rather than their raw values. In particular, we ask whether an item that ranks highly at the first agent also ranks highly downstream and has a strong final causal influence. An item can rank low locally but much higher downstream, or it can look important locally and have little effect on the final decision.

Our formulation is not specific to trading. It applies when external evidence passes through identifiable downstream stages, individual evidence items can be removed, and the final system output can be compared before and after the removal. We use TradingAgents as our testbed because it provides this full evidence-to-decision path and allows us to evaluate the removals exhaustively.

4 PROPAGATION-AWARE REPLAY

Exhaustive attribution carries every evidence removal through the affected part of the system until the final decision. This gives us the exhaustive causal reference, but it is expensive because every candidate must be replayed through the downstream workflow. **Propagation-Aware Replay (PAR)** reduces this cost by screening candidates at intermediate stages and reserving full end-to-end verification for the items that remain.

Controlled replay. For an item d_i, PAR reruns only the components that can be affected by its removal. Components outside this path reuse their clean state, while the affected components are replayed downstream using the same random seed as the corresponding clean run. This avoids rerunning branches that cannot depend on d_i.

Checkpoint sensitivity. For run r, let $H^0_{k,r}$ and $H^{-i}_{k,r}$ denote the clean and removal outputs at checkpoint k. We compute

$$s_{k,r}(d_i) = \rho_k\left(H^0_{k,r}, H^{-i}_{k,r}\right),$$

where ρ_k is the sensitivity measure used at that checkpoint. In our evaluation, $s_{1,r}$ measures local sensitivity at the affected analyst and $s_{2,r}$ measures propagated sensitivity at the Research Manager. These scores are used only for screening; they are not interpreted as final causal influence.

Progressive screening. PAR screens candidates separately in each run. At the first checkpoint, candidates are ranked within their evidence modality because the local outputs of different analyst branches are not directly comparable. PAR keeps the top b_1 fraction from each modality. Exact ties at the cutoff are kept, and candidates whose score cannot be resolved also continue.

The surviving candidates are then replayed to the shared downstream checkpoint. Because they are now evaluated at the same representation, they can be ranked together. PAR keeps the top b_2 fraction, again retaining cutoff ties and unresolved candidates. The retained sets from the R runs are then combined by union, so an item can still continue if it shows a strong response in only one run.

Algorithm 1 summarizes the full procedure.

Algorithm 1 Propagation-Aware Replay (PAR)

Require: Evidence sets $\{D_m\}_{m\in\mathcal{M}}$, runs R, screening budgets b_1, b_2
for $r = 1, \ldots, R$ **do**
  $C_{1,r} \leftarrow \emptyset$
  for each modality $m \in \mathcal{M}$ **do**
    compute local sensitivity $s_{1,r}(d)$ for $d \in D_m$
    separate rankable candidates $E_{1,m,r}$ from unresolved candidates $U_{1,m,r}$
    $C_{1,r} \leftarrow C_{1,r} \cup \mathrm{TopTieExpanded}(E_{1,m,r}, s_{1,r}, \lceil b_1|D_m|\rceil) \cup U_{1,m,r}$
  end for
  replay candidates in $C_{1,r}$ to the shared downstream checkpoint
  compute propagated sensitivity $s_{2,r}(d)$
  separate rankable candidates $E_{2,r}$ from unresolved candidates $U_{2,r}$
  $C_{2,r} \leftarrow \mathrm{TopTieExpanded}(E_{2,r}, s_{2,r}, \lceil b_2|C_{1,r}|\rceil) \cup U_{2,r}$
end for
$C_1 \leftarrow \bigcup_{r=1}^{R} C_{1,r}$, $C_2 \leftarrow \bigcup_{r=1}^{R} C_{2,r}$
for each $d_i \in C_2$ **do**
  continue the removal to the final decision over all R runs
  if all required clean–removal pairs are valid **then**
    compute final causal influence $\widehat{I}_R(d_i)$
  end if
end for
return C_1, C_2, and verified final causal influences

Here, $\mathrm{TopTieExpanded}(E, s, k)$ denotes the top k rankable candidates in E under score s, together with any candidates exactly tied at the cutoff. Unresolved candidates are handled separately and do not consume ranked shortlist positions.

Final verification. Only candidates in C_2 receive full end-to-end causal verification using $\widehat{I}_R(d_i)$ from Section 3. Final decisions are never used to construct the screening sets. An item that is screened out is therefore not assigned zero causal influence; it is simply not verified to the final decision under the selected PAR budget.

This distinction is important. PAR is not intended to replace the final causal score with an intermediate sensitivity measure. The intermediate checkpoints only decide where the more expensive final verification is performed.

Projected cost. We compare PAR with exhaustive controlled replay using structural stage-slot cost. Let $C_{\text{exh}}^{\text{ctrl}}$ denote the cost of exhaustive controlled replay and $C_{\text{PAR}}^{\text{proj}}$ the cost obtained by applying the PAR survivor sets to the same execution structure. We report

$$S_{\text{proj}} = 1 - \frac{C_{\text{PAR}}^{\text{proj}}}{C_{\text{exh}}^{\text{ctrl}}}. \tag{2}$$

Because PAR is reconstructed from exhaustive trajectories in our evaluation, this quantity is a projected structural reduction rather than measured savings in model calls, tokens, or wall-clock time.

Implementation details for the checkpoint scores, cutoff ties, unresolved outputs, and the equality-based heuristic are provided in Appendix A.

5 Experimental Setup

TradingAgents and interventions. We use TradingAgents (Xiao et al., 2024) as our testbed because it exposes the evidence-to-decision hierarchy required by our study: external evidence enters through specialized agents, passes through several downstream aggregation and decision stages, and contributes to a measurable final decision. This allows us to remove one evidence item, observe its effect at different points in the hierarchy, and replay the affected path to the system output.

TradingAgents is a 12-agent hierarchical multi-agent trading system with social, news, technical, and fundamental analyst branches. Their outputs are consumed by bullish and bearish researchers, aggregated by a Research Manager, passed to a Trader and three risk-management agents, and finally converted by a Portfolio Manager into a BUY, HOLD, or SELL decision.

Our intervention unit is one externally retrieved evidence item. We remove individual social-media or historical-news items while keeping all other evidence fixed. Technical and fundamental branches remain active as context but are not independently intervened upon. For each intervention, we observe three points along the same execution path: G_1, the directly affected Sentiment or News Analyst; G_2, the Research Manager; and the final system decision.

Cases and evidence. Our study covers 25 ticker–date cases in total. Ten cases were used during the initial analysis and method development, while the main quantitative evaluation uses a 15-case cohort. These 15 cases contain 450 social-media and 332 historical-news items, giving 782 removable evidence items per model. Because some cases in this cohort were inspected during development, we treat the analysis as a retrospective evaluation on frozen trajectories rather than prospective held-out validation. The complete case composition is reported in Appendix A.1.

Models and execution protocol. We evaluate three LLM backends: `llama3.1:8b-ctx32768`, `qwen3:8b-ctx32768`, and `ministral3:8b-ctx40960`. Llama and Qwen use a 32,768-token context, while Ministral uses 40,960 tokens. All models use temperature 0 and top-$p = 1$; Qwen additionally uses top-$k = 40$.

Each intervention is evaluated with $R = 3$ runs using fixed seeds

$$1950972707, \qquad 911088213, \qquad 603840044.$$

For each model, the 15-case evaluation contains 45 clean trajectories and 2,346 evidence-removal trajectories. The clean and removal runs use the same seed. Candidates are included only when all required runs are valid; missing or invalid runs are never imputed as zero effect or HOLD.

Checkpoint scores and propagation metrics. At G_1, we compare the clean and intervention outputs of the affected analyst using character-sequence dissimilarity. Social and news candidates are ranked separately because their local representations are modality-specific.

At G_2, we compare the Research Manager's native investment-plan output. The screening score is lexicographic: absolute change in the parsed five-level investment rating, whether the rating changes, and textual dissimilarity between the clean and intervention outputs. Candidates whose native rating cannot be parsed unambiguously are treated as unresolved and advance conservatively.

For the propagation analysis, checkpoint scores are aggregated across the three runs and compared with final causal influence $\widehat{I}_R(d_i)$ using within-case Kendall's τ_b and Spearman's ρ. Case-level coefficients are then averaged across the 15 evaluation cases.

PAR configuration and faithfulness. We evaluate one frozen PAR operating point with nominal replay budgets $b_1 = 0.8$ and $b_2 = 0.5$. Screening is performed independently for each run. At G_1, ranking is performed within evidence modality; at G_2, surviving candidates are ranked jointly. Exact cutoff ties are retained, unresolved candidates advance, and the resulting survivor sets are combined by union. Final decisions are never used during screening.

Using the final-important set F defined in Eq. 1, if C_K is the set retained for final verification, we report

$$\mathrm{Recall}_{\mathrm{PAR}} = \frac{|F \cap C_K|}{|F|}, \qquad \mathrm{Precision}_{\mathrm{PAR}} = \frac{|F \cap C_K|}{|C_K|}.$$

Efficiency. Exhaustive interventions use controlled replay: unaffected analyst branches reuse their clean state, while the affected analyst and downstream components are rerun. We measure computation in structural stage slots and report the projected PAR reduction defined in Eq. 2. Because PAR survivor sets are reconstructed from exhaustive trajectories, this quantity is a structural projection rather than measured model-call, token, or wall-clock savings.

6 Results

We organize the results around three questions: (i) how strongly sensitivity measured at different propagation checkpoints aligns with final causal influence, (ii) how faithfully PAR retains evidence with measured final causal influence, and (iii) how much structural computation PAR can avoid. Unless stated otherwise, correlations are computed within each ticker–date case and then averaged across the 15 evaluation cases.

6.1 Alignment with Final Causal Influence Is Higher Downstream

We first compare local sensitivity at G_1 and propagated sensitivity at G_2 with the final end-to-end intervention score $\widehat{I}_R$. The evaluation contains 782 frozen evidence items: 450 social-media items and 332 historical-news items. The same evidence set is evaluated under Llama 3.1 8B, Qwen3 8B, and Ministral 3 8B, yielding 2,346 candidate–model observations.

Local sensitivity is only weakly related to final causal influence. For Llama, the mean within-case Spearman correlation with final influence is $\rho = -0.055$ for social evidence and $\rho = -0.102$ for news. For Qwen, the corresponding values are $\rho = -0.008$ and $\rho = 0.075$, and for Ministral they are $\rho = -0.008$ and $\rho = 0.046$. At the downstream Research Manager checkpoint, agreement is substantially higher: $\rho = 0.463$ for Llama, $\rho = 0.760$ for Qwen, and $\rho = 0.629$ for Ministral. The corresponding Kendall correlations show the same pattern (Table 1).

The disagreement is also visible directly in the rankings. Among evidence items with non-zero final causal influence, 133/590 (22.5%) under Llama, 154/578 (26.6%) under Qwen, and 140/532 (26.3%) under Ministral fall in the bottom quartile of their local G_1 ranking. Conversely, 56 Llama, 52 Qwen, and 62 Ministral candidates appear in the top local quartile despite having $\widehat{I}_R = 0$. Thus, evidence that appears weak under native local sensitivity can still have non-zero final causal influence, while strong local sensitivity does not necessarily correspond to a strong final effect.

The same pattern is visible at the individual-case level. On matched candidate subsets, G_2 has higher Spearman agreement with final causal influence than G_1 in 12/15 Llama social cases, 12/15 Llama

Table 1: **Agreement between intermediate sensitivity and final causal influence.** Correlations are computed within each ticker–date case and then averaged across the 15 evaluation cases. G_1 is reported separately for social and news evidence because local scores are modality-specific.

Model	Comparison	τ_b	ρ
Llama	G_1 Social $\rightarrow$ Final	-0.048	-0.055
Llama	G_1 News $\rightarrow$ Final	-0.080	-0.102
Llama	$G_2 \rightarrow$ Final	0.387	0.463
Qwen	G_1 Social $\rightarrow$ Final	-0.005	-0.008
Qwen	G_1 News $\rightarrow$ Final	0.069	0.075
Qwen	$G_2 \rightarrow$ Final	0.662	0.760
Ministral	G_1 Social $\rightarrow$ Final	-0.008	-0.008
Ministral	G_1 News $\rightarrow$ Final	0.035	0.046
Ministral	$G_2 \rightarrow$ Final	0.527	0.629

Table 2: **Faithfulness of the frozen PAR operating point.** *Final-important* denotes candidate–model observations with $\widehat{I}_R > 0$. *Retained* denotes observations propagated to final verification.

Model	Candidates	Final-important	Retained	Recall	Precision
Llama	782	590	590	80.85%	80.85%
Qwen	782	578	516	79.07%	88.57%
Ministral	782	532	534	78.57%	78.28%
Pooled	2,346	1,700	1,640	79.53%	82.44%

news cases, 15/15 Qwen social cases, 14/15 Qwen news cases, and 14/15 Ministral cases for both social and news evidence. The downstream checkpoint is therefore more aligned with final influence in most cases, although we do not interpret downstream sensitivity as a substitute for final causal attribution.

A post-hoc example illustrates this local-to-downstream mismatch. For Llama on AAPL (2026-05-29), one social-media item ranks 29/30 by local sensitivity at G_1 but ranks 1/55 at the downstream Research Manager checkpoint. Its final causal influence is $\widehat{I}_R = 2$, tied at rank 2.5/55 among all evidence candidates in the case. The clean system outputs BUY in all three runs, while removing this single evidence item produces SELL in all three runs, corresponding to paired effects $\Delta = (-2, -2, -2)$. This example is selected for illustration rather than used as an independent statistical result.

6.2 PAR RETAINS MOST FINAL-IMPORTANT EVIDENCE

We next evaluate the frozen PAR operating point against exhaustive end-to-end intervention results. Across the three model backends, 1,700/2,346 candidate–model observations have non-zero final causal influence. PAR retains 1,640 observations for final verification, of which 1,352 are final-important. This gives a pooled recall of 1,352/1,700 = 79.53% and a precision of 1,352/1,640 = 82.44%.

The result is similar across model backends. Under Llama, PAR retains 477/590 final-important candidates, corresponding to 80.85% recall. Under Qwen, it retains 457/578, corresponding to 79.07% recall. Under Ministral, it retains 418/532, corresponding to 78.57% recall. Across the three models, PAR misses 348 final-important candidates. Of these, 199 are screened out at G_1, while a further 149 survive G_1 but are screened out at G_2. The recall loss is therefore distributed across both propagation checkpoints rather than being concentrated at a single stage.

The retained set corresponds to 75.45% of candidates for Llama, 65.98% for Qwen, 68.29% for Ministral, and 69.91% overall. Because shortlist construction is performed per run and then unioned, and because cutoff ties and unresolved candidates advance, the realized retention rate can exceed the nominal replay budgets.

Table 3: **Screening-stage ablations and projected structural efficiency.** Reduction is measured relative to exhaustive controlled replay. The variants retain different numbers of candidates and are therefore not equal-budget baselines.

Policy	Retained	Recall	Precision	Reduction
G_1 only	2,066	88.29%	72.65%	10.61%
G_2 only	1,851	90.12%	82.77%	11.72%
PAR	1,640	79.53%	82.44%	20.70%

PAR is a budgeted approximation to exhaustive end-to-end attribution rather than an exact substitute for it. At the evaluated operating point, it sends 69.91% of candidates to final verification while retaining 79.53% of final-important candidate–model observations. More conservative screening would retain more candidates and approach exhaustive replay. Additional case-level and screening-stage analyses are provided in the appendix.

6.3 PAR Reduces Projected Verification Cost

Exhaustive controlled replay requires 63,342 structural stage slots across the three models. Applying the frozen PAR survivor sets reduces the projected cost to 50,232 slots, avoiding 13,110 slots, or 20.70%. Table 3 compares PAR with screening at only one checkpoint.

Screening only at G_2 gives higher precision than screening only at G_1, consistent with its stronger alignment with final causal influence. Using both stages retains fewer candidates and gives the largest projected reduction, with lower recall.

These results are structural projections from exhaustive trajectories, not measured savings in model calls, tokens, or wall-clock runtime.

7 Discussion and Limitations

We evaluate PAR in TradingAgents. The method itself is not specific to trading: it applies when external evidence moves through identifiable downstream stages and individual evidence items can be removed. We do not test other agent architectures or application domains in this paper.

The two checkpoints do not produce the same type of output. G_1 reflects the local analyst, while G_2 is already a higher-level representation closer to the final decision. For this reason, we cannot say that the stronger agreement at G_2 is caused only by propagation. We can only show that, in our setting, the downstream checkpoint is more aligned with final causal influence than the local one.

PAR trades verification cost for coverage. More aggressive screening reduces the number of interventions propagated through the full system, while more conservative screening approaches exhaustive replay. At our evaluated operating point, PAR reduces projected structural verification cost while retaining most evidence with measured final causal influence. These efficiency results are structural projections from exhaustive trajectories, not measured runtime, token, or model-call savings.

8 Conclusion

We studied evidence-level causal attribution in hierarchical agentic systems by following external evidence from ingestion to the final decision. Across three LLM backends, native local sensitivity showed weak agreement with final causal influence, while sensitivity at a shared downstream representation was substantially more aligned with it.

We introduced *Propagation-Aware Replay (PAR)*, which uses intermediate propagation states to allocate end-to-end causal verification to a subset of evidence candidates. Our results support treating attribution in hierarchical evidence-consuming agentic systems as an end-to-end propagation problem rather than a purely local explanation problem.

AI USE STATEMENT

Generative AI tools were used to assist with code development and review, as well as grammar correction and language editing of the manuscript. All AI-assisted code and text were reviewed by the authors, who take responsibility for the final content of this work.

REFERENCES

Tanzim Ahad, Ismail Hossain, Md Jahangir Alam, Sai Puppala, Syed Bahauddin Alam, and Sajedul Talukder. The misattribution gap: When memory poisoning looks like model failure in agentic ai systems. *arXiv preprint arXiv:2605.22842*, 2026.

Shraddha Barke, Arnav Goyal, Alind Khare, Avaljot Singh, Suman Nath, and Chetan Bansal. Agentrx: Diagnosing ai agent failures from execution trajectories. *arXiv preprint arXiv:2602.02475*, 2026.

Ines Besrour, Jingbo He, Tobias Schreieder, and Michael Färber. Ragenta: Multi-agent retrieval-augmented generation for attributed question answering. *arXiv preprint arXiv:2506.16988*, 2025.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. Contextcite: Attributing model generation to context. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3035.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 8048–8057, 2024. doi: 10.24963/ijcai.2024/890.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Wang Jinlin, 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 *International Conference on Learning Representations*, 2024.

Jun Hou, Priya Pitre, Yi Fang, and Xuan Wang. Edge: Error dependency graph-guided multi-error attribution in multi-agent llm systems. *arXiv preprint arXiv:2609.01360*, 2026.

Dong Ho Kang, Hyeonjeong Cha, and Daein Weon. Knowledge-based zero-replay debugging of multi-agent llm traces. *arXiv preprint arXiv:2606.14805*, 2026.

Fengyuan Liu, Nikhil Kandpal, and Colin Raffel. Attribot: A bag of tricks for efficiently approximating leave-one-out context attribution. In *International Conference on Learning Representations*, volume 2025, pp. 16687–16710, 2025.

Mingyu Lu, Yushan Huang, Chris Lin, and Su-In Lee. Agents that matter: Optimizing multi-agent llms via removal-based attribution. *arXiv preprint arXiv:2605.27621*, 2026.

Guoqing Ma, Jia Zhu, Hanghui Guo, Weijie Shi, Jiawei Shen, Jingjiang Liu, and Yidan Liang. Automatic failure attribution and critical step prediction method for multi-agent systems based on causal inference. *arXiv preprint arXiv:2509.08682*, 2025.

Ming Ma, Jue Zhang, Fangkai Yang, Yu Kang, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. Dover: Intervention-driven auto debugging for llm multi-agent systems. In *International Conference on Learning Representations*, volume 2026, pp. 28059–28087, 2026.

Chen Qian, Peng Wang, Dongrui Liu, Junyao Yang, Dadi Guo, Ling Tang, Jilin Mei, Qihan Ren, Shuai Shao, Yong Liu, et al. The why behind the action: Unveiling internal drivers via agentic attribution. *arXiv preprint arXiv:2601.15075*, 2026.

Jaineet Shah. Causal agent replay: Counterfactual attribution for llm-agent failures. *arXiv preprint arXiv:2606.08275*, 2026.

Stelios Triantafyllou, Aleksa Sukovic, Debmalya Mandal, and Goran Radanovic. Agent-specific effects: A causal effect propagation analysis in multi-agent mdps. *arXiv preprint arXiv:2310.11334*, 2023.

Stelios Triantafyllou, Aleksa Sukovic, Yasaman Zolfimoselo, and Goran Radanovic. Counterfactual effect decomposition in multi-agent sequential decision making. *arXiv preprint arXiv:2410.12539*, 2024.

Yawen Wang, Wenjie Wu, Junjie Wang, and Qing Wang. From flat logs to causal graphs: Hierarchical failure attribution for llm-based multi-agent systems. *arXiv preprint arXiv:2602.23701*, 2026a.

Zehao Wang, Lanjun Wang, Shilong Jin, Junjie Chen, and Yanghua Xiao. Dcfa: Dual-view causal-inspired attribution for failure reasoning in llm-based multi-agent systems. *arXiv preprint arXiv:2609.04749*, 2026b.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *Proceedings of the First Conference on Language Modeling*, 2024.

Yihan Xia, Taotao Wang, Shengli Zhang, Zhangyuhua Weng, Bin Cao, and Soung Chang Liew. Hivemind: Contribution-guided online prompt optimization of llm multi-agent systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29767–29774, 2026.

Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. Tradingagents: Multi-agents llm financial trading framework. *arXiv preprint arXiv:2412.20138*, 2024.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, et al. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. *arXiv preprint arXiv:2505.00212*, 2025.

A Additional Experimental Details

All analyses in this appendix use the same frozen trajectories and screening rules as the main evaluation unless stated otherwise.

A.1 Main Evaluation Cases

The main evaluation contains 15 ticker–date cases with 450 social-media and 332 historical-news items, giving 782 removable evidence items per model. Table 4 reports the full case composition.

Table 4: **Main TradingAgents evaluation cohort.** N_{soc} and N_{news} denote independently removable social-media and historical-news items, respectively.

Case	Ticker / Date	$\mathbf{N}_{\mathrm{soc}}$	$\mathbf{N}_{\mathrm{news}}$	N
11	MSFT / 2026-06-04	30	16	46
12	NVDA / 2026-02-20	30	15	45
13	GOOGL / 2026-03-23	30	24	54
14	NVDA / 2026-05-07	30	15	45
15	AMZN / 2026-01-23	30	30	60
16	MSFT / 2026-02-26	30	19	49
17	GOOGL / 2026-04-22	30	22	52
18	AMZN / 2026-05-29	30	22	52
19	AAPL / 2026-04-21	30	22	52
20	AAPL / 2026-05-12	30	21	51
21	GOOGL / 2026-02-10	30	30	60
22	AAPL / 2026-05-29	30	25	55
23	AAPL / 2026-06-01	30	26	56
24	AAPL / 2026-06-02	30	26	56
25	AAPL / 2026-06-05	30	19	49
	Total	**450**	**332**	**782**

A.2 Controlled Replay and Screening Rules

For each removable item d_i and run r, we remove only that item. The affected analyst and its downstream components are replayed, while components outside this path reuse their clean state from the run with the same random seed.

PAR performs screening separately for each run. At G_1, social-media and historical-news items are ranked within their own modality because their local scores come from different analyst branches. With the frozen budget $b_1 = 0.8$, PAR retains the top $\lceil 0.8n \rceil$ rankable items within each modality, where n is the number of items in that modality.

At G_2, the candidates that survive G_1 are evaluated at the shared Research Manager checkpoint and ranked together. If m candidates reach this stage, the frozen budget $b_2 = 0.5$ retains the top $\lceil 0.5m \rceil$ rankable candidates.

At both checkpoints, exact ties at the cutoff are retained. Candidates whose checkpoint output cannot be scored unambiguously are treated as unresolved and also continue. Unresolved items do not consume ranked shortlist positions.

We use

$$\mathrm{TopTieExpanded}(E, s, k)$$

to denote the top k candidates in the rankable set E under score s, together with every candidate exactly tied at the cutoff. We do not introduce a numerical tolerance, additional rounding, or deterministic tie-breaking.

The survivor sets are constructed independently for each run and then combined by union. An item therefore survives if it is retained in at least one of the R runs. For this reason, the realized retained fraction can be larger than the nominal screening budgets.

Final decisions are never used during screening. An item that is screened out is simply not propagated to final causal verification; it is not assigned zero causal influence.

A.3 Checkpoint Scores

Local checkpoint (G_1). The local score compares the clean output of the affected analyst with its output after removing item d_i:

$$s_{1,r}(d_i) = 1 - \text{SequenceMatcher}\left(H^0_{1,r}, H^{-i}_{1,r}\right).$$

The implementation uses Python's `difflib.SequenceMatcher`. A larger value means that removing the item changes the analyst output more. Social-media and historical-news items are ranked separately at this stage.

Downstream checkpoint (G_2). The Research Manager produces an investment-plan output containing a native recommendation. When this recommendation can be parsed unambiguously, we map

$$\textsc{Sell} = -2, \qquad \textsc{Underweight} = -1, \qquad \textsc{Hold} = 0, \qquad \textsc{Overweight} = 1, \qquad \textsc{Buy} = 2.$$

Let a^0_r and a^{-i}_r denote the clean and removal ratings. Candidates are ranked lexicographically by

$$\left(|a^{-i}_r - a^0_r|,\ \mathbb{1}[a^{-i}_r \neq a^0_r],\ 1 - \text{SequenceMatcher}(H^0_{2,r}, H^{-i}_{2,r})\right).$$

The first component measures the size of the rating change. The second records whether the rating changes at all. The third uses the change in the full Research Manager output to break remaining ties. Missing or conflicting ratings are treated as unresolved and advance.

For the descriptive propagation analysis, checkpoint scores are aggregated across the three runs before computing within-case rank correlations. PAR itself does not use these aggregated scores. Screening is performed separately for each run.

A.4 Equality-Based Early Stop

The frozen implementation also contained an equality-based early-stop heuristic from an earlier development version. Replay could stop when the saved checkpoint handoff was identical between the clean and removal runs.

This happened once in the complete evaluation. The affected item still had non-zero final causal influence,

$$\widehat{I}_R = \frac{2}{3},$$

and the downstream Research Manager output differed. We therefore do not treat equality of the saved checkpoint handoff as a causal certificate. Equality at one stored representation does not guarantee that all state visible to later components, or the final decision itself, is unchanged.

B Additional Propagation Results

B.1 Matched-Candidate Agreement

The main analysis compares the two checkpoints with final causal influence. Here, we additionally compare G_1 and G_2 on exactly the same candidate subsets, restricting the analysis to items with valid downstream scores.

Table 5: **Matched-subset agreement with final causal influence.** G_1 and G_2 are evaluated on the same candidate subset. Values are mean within-case Spearman correlations.

Model	Source	N	$\rho(G_1, \widehat{I}_R)$	$\rho(G_2, \widehat{I}_R)$
Llama	Social	436	-0.049	0.499
Llama	News	326	-0.119	0.355
Qwen	Social	450	-0.008	0.700
Qwen	News	332	0.075	0.778
Ministral	Social	450	-0.008	0.645
Ministral	News	332	0.046	0.560

The downstream checkpoint is more aligned with final causal influence on all six matched subsets. Research Manager ratings are parseable for 762/782 Llama items and all 782 Qwen and Ministral items.

The same pattern appears at the case level. G_2 has higher Spearman agreement with final causal influence than G_1 in 12/15 Llama social cases, 12/15 Llama news cases, 15/15 Qwen social cases, 14/15 Qwen news cases, and 14/15 Ministral cases for both social and news evidence.

B.2 Rank Mismatches

The disagreement between local sensitivity and final causal influence is also visible directly in the rankings. Among items with non-zero final causal influence, 133/590 (22.5%) under Llama, 154/578 (26.6%) under Qwen, and 140/532 (26.3%) under Ministral fall in the bottom quartile of their local G_1 ranking.

The opposite mismatch also occurs: 56 Llama, 52 Qwen, and 62 Ministral items appear in the top local quartile despite having $\widehat{I}_R = 0$.

A smaller subset shows an even stronger disagreement. Under Llama, 54 items are in the bottom quartile at G_1 but the top quartile by final causal influence. This occurs for 42 items under Qwen and 44 under Ministral.

The AAPL example from the main paper is one of the strongest cases. Under Llama on 2026-05-29, the selected social-media item ranks 29/30 at G_1 and 1/55 at the Research Manager checkpoint. Removing it changes the final decision from Buy to Sell in all three runs, with

$$\Delta = (-2, -2, -2), \qquad \widehat{I}_R = 2.$$

C Screening-Stage Ablations and Misses

C.1 Screening-Stage Ablations

We reconstruct the single-checkpoint variants from the same exhaustive trajectories. The variants retain different numbers of candidates, so they show what changes when screening is performed at different points in the system. They are not equal-budget baselines.

G_1-only screening retains more candidates but gives the smallest projected reduction. Screening only at G_2 gives higher pooled recall and precision while retaining fewer candidates, consistent with its stronger alignment with final causal influence.

Using both stages retains fewer candidates and produces the largest projected reduction among the evaluated screening policies, with lower recall. This is the cost–coverage trade-off PAR is designed to expose.

Table 6: **Detailed screening-stage ablations.** Reduction denotes projected structural stage-slot reduction relative to exhaustive controlled replay.

Model	Policy	Retained	Recall	Precision	Reduction
Llama	Exhaustive	782	100.00%	75.45%	0.00%
Llama	G_1 only	720	92.37%	75.69%	7.05%
Llama	G_2 only	640	87.80%	80.94%	10.09%
Llama	PAR	590	80.85%	80.85%	16.28%
Qwen	Exhaustive	782	100.00%	73.91%	0.00%
Qwen	G_1 only	682	87.20%	73.90%	11.37%
Qwen	G_2 only	583	90.31%	89.54%	14.14%
Qwen	PAR	516	79.07%	88.57%	23.16%
Ministral	Exhaustive	782	100.00%	68.03%	0.00%
Ministral	G_1 only	664	84.96%	68.07%	13.41%
Ministral	G_2 only	628	92.48%	78.34%	10.94%
Ministral	PAR	534	78.57%	78.28%	22.65%
Pooled	Exhaustive	2,346	100.00%	72.46%	0.00%
Pooled	G_1 only	2,066	88.29%	72.65%	10.61%
Pooled	G_2 only	1,851	90.12%	82.77%	11.72%
Pooled	PAR	1,640	79.53%	82.44%	20.70%

Table 7: **Final-important misses by model and screening stage.**

Model	Lost at G_1	Lost at G_2	Total
Llama	45	68	113
Qwen	74	47	121
Ministral	80	34	114
Pooled	199	149	348

C.2 Miss Analysis

At the frozen operating point, PAR misses 348 of the 1,700 candidate–model observations with non-zero final causal influence. Of these, 199 are removed at G_1, while 149 survive G_1 and are removed at G_2.

The split differs across the three models. More Qwen and Ministral misses occur at the first checkpoint, while more Llama misses occur after the candidate has already survived G_1. Because the screening stages are sequential, an item removed at G_1 cannot be recovered at G_2.

C.3 Case-Level Faithfulness

The number of candidates differs across ticker–date cases. We therefore also compute recall and precision separately for each case before averaging them.

Across the three model backends, case-macro recall is 80.42% and case-macro precision is 82.91%. These values are close to the pooled candidate-level results reported in the main paper.

D Structural Cost Accounting

The exhaustive controlled-replay implementation reuses analyst branches that cannot depend on the removed item. Each removal trajectory reruns the affected analyst and eight downstream components, corresponding to nine structural stage slots.

For each model,

$$782 \times 3 \times 9 = 21{,}114$$

stage slots are required for exhaustive controlled replay. Across all three models,

$$C_{\mathrm{exh}}^{\mathrm{ctrl}} = 63{,}342.$$

Applying the frozen PAR survivor sets to the same execution structure gives 17,676 projected slots for Llama, 16,224 for Qwen, and 16,332 for Ministral:

$$C_{\mathrm{PAR}}^{\mathrm{proj}} = 50{,}232.$$

The projected reduction is therefore

$$1 - \frac{50{,}232}{63{,}342} = 20.70\%.$$

These values count structural execution slots rather than actual model calls. The executed trajectories can contain additional calls because of internal loops, retries, or repair behavior. PAR was also not run separately as a native end-to-end screened execution. We therefore report these values only as a structural projection.